RZ 3492 (# 99340) 05/12/03
Computer Science 20 pages

Research Report

The Performance of Measurement-Based Routing on Overlay Networks

Ilias Iliadis, Daniel Bauer, Sean Rooney, Paolo Scotton, and Sonja Buchegger

IBM Research
Zurich Research Laboratory
8803 Rüschlikon
Switzerland
{ili,dnb,sco,psc}@zurich.ibm.com

The Performance of Measurement-Based Routing on Overlay Networks

Ilias Iliadis *, Daniel Bauer, Sean Rooney, Paolo Scotton, Sonja Buchegger

{ili,dnb,sro,psc,sob}@zurich.ibm.com
IBM Research, Zurich Laboratory, Säumerstrasse 4, 8803 Rüschlikon, Switzerland

Abstract

The literature contains propositions for the use of overlay networks to supplement the normal IP routing functions with higher-level information in order to improve network-behavior aspects. We consider the use of such an overlay to optimize the end-to-end behavior of some special traffic flows. Measurements are used both to construct the virtual links of the overlay and to establish the link costs for use in a link-state routing protocol. The overlay attempts to forward certain packets over the least congested rather than the shortest path. We present simulation results showing that, contrary to common belief, overlay networks are not always beneficial and can be detrimental. The main aspects and circumstances influencing the behavior of overlay networks are identified.

Key words: Overlay network; QoS routing; Link measurement

1 Introduction

Quality of Service (QoS) in large networks is achievable through the presence of control logic for allocating resources at network nodes coupled with inter-router coordination protocols. The various approaches — ATM, DiffServ, IntServ — differ in the trade-off between the precision with which the behavior of flows can be specified and the cost of the additional control logic. However, none of the approaches are widely used in the public Internet. Increased network capacity has meant that the benefits of resource guarantees are reduced and consequently outweighed by the management overhead. Moreover, for HTTP-type request/response traffic this is unlikely to change as the

* Corresponding author. Tel: +41-1-7248646; fax: +41-1-7248955.

majority of the delay incurred is in the servers [1] rather than in the network, so network guarantees for such flows are of marginal importance.

Applications in which the timeliness of the arrival of data is important, such as continuous media streams, distributed games and sensor applications, would benefit from resource guarantees. Whereas the fraction of Internet traffic that such applications constitute may increase, it is unlikely that this increase will be sufficient to force internet service providers (ISPs) to instrument flow or aggregated flow guarantees. Moreover, it would involve the difficult coordination of policy between the border gateways of autonomous systems of different ISPs.

In an overlay network higher-layer control and forwarding functions are built on top of those of the underlying network in order to achieve a special behavior for certain traffic classes. Nodes of such a network may be entities other than IP routers, and typically these networks have a topology that is distinct from that of the underlying physical network. Nodes in the overlay network use the IP network to carry both their data and control information but have their own forwarding tables as a basis for routing decisions. Examples of overlays are the Gnutella file-sharing network and the Mbone multicast network.

Our approach is to treat traffic requiring guarantees as the exception rather than as the rule. This special traffic is forwarded between network servers with hardware-based packet forwarding across a dedicated overlay. We call these servers *booster boxes* [2]. The routing logic between the booster boxes uses dynamic measurements and prediction to determine the least congested path over the overlay. Traffic that is carried over the booster overlay network is called overlay traffic.

While it is trivial to describe simple idealized scenarios in which overlays bring a gain, the more pertinent question is whether and under what circumstances measurement-based overlay networks are beneficial in realistic networks. The focus of this paper is on the applicability and performance of a measurement-based overlay network.

The remainder of the paper is organized as follows. After a review of related work in Section 2, we outline the general architecture of such an overlay network of booster boxes in Section 3. In Section 4 we describe our detailed simulation of its behavior in diverse scenarios. The simulation results and discussion can be found in Section 5, and the conclusions in Section 6.

2 Related Work

The resilient overlay network (RON) architecture [3] addresses the problem that network outages and congestion result in poor performance of IP routing and long recovery time due to slow convergence of Internet routing protocols such as BGP. RON uses active probing and passive monitoring in a fully meshed topology in order to detect network problems within seconds. Experimental results from a small real-world deployment of a RON have been obtained and demonstrate fast recovery from failure and improved latency and loss rates. Note that Andersen et al. do not claim that their results are representative of anything other than their deployment and no general results for different topologies, increased RON traffic, etc., have been published.

The Detour [4] framework pointed out several routing inefficiencies in the Internet and mainly attributed them to poor routing metrics, restrictive routing policies, manual load balancing, and single-path routing. By comparing actual routes with measurement traces between hosts, Savage et al. found that in almost every case there would have been a better alternative path. They envision an overlay network based on IP tunnels to prototype new routing algorithms on top of the existing Internet; however, they concede that measurement-based adaptive routing can lead to instability and, to the best of our knowledge, no evaluation of the overlay performance has been published.

A different application of overlay networks is content-based navigation in peer-to-peer networks. The goal of content-addressable networks such as Chord, CAN [5], Tapestry, and Pastry [6] is efficient, fault-tolerant routing, object location and load-balancing within a self-organizing overlay network. These approaches provide a scalable fault-tolerant distributed hash table enabling item location within a small number of hops. These overlay networks exploit network proximity in the underlying Internet. Most use a separate address space to reflect network proximity. Pastry, for example, routes on address prefixes [7], and uses probing for network proximity to add new nodes to the topology and can be triggered periodically. Pastry assumes the ability of nodes to determine proximity, e.g. by measuring the round-trip time (RTT) between nodes. The published experimental results have been confined to an emulated network environment where no such measurements have been carried out; instead proximity information has been maintained by the emulated network environment itself.

Although these overlay networks have been shown to work in some specific cases, no extensive simulations or practical measurements on a wide range of topologies have been carried out.

3 Architectural Overview

In this section we briefly outline the overlay architecture which we evaluate by simulation. The overlay network consists of a set of booster boxes interconnected by IP tunnels. Packets are forwarded across virtual links, i.e. the IP tunnels, using normal IP routing. The IP routers are not modified and are entirely unaware of the existence of the overlay network.

Booster boxes that are directly connected across a virtual link are called peers. Note that a single virtual link may correspond to multiple IP paths. Booster boxes peer with other booster boxes with which they are likely to have good connectivity. This is determined using pathchar [8] and/or packet tailgating [9]. Pathchar provides more information than packet tailgating about the entire path but has more restrictive assumptions. Both techniques are known to fail beyond a certain threshold number of hops, because of error amplification. We therefore restrict the hop count of the virtual links to a small number (e.g. four).

Although the establishment of a virtual link between two booster boxes is asymmetrical, both sides must agree to the peering. To determine the accuracy of the link measurement, we use this, together with the fact that boosters boxes have good knowledge of the links they are directly attached to.

The techniques for determining link characteristics require the transmission of a large number of packets and accordingly take a significant amount of time to determine a result. They are adequate for the construction of the overlay network but not for the transient state link measurements used to make forwarding decisions. Booster boxes, on the other hand, measure the current latency and loss probability of the virtual links by periodically exchanging network probes with their peers. This is similar to the Network Weather Service (NWS) described in [10]. The overlay routing and measurement process requires additional traffic, the so-called *overlay traffic overhead*.

Booster boxes maintain the overlay-network forwarding tables using a link-state routing protocol. If the link state on a booster box changes significantly, the forwarding tables are recomputed. Packets are forwarded between booster boxes using classical encapsulation techniques.

4 Overlay-Network Simulation

The simulation process contains the following steps. First, we generate a physical network topology using the Brite [11] topology generator. The result is

a graph consisting of nodes that represent autonomous systems (ASs) and of edges that represent network links with certain capacities and delays, as depicted in Figure 1. In a second step, we populate the network with applications with four sources sending a constant stream of packets to a single sink using UDP as the transport protocol; this corresponds to a sensor-type application, which is one of the target application types presented in the Introduction. To generate "background" traffic and thus congestion we add several TCP or UDP sources. The result of this step is a TCL script that is fed into the NS-2 network simulator [12]. Then, we create an overlay network by adding booster boxes to the network topology. Finally, a small fraction of the applications are reconfigured such that they send traffic over the overlay network. The result of this last step is another TCL script, executable by the NS-2 network simulator.

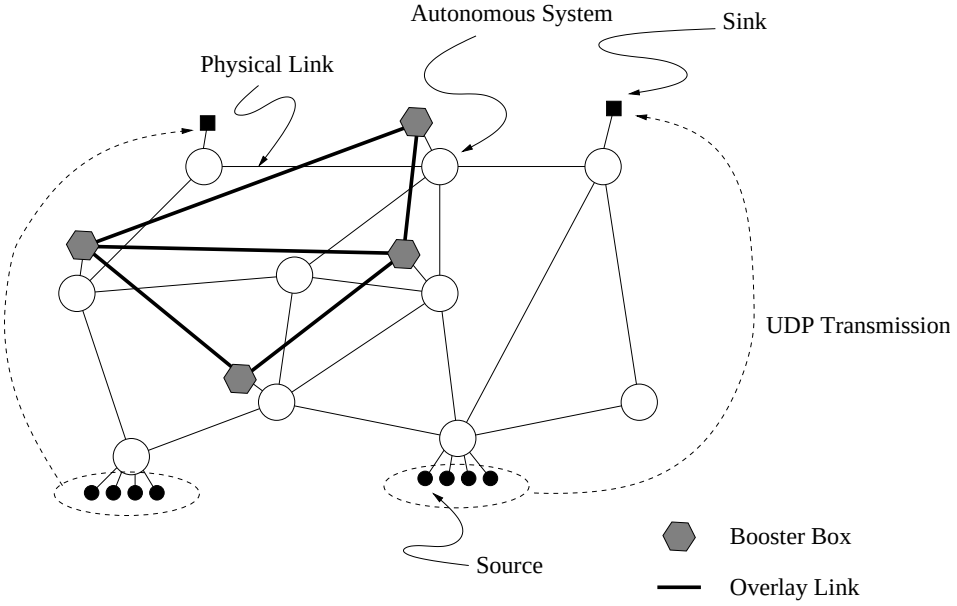

Fig. 1. Sample network

The traffic generated by the applications is analyzed. In particular, we are interested in the packet-drop ratio, i.e. the ratio of dropped packets to the total number of packets sent. We are aware that other alternatives to packet-drop ratio are latency or end-to-end delay. However, packet losses are due to congestion, which implies high delays. For a given topology, we consider two simulation experiments: the first one without an overlay network, called *reference experiment*, and a second one of the same topology with an overlay network formed by the booster boxes, called *overlay experiment*. The applications are divided into two groups: those that use the overlay network, called *boosted applications*, and those that do not use it, called *normal applications*. We obtain the following two sets of results. The first set of results is used as reference and is obtained when no overlay network is present. Then, with an overlay network present, we obtain a second set of results. The two sets of results consist of three average drop-ratios corresponding to all the applications

combined, the boosted applications, and the normal applications, respectively. Figure 2 shows the four steps involved and the resulting two experiments.

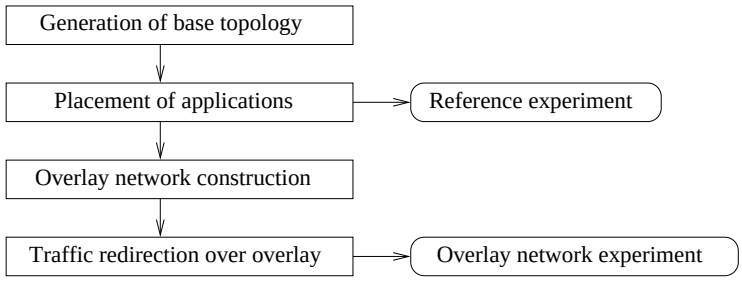

Fig. 2. Steps involved in a single simulation run

The following outlines the assumptions we make:

4.1 Physical Network Topology

We generate random topologies consisting of N nodes based on the Waxman model and using the Brite topology generator. The Waxman-specific parameters are set to $\alpha = 0.9$ and $\beta = 0.2$. A low value for β was chosen to prioritize local connections [13]. The ratio of links to nodes is determined by m and is set to 2, unless otherwise indicated. This means that there are m bidirectional, or $2m$ unidirectional links. The link capacity varies randomly from 1 to 4 Mb/s, and the link propagation delay in the range of 1 to 10 ms. We consider a network size of 100 ASs. This is rather small compared to the Internet but we are constrained by the performance of the NS-2 tool. It would perhaps be more realistic to use the power law [14] rule of Internet ASs, but our network is too small for this to be feasible.

We assume the physical topology to be invariant during a run of the simulation, i.e. nodes and links do not fail and therefore no dynamic routing protocol is needed. Given the fact that routing convergence in the Internet using BGP is rather slow [15] compared with the convergence time of the link-state routing protocol used in the overlay network, it would be expected that the overlay would react to failure more quickly than the physical network does.

To increase the degree of confidence of the results obtained, we conduct a series of E independent simulation runs. At each run, a different topology is generated using the Brite generator with a different seed. We proceed to describe the steps followed at each run.

4.2 Overlay-Network Topology

The overlay network is constructed by adding booster boxes to the ASs with the highest degree of connectivity. This is to ensure that booster boxes are mainly placed in transit ASs. Booster boxes construct virtual links to the four closest neighboring booster boxes. Closeness is determined in terms of hop count, and the path capacity is used as a tie-breaker in the case of equal hop counts. We assume perfect knowledge of the capacities as opposed to the description in the architecture where some error of the measurement of the capacities is inevitable. A booster box never refuses to peer with another booster box, therefore the total number of bidirectional virtual links could be more than four. In our simulation model, an AS is equipped with a booster box by creating an additional node of type "booster box" and connecting it to a single AS node using a high-capacity link of 100 Mb/s and a latency of 100 μs. Inter-AS communication has much longer delay and is more susceptible to packet loss than AS-booster box communication between the AS-booster boxes and the corresponding transit ASs. The hop-count closeness criterion is also used to associate each of the remaining nodes to a node equipped with a booster box. The routing of the boosted applications is realized as follows. First, the two booster boxes associated with the source and destination nodes are identified. [1] The traffic is then sent from the source to the former, then to the latter booster box (if it is not the same), and finally to the destination.

To investigate the effect of the number of booster boxes deployed, various experiments are conducted. We carry out our simulation for three booster/AS ratios: 1:10, 1:5, and 1:2. The remaining parameters, such as the network topology and the positioning and characteristics of the applications, are kept fixed. Let us consider two experiments with a number of B_1 and B_2 ($B_1 < B_2$) booster boxes, respectively. Owing to the nature of the placement algorithm considered, in the second experiment, B_1 of the B_2 booster boxes are placed at exactly the same positions as in the the former experiment.

4.3 Traffic Characterization

Each application consists of four sources sending a constant bit-rate stream of UDP packets to a single sink. Different applications use different bit rates. The bit rates follow a normal distribution with mean of 250 kb/s and standard deviation of 50 kb/s. The packet size is 576 bytes for all applications, as this is the predominant data packet size in the Internet. The background traffic is considered to be either TCP or UDP, with the emphasis given to the former

[1] It may well be that both the source and destination nodes are associated with the same booster box.

as it is the predominant protocol in the Internet. To reflect the diurnal nature of Internet traffic, we consider the traffic sources belonging to one of the two categories: a) sources without silent periods, and b) sources with alternating silent and active periods of length 3.5 and 2.3 s, respectively. A source is only active during the active periods. By assuming that half of the traffic sources belong to the first category and half to the second, the number of active background traffic sources during the active periods is twice the number of active background traffic sources during the silent periods. This reflects the diurnal nature of Internet traffic. Active periods consist of an alternating sequence of on and idle periods. Each source sends traffic at a constant bit rate only during the on periods within active periods. During the idle (within active periods) and silent periods no traffic is being sent. The on and idle periods are independently and exponentially distributed with a mean of B_s ms. Different sources use different bit rates following a normal distribution. Two different levels of congestion are simulated. For light congestion, the bit rates are normally distributed with a mean of 250 kb/s and a standard deviation of 50 kb/s. A high congestion level is simulated using a normal distribution with mean of 500 kb/s and a standard deviation of 100 kb/s. The packet size is 576 bytes for all sources, as this is the predominant data packet size in the Internet. For every topology considered, we conducted a series of experiments by varying the mean length of the on and idle periods of the background traffic, such that their mean is either 1, 10, 100, or 1000 ms. In the remainder of this paper this is referred to as *burst length*. Figure 3 shows the effect of the diurnal traffic model over a link that carries 10 TCP streams with burst length of 1 ms. The application and background traffic sources and sinks are placed at the edge of the network. This is done by randomly distributing the sources and sinks among the g percent of the ASs that have the lowest connectivity. This ensures that the applications are mainly placed at the edge of the network. We consider 20 applications (80 application sources) with 5 of them boosted (20 boosted sources) and 15 normal, as well as 500 background traffic sources. We do not attempt a realistic characterization of traffic produced by an AS, but simply try to ensure that congestion occurs at arbitrary times and for different periods.

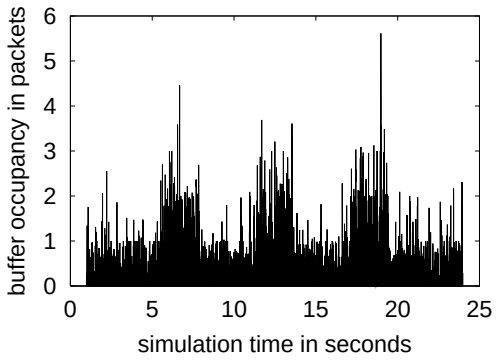

Fig. 3. Buffer occupancy per time for the diurnal traffic model

4.4 Ratio of Overlay Traffic

The average rate of traffic produced by the applications is 20 Mb/s (20 × 4 × 250 Kb/s), of which 5 Mb/s (5 × 4 × 250 Kb/s) is sent over the overlay network. In the case of low congestion, the average rate of traffic produced by the background sources (assuming they are UDP) is given by

$$250 \times \frac{1}{2} \times 250 + 250 \times \frac{2.3 \times \frac{1}{2}}{2.3 + 3.5} \times 250 = 43.642 \text{ Mb/s} ,$$

where the first term of the summation represents the 250 sources with no silent periods, and the second term the 250 sources with alternating silent and active periods of length 3.5 and 2.3 s, respectively. All sources transmit at a rate of 250 kb/s, and the factor 1/2 accounts for the fact that on the average they transmit half of the time because the mean of the on and idle exponential periods is the same.

From the above, it follows that the ratio of overlay to total traffic is in the range of 8% (5/(20+43.642)). In the case of high congestion, the rate of the background traffic is doubled, such that the ratio is in the range of 4% (5/(20 + 2 × 43.642)). In the case of TCP background sources, the average rate of the background traffic is expected to be lower owing to the TCP backoff and slow-start mechanisms. Consequently, the corresponding ratios are expected to be higher.

4.5 Measurement of the Dynamic Metrics

In the experiments the cost of a link is a linear function of the TCP smoothed round-trip time (SRTT) as measured between each booster box and its peers. Using the SRTT prevents the link cost from oscillating wildly. An alternative would have been to use the Retransmission Time Out, which combines both the SRTT and the smoothed mean variance, but as our reaction to congestion is mainly determined by the rate at which we measure and the frequency with which we propagate routing information it is not clear whether this would have brought any additional benefit. As the RTT is not updated using retransmitted packets [16] when TCP times out, and when therefore potentially a packet has been lost, we set the cost of the link to a much higher value than any observed RTT; in this way more weight is attached to loss than to delay. We send a single 50-byte probe every 500 ms between peered booster boxes. The smaller the interval between probes the more responsive the system, but the higher the overlay traffic overhead. We chose the value of the probing interval such

that the network can react to congestion on a subsecond timescale. For reasons of simplicity, the simulation does not use a predictive model for attempting to identify future values of the SRTT such as those described in [10]. The booster-box routing agents update their link-state values every second.

4.6 Simulation Scenarios

We considered a network consisting of 100 ASs and conducted simulations for the following sets of experiments:

- *Background traffic*: two sets of experiments, one for only TCP background traffic sources, and one for only UDP background traffic sources. These represent two extreme cases, although in practice the traffic is expected to be some mix of them.
- *Source/sink distribution*: two sets of experiments, one in which the sources and sinks are randomly distributed among the 50% of the ASs that have the lowest connectivity, and one in which they are distributed among the 80% of the ASs, i.e. $g = 0.5$ and 0.8, respectively.
- *Congestion*: two sets of experiments: *light congestion*, where the mean rate of the background traffic sources is 250 kb/s, and *heavy* congestion, where the mean rate of the background traffic sources is 500 kb/s.
- *Booster boxes*: three sets of experiments corresponding to a deployment of 10, 20, and 50 booster boxes.
- *Burst length*: four sets of experiments for mean burst lengths of the background traffic of 1, 10, 100, and 1000 ms.

Combining the above cases results in a total number of 96 sets. For each we conducted a series of one hundred runs.

4.7 Performance Measures

The objective of this paper is to assess the performance difference observed owing to the deployment of the overlay networks. To realize this, the following measures are considered for each experiment:

- the packet drop ratio of all applications,
- the packet drop ratio of the boosted applications, and
- the packet drop ratio of the normal applications.

The performance of boosted applications is assessed based on the packet drop ratios corresponding to the reference and overlay experiment. There are three possible outcomes: the boosted applications are enhanced, degraded, or un-

changed, depending on whether the packet drop ratio of the overlay experiment is significantly less than, greater than, or equal (within ± 5%) to the packet drop ratio of the reference experiment. The same performance criteria are also used for the normal applications. Note also that the packet drop ratios depend on the simulated time of network operation. We have observed that they converge to a fixed value as the time increases, which implies that the process is ergodic. In our case, it turns out that 24 s of network operation is sufficient to obtain a high degree of convergence within ± 5%.

After the completion of the E runs, the average of the packet drop ratios corresponding to the E runs is calculated. Therefore, there are three average packet drop ratios:

- the average packet drop ratio of all applications,
- the average packet drop ratio of the boosted applications, and
- the average packet drop ratio of the normal applications.

Furthermore, the percentage of the runs in which the boosted and normal applications are enhanced, unchanged, or degraded is also calculated. Owing to the ergodicity, the measures of interest converge to their expected values as the number of runs increases. It turns out that in all sets the measures have practically converged (within a margin of 6%) to the expected values already after 50 runs. The results presented for each of the 96 sets considered correspond to 100 runs, they are therefore extremely accurate.

5 Results

Tables 1 and 2 show the experimental results for the case of TCP background traffic sources, with the sources and sinks randomly distributed among 50% ($g = 0.5$) and 80% ($g = 0.8$) of the ASs having the lowest connectivity, respectively. Similarly, Tables 3 and 4 show the experimental results for the case of UDP background traffic, under the same source/sink distributions of 50% and 80%. The tables contain the following information:

- The number of booster boxes deployed.
- The mean burst length, i.e. the B_s value of the average on and idle periods of the background traffic sources in ms.
- The column labeled "Ref Drop" shows the average packet drop ratio of all applications in the reference experiments. Note that this ratio in the reference experiments does not depend on the number of booster boxes deployed.
- The column labeled "Boosted: Overlay vs. Ref" shows the results for the boosted applications in the overlay and reference experiments.

  - The column labeled "Drops" shows the average packet drop ratios of the boosted applications in the overlay and reference experiments. Note that this ratio in the reference experiments does not depend on the number of booster boxes deployed and may be different from the average packet drop ratio of all applications listed in the third column.
  - The columns labeled "E", "U", and "D" show the percentage of the runs (or the number of runs, given that 100 runs were conducted) in which the boosted applications are enhanced, unchanged, and degraded, respectively.
- The column labeled "Normal: Overlay vs. Ref" shows the results for the normal applications in the overlay and reference experiments.
  - The column labeled "Drops" shows the average packet drop ratios of the normal applications in the overlay and reference experiments. Note that this ratio in the reference experiments does not depend on the number of booster boxes deployed and may be different from the average packet drop ratio of all applications listed in the third column.
  - The columns labeled "E", "U", and "D" show the percentage of the runs (or the number of runs, given 100 runs were conducted) in which the normal applications are enhanced, unchanged, and degraded, respectively

Let us first consider the reference experiments without overlay network. From the results shown, it follows that the average packet drop ratio during the reference experiments (third column) decreases as the mean duration of the on/idle periods (burst length in the second column) of the background traffic increases. In the case of TCP background traffic the losses are in the order of 2 to 4%. However, in the case of UDP background traffic, they are substantially higher owing to the absence of any congestion control mechanism, and range between 8 and 40%. Although such high losses are unusual, they have been observed on backbone routers [17]. Furthermore, the average packet drop ratio during the reference experiments is, as expected, smaller for light congestion than for heavy congestion in all four sets of experiments, i.e. regardless of the distribution value of the sources/sinks ($g = 0.5$ or 0.8) and the nature of the background traffic (TCP or UDP). On the other hand, the average packet drop ratio in the reference experiments is larger for $g = 0.5$ than for $g = 0.8$ in all four sets of experiments, i.e. regardless of the nature of the background traffic (TCP or UDP) and the degree of congestion (light of heavy). This is because as the distribution value increases, there are more nodes sharing the traffic, and therefore congestion at these nodes decreases. It also follows that the trends are the same regardless of the duration of the mean burst length. In particular, in the case of TCP background traffic sources, the percentages of the experiments in which the boosted applications are enhanced, unchanged, and degraded are found to be insensitive to this parameter.

Let us now consider the overlay experiments. In all overlay experiments and scenarios considered, increasing the number of booster boxes causes the average packet drop ratio of the boosted applications to decrease, the percentage

Table 1
TCP background traffic ($g = 0.5$)

Light background traffic										
#BBs	**Burst**	**Ref**	**Boosted: Overlay vs. Ref**				**Normal: Overlay vs. Ref**			
	Length	**Drop**	**Drops**	**E**	**U**	**D**	**Drops**	**E**	**U**	**D**
10	1	3.9	6.5/ 3.9	17	2	81	4.0/ 3.8	17	34	49
10	10	3.4	6.1/ 3.5	20	1	79	3.6/ 3.4	21	31	48
10	100	3.1	5.5/ 3.2	22	3	75	3.3/ 3.1	19	33	48
10	1000	2.8	5.0/ 2.8	23	0	77	3.0/ 2.8	25	25	50
20	1	3.9	4.7/ 3.9	37	6	57	3.9/ 3.8	22	30	48
20	10	3.4	4.1/ 3.5	38	5	57	3.5/ 3.4	21	36	43
20	100	3.1	3.8/ 3.2	39	6	55	3.2/ 3.1	23	29	48
20	1000	2.8	3.3/ 2.8	37	4	59	2.8/ 2.8	24	26	50
50	1	3.9	2.4/ 3.9	71	5	24	3.9/ 3.8	24	28	48
50	10	3.4	2.0/ 3.5	71	4	25	3.4/ 3.4	25	27	48
50	100	3.1	1.7/ 3.2	70	3	27	3.1/ 3.1	29	30	41
50	1000	2.8	1.5/ 2.8	72	1	27	2.7/ 2.8	32	21	47
Heavy background traffic										
#BBs	**Burst**	**Ref**	**Boosted: Overlay vs. Ref**				**Normal: Overlay vs. Ref**			
	Length	**Drop**	**Drops**	**E**	**U**	**D**	**Drops**	**E**	**U**	**D**
10	1	4.3	7.2/ 4.3	16	5	79	4.5/ 4.3	17	37	46
10	10	4.1	6.7/ 4.1	17	3	80	4.3/ 4.1	18	38	44
10	100	4.0	6.7/ 4.0	19	1	80	4.2/ 4.0	17	39	44
10	1000	3.6	6.0/ 3.6	18	3	79	3.8/ 3.6	17	35	48
20	1	4.3	5.4/ 4.3	29	11	60	4.4/ 4.3	16	41	43
20	10	4.1	4.9/ 4.1	34	7	59	4.1/ 4.1	17	46	37
20	100	4.0	4.8/ 4.0	37	5	58	4.1/ 4.0	18	41	41
20	1000	3.6	4.4/ 3.6	33	10	57	3.7/ 3.6	22	35	43
50	1	4.3	3.1/ 4.3	63	6	31	4.4/ 4.3	24	29	47
50	10	4.1	2.6/ 4.1	68	6	26	4.1/ 4.1	23	25	52
50	100	4.0	2.6/ 4.0	70	4	26	4.0/ 4.0	20	29	51
50	1000	3.6	2.3/ 3.6	65	8	27	3.6/ 3.6	23	26	51

of the experiments in which the boosted applications are enhanced to increase, and the percentage of the experiments in which the boosted applications are degraded to decrease. Furthermore, for $g = 0.5$, increasing the number of booster boxes causes the average packet drop ratio of the normal applications to decrease slightly and the percentage of the experiments in which the normal applications are enhanced to increase. In contrast, for $g = 0.8$, this no longer holds. More specifically, this trend holds when the booster boxes are increased from 10 to 20, but it is reversed when they are further increased to 50. This behavior is explained below.

The overlay traffic overhead, which increases quadratically with the number of booster boxes, may interfere with the traffic of the normal applications,

Table 2
TCP background traffic ($g = 0.8$)

Light background traffic										
#BBs	**Burst**	**Ref**	**Boosted: Overlay vs. Ref**				**Normal: Overlay vs. Ref**			
	Length	**Drop**	**Drops**	**E**	**U**	**D**	**Drops**	**E**	**U**	**D**
10	1	3.1	6.2/ 3.3	13	4	83	3.2/ 3.0	16	33	51
10	10	2.7	5.7/ 2.9	15	5	80	2.8/ 2.7	15	30	55
10	100	2.4	5.1/ 2.6	19	5	76	2.6/ 2.4	17	33	50
10	1000	2.2	4.7/ 2.3	15	8	77	2.3/ 2.1	18	29	53
20	1	3.1	3.5/ 3.3	34	7	59	3.1/ 3.0	23	33	47
20	10	2.7	2.9/ 2.9	40	6	54	2.7/ 2.7	23	30	45
20	100	2.4	2.6/ 2.6	49	2	49	2.4/ 2.4	27	33	45
20	1000	2.2	2.3/ 2.3	49	4	47	2.2/ 2.1	25	29	43
50	1	3.1	6.2/ 3.3	55	6	39	3.4/ 3.0	9	17	74
50	10	2.7	5.7/ 2.9	65	3	32	2.9/ 2.7	14	19	67
50	100	2.4	5.1/ 2.6	68	4	28	2.5/ 2.4	18	20	62
50	1000	2.2	4.7/ 2.3	60	7	33	2.3/ 2.1	16	24	60
Heavy background traffic										
#BBs	**Burst**	**Ref**	**Boosted: Overlay vs. Ref**				**Normal: Overlay vs. Ref**			
	Length	**Drop**	**Drops**	**E**	**U**	**D**	**Drops**	**E**	**U**	**D**
10	1	3.6	7.1/ 3.7	9	4	87	3.7/ 3.6	15	41	44
10	10	3.3	6.6/ 3.5	11	4	85	3.4/ 3.3	16	33	51
10	100	3.3	6.6/ 3.4	11	5	84	3.4/ 3.3	17	35	48
10	1000	3.0	6.0/ 3.1	13	3	84	3.1/ 2.9	13	37	50
20	1	3.6	4.3/ 3.7	27	6	67	3.6/ 3.6	19	43	57
20	10	3.3	3.8/ 3.5	33	9	58	3.3/ 3.3	17	43	54
20	100	3.3	3.7/ 3.4	32	10	58	3.3/ 3.3	25	32	53
20	1000	3.0	3.3/ 3.1	36	4	60	3.0/ 2.9	19	39	56
50	1	3.6	3.1/ 3.7	47	6	47	3.9/ 3.6	8	14	72
50	10	3.3	2.6/ 3.5	57	4	39	3.6/ 3.3	8	21	69
50	100	3.3	2.5/ 3.4	56	7	37	3.5/ 3.3	10	25	67
50	1000	3.0	2.4/ 3.1	56	3	41	3.2/ 2.9	7	17	67

resulting in increased congestion and, consequently, in a negative influence. As the booster boxes and the normal applications are placed at the ASs with the highest and lowest degree of connectivity, respectively, the degree of interference depends on the extent of overlap between these two regions. Clearly, the overlap increases as either the number of booster boxes or the distribution factor g increases. Apparently, in the case of 50 booster boxes and $g = 0.8$, this interference is so pronounced that adverse effects result.

We now proceed to examine whether and under what circumstances the overlay networks are beneficial. In general, an overlay network can be described as either *beneficial*, *partially beneficial*, *neutral*, or *detrimental*. An overlay network can be described as *beneficial* when the traffic of the boosted applications

Table 3
UDP background traffic ($g = 0.5$)

Light background traffic										
#BBs	**Burst**	**Ref**	**Boosted: Overlay vs. Ref**				**Normal: Overlay vs. Ref**			
	Length	**Drop**	**Drops**	**E**	**U**	**D**	**Drops**	**E**	**U**	**D**
10	1	21.5	25.3/ 21.6	33	7	60	21.7/ 21.5	4	85	11
10	10	13.5	17.1/ 13.7	32	9	59	13.6/ 13.5	15	56	29
10	100	9.1	12.6/ 9.4	30	7	63	9.2/ 9.0	24	35	41
10	1000	8.6	12.0/ 8.9	29	7	64	8.8/ 8.6	24	38	38
20	1	21.5	19.3/ 21.6	59	8	33	21.5/ 21.5	2	93	5
20	10	13.5	12.2/ 13.7	60	4	36	13.4/ 13.5	17	70	13
20	100	9.1	8.6/ 9.4	53	7	40	8.9/ 9.0	28	52	20
20	1000	8.6	8.3/ 8.9	53	7	40	8.4/ 8.6	36	45	19
50	1	21.5	9.9/ 21.6	91	1	8	21.2/ 21.5	11	88	1
50	10	13.5	5.7/ 13.7	86	4	10	13.1/ 13.5	28	66	6
50	100	9.1	3.7/ 9.4	84	0	16	8.6/ 9.0	40	48	12
50	1000	8.6	3.4/ 8.9	84	2	14	8.1/ 8.6	50	38	12
Heavy background traffic										
#BBs	**Burst**	**Ref**	**Boosted: Overlay vs. Ref**				**Normal: Overlay vs. Ref**			
	Length	**Drop**	**Drops**	**E**	**U**	**D**	**Drops**	**E**	**U**	**D**
10	1	40.2	45.0/ 39.7	21	16	63	40.5/ 40.3	0	100	0
10	10	25.9	29.9/ 25.9	30	11	59	26.2/ 26.0	2	94	4
10	100	22.2	26.3/ 22.2	28	13	59	22.5/ 22.2	5	78	17
10	1000	21.7	25.7/ 21.7	25	13	62	21.8/ 21.7	6	78	16
20	1	40.2	36.6/ 39.7	52	22	26	40.3/ 40.3	0	100	0
20	10	25.9	23.2/ 25.9	55	18	27	26.0/ 26.0	0	98	2
20	100	22.2	19.9/ 22.2	57	15	28	22.2/ 22.2	1	93	6
20	1000	21.7	19.8/ 21.7	58	10	32	21.7/ 21.7	10	81	9
50	1	40.2	23.1/ 39.7	96	2	2	40.3/ 40.3	0	100	0
50	10	25.9	12.8/ 25.9	91	3	6	25.8/ 26.0	5	94	1
50	100	22.2	10.5/ 22.2	90	2	8	22.0/ 22.2	6	92	2
50	1000	21.7	10.5/ 21.7	91	3	6	21.2/ 21.7	22	74	4

using the overlay network behaves better and the traffic of the normal applications not using the overlay network no worse than in the reference case. An overlay network can be described as *partially beneficial* when the traffic of the boosted applications using the overlay behaves better and the traffic of the normal applications not using the overlay network worse than in the reference case. An overlay network can be described as *neutral* when the traffic of the boosted applications using the overlay network behaves the same as in the reference case. An overlay network can be described as *detrimental* when the traffic of the boosted applications using the overlay network behaves worse than in the reference case. All four cases are observed in the results.

Table 4
UDP background traffic ($g = 0.8$)

Light background traffic										
#BBs	**Burst**	**Ref**	**Boosted: Overlay vs. Ref**				**Normal: Overlay vs. Ref**			
	Length	**Drop**	**Drops**	**E**	**U**	**D**	**Drops**	**E**	**U**	**D**
10	1	16.1	22.0/ 16.3	26	4	70	16.2/ 16.0	10	66	24
10	10	9.9	14.4/ 10.2	30	5	65	10.0/ 9.8	18	47	35
10	100	6.6	10.6/ 6.8	29	5	66	6.6/ 6.5	24	32	44
10	1000	6.2	10.1/ 6.5	27	6	67	6.4/ 6.1	19	34	47
20	1	16.1	13.3/ 16.3	69	6	25	15.9/ 16.0	13	77	10
20	10	9.9	8.0/ 10.2	63	4	33	9.7/ 9.8	28	54	18
20	100	6.6	5.5/ 6.8	59	4	37	6.3/ 6.5	33	44	23
20	1000	6.2	5.2/ 6.5	60	5	35	5.9/ 6.1	41	38	21
50	1	16.1	7.9/ 16.3	91	3	6	16.0/ 16.0	12	80	8
50	10	9.9	4.1/ 10.2	89	3	8	9.7/ 9.8	24	56	20
50	100	6.6	2.5/ 6.8	81	2	17	6.3/ 6.5	39	34	27
50	1000	6.2	2.5/ 6.5	81	1	18	5.9/ 6.1	39	40	21

Heavy background traffic										
#BBs	**Burst**	**Ref**	**Boosted: Overlay vs. Ref**				**Normal: Overlay vs. Ref**			
	Length	**Drop**	**Drops**	**E**	**U**	**D**	**Drops**	**E**	**U**	**D**
10	1	33.0	41.5/ 33.1	15	13	72	33.1/ 32.9	0	98	2
10	10	19.9	26.3/ 20.1	20	7	73	20.0/ 19.8	4	80	16
10	100	16.8	22.8/ 17.0	20	12	68	16.9/ 16.7	9	70	21
10	1000	16.6	22.6/ 16.8	23	6	71	16.7/ 16.5	15	51	34
20	1	33.0	30.4/ 33.1	50	23	27	32.9/ 32.9	0	99	1
20	10	19.9	16.8/ 20.1	58	19	23	19.7/ 19.8	6	86	8
20	100	16.8	14.3/ 17.0	62	8	30	16.6/ 16.7	8	79	13
20	1000	16.6	14.3/ 16.8	60	10	30	16.4/ 16.5	16	72	12
50	1	33.0	21.3/ 33.1	91	3	6	32.9/ 32.9	0	100	0
50	10	19.9	10.6/ 20.1	91	2	7	19.8/ 19.8	7	89	4
50	100	16.8	8.5/ 17.0	89	3	8	16.7/ 16.7	7	82	11
50	1000	16.6	8.8/ 16.8	89	4	7	16.4/ 16.5	22	59	19

Let us first consider the case of TCP background traffic sources. According to the results, if 10 or 20 booster boxes are deployed, the overlay network will most likely be **detrimental**, as the percentage of the experiments in which the boosted applications are degraded is about 80% and 57%, respectively. The reason for the detrimental behavior is the increased congestion, caused by the concentration of the boosted traffic around a small number of booster boxes, which cancels any potential benefit. In contrast, if 50 booster boxes are deployed the overlay network will most likely be beneficial or partially beneficial, as the percentage of enhanced experiments is significantly greater than the percentage of degraded experiments. More specifically, for $g = 0.5$, the overlay network will most likely be **partially beneficial** because the percentage of the experiments in which the boosted applications are enhanced is

about 70%, whereas the percentage of the experiments in which the normal applications are enhanced or degraded is about 25% and 50%, respectively. Note, however, that the average packet drop ratio of the normal applications remains practically unaffected. This implies that the reduction of the average packet drop ratio of the normal applications at the runs where they are enhanced is significantly larger than the packet drop ratio increase at the runs where they are degraded. For $g = 0.8$, the overlay network will also most likely be **partially beneficial** because the percentage of the experiments in which the normal applications are enhanced is about 60% in the case of light traffic and 56% in the case of heavy traffic, and the percentage of the experiments in which the normal applications are degraded is more than 60%.

In the case of UDP background traffic sources, the boosted applications are enhanced when deploying 20 or 50 booster boxes, with the greater benefit in the latter case. More specifically, the percentage of the experiments in which the boosted applications are enhanced is about 60% in the former and 90% in the latter case. The highest observed value is 96%, corresponding to the case of $g = 0.5$, with heavy congestion, $B_s = 1$ ms, and 50 booster boxes deployed. In this case the average packet drop ratio of the boosted applications decreases from 39.7% in the reference experiments to 23.1% in the overlay experiments. The normal applications are unchanged, as the average packet drop ratios in the reference and overlay experiments are practically the same. For example, in the case of heavy congestion, and $B_s = 1$ ms, in all the experiments the average packet drop ratio of the normal applications was unchanged. In the remaining cases ($B_s \geq 10$ ms), the normal applications are enhanced, resulting in reduced average packet drop ratios. From the above, it follows that the overlay network will most likely be **detrimental** in the case of 10 booster boxes, and **beneficial** in the case of 20 and 50 booster boxes.

The detrimental behavior is observed for a small number of booster boxes and is due to an aggregation effect that comes in two flavors. First, flows that would have taken different paths over the physical network are destined to these booster boxes, thereby increasing the congestion around them and canceling any potential benefit. Second, these flows are forced to take the paths determined by the overlay network. This causes unnecessary congestion and is strongly dependent on both the physical and the overlay network topologies. The detrimental behavior may also be observed for a large number of booster boxes owing to the substantial overlay traffic overhead.

The beneficial and partially beneficial behavior is observed when the number of booster boxes deployed is sufficiently large so as to avoid the detrimental behavior. The deployment of the overlay network affects the normal applications in two ways, one positive and one negative. Paths that the normal applications shared in the reference experiment with the boosted ones tend to carry less traffic in the overlay experiment as the boosted applications are routed

through other, less congested, paths. This in turn implies less congestion and reduced packet drop ratios for the normal applications, which is therefore a positive influence. The negative influence, as described above, is due to the interference of the overlay traffic overhead with the traffic of the normal applications. The traffic not using the overlay network is affected by this overhead without deriving any benefits from it. This effect is worsened by the fact that exchanges of link-state advertisements occur most often in the case of congestion. In conclusion, the beneficial behavior occurs when the positive influence dominates the negative one, and the partial beneficial behavior occurs when the negative influence dominates the positive one.

A distinction can be made between two types of parameters that influence the performance of the overlay: those under the control of the booster-box operators, such as frequency of measurements, routing, or peering strategy, and those not under the control of the booster-box operators, such as network topology or pattern of background traffic. For the first set it is feasible that extensive simulation for precise scenarios would allow useful heuristics to be derived, e.g. never send more than 10% of the total traffic over the overlay. The second are, in general, unknown to the operator.

The results obtained show that in the situations tested overlaying can cause significant deterioration of the network unless the parameters are chosen wisely. The scenarios may or may not be realistic. However, more simulations and modeling are clearly necessary to better understand the behavior of overlays before they can be deployed.

6 Conclusion

We have outlined an architecture for measurement-based overlay networks that allows certain traffic flows to be privileged over others. We presented simulation results showing how this architecture might behave in the public Internet. For a fixed set of parameters, we found that the overlay can be beneficial or detrimental, depending on its size, the underlying topology, the placement of the applications, and the nature of the traffic. The largest degree of enhancement is observed when the number of booster boxes is large and the packet drop ratio high. In practice however, the relative size of the overlay network should be significantly smaller than that of the underlying network. Furthermore, the circumstances under which the beneficial behavior is observed may not directly reflect the dynamic behavior of the Internet. As the behavior of the Internet is more complicated than the simple scenarios considered here, it is not evident that deployment of measurement-based overlay networks would bring any benefit. As a final remark we suggest that proponents of overlay networks need to investigate the effect of their deployment, not only in simple,

idealized scenarios, but on the network as a whole. Techniques for improving and extending the beneficial behavior of the overlay networks are a subject of further investigation.

References

[1] John Cleary, Ian Graham, Tony McGregor, Murray Pearson, Ilze Ziedins, James Curtis, Stephen Donnelly, Jed Martens, and Stele Martin. High Precision Traffic Measurement. *IEEE Communications Magazine*, 40:167–183, March 2002.

[2] Daniel Bauer, Sean Rooney, and Paolo Scotton. Network Infrastructure for Massively Distributed Games. In *NetGames 2002 – First Workshop on Network and System Support for Games*, pages 36–43, Braunschweig, Germany, April 2002.

[3] David G. Andersen, Hari Balakrishnan, M. Frans Kaashoek, and Robert Morris. Resilient Overlay Networks. In *Proc. 18th ACM Symposium on Operating Systems Principles*, Banff, Canada, October 2001.

[4] S. Savage, T. Anderson, A. Aggarwal, D. Becker, N. Cardwell, A. Collins, E. Hoffman, J. Snell, A. Vahdat, G. Voelker, and J. Zahorjan. Detour: a Case for Informed Internet Routing and Transport. Technical Report TR-98-10-05, University of Washington, 1998.

[5] Sylvia Ratnasamy, Paul Francis, Mark Handley, Richard Karp, and Scott Shenker. A Scalable Content Addressable Network. In *Proc. ACM SIGCOMM*, pages 161–172, August 2001.

[6] Antony Rowstron and Peter Druschel. Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems. In *Proc. IFIP/ACM International Conference on Distributed Systems Platforms (Middleware)*, pages 329–350, November 2001.

[7] Miguel Castro, Peter Druschel, Y.Charlie Hu, and Antony Rowstron. Exploiting network proximity in peer-to-peer overlay networks. Submitted for publication, http://www.research.microsoft.com/ antr/pastry/pubs.htm, 2002.

[8] Van Jacobson. How to Infer the Characteristics of Internet Paths. Presentation to Mathematical Sciences Research Institute, April 1997. ftp://ftp.ee.lbl.gov/ pathchar/msri-talk.pdf.

[9] Kevin Lai and Mary Baker. Measuring link bandwidths using a deterministic model of packet delay. In *Proc. ACM SIGCOMM 2000, Stockholm, Sweden*, pages 283–294, 2000.

[10] Rich Wolski, Neil Spring, and Jim Hayes. The Network Weather Service: A Distributed Resource Performance Forecasting Service for Metacomputing. In *Journal of Future Generation Computing Systems*, 1998.

[11] A. Medina, A. Lakhina, I. Matta, and J. Byers. BRITE: Universal Topology Generation from a User's Perspective. Technical Report BUCS-TR2001 -003, Boston University, 2001.

[12] Kevin Fall and Kannan Varadhran. The ns Manual (formerly ns Notes and Documentation. http://www.isi.edu/nsnam/ns/ns-documentation.html, February 2002.

[13] B. Waxman. Routing of multipoint connections. *IEEE Journal on Selected Areas in Communications*, 6(9):1617–1622, December 1988.

[14] M Faloutsos, P Faloutsos, and C Faloutsos. On Power-Law Relationships of the Internet Topology. In *Proc. ACM SIGCOMM*, pages 251–262, Cambridge, USA, August 1999.

[15] Craig Labovitz, Abha Ahuja, Abhijit Bose, and Farnam Jahanian. Delayed Internet Routing Convergence. In *Proc. ACM SIGCOMM 2000, Stockholm, Sweden*, pages 175–187, 2000.

[16] P Karn and C Partridge. Improving Round-Trip Time Estimates in Reliable Transport Protocols. *Computer Communications Review*, 17(5):2–7, August 1987.

[17] Sally Floyd and Vern Paxson. Difficulties in Simulating the Internet. *IEEE/ACM Transactions on Networking*, 9(4):392–403, August 2001.