Predicting and Bypassing End-to-End Internet Service Degradations

Anat Bremler-Barr * Edith Cohen ** Haim Kaplan * Yishay Mansour *

Abstract

We study the patterns and predictability of Internet End-to-End service *degradations*, where a degradation is a significant deviation of the round trip time between a client and a server. We use simultaneous RTT measurements collected from several locations to a large representative set of Web sites and study the duration and extent of degradations. We combine these measurements with BGP cluster information to learn on the location of the cause.

We evaluate a number of predictors based upon Hidden Markov Models and Markov Models. Predictors typically exhibit a tradeoff between two types of errors, false positives (incorrect degradation prediction) and false negatives (a degradation is not predicted). The costs of these error-types is application dependent, but we capture the entire spectrum using a precision versus recall tradeoff. Using this methodology, we learn what information is most valuable for prediction (recency versus quantity of past measurements). Surprisingly, we also conclude that predictors that utilize history in a very simple way perform as well as more sophisticated ones.

One important application of prediction is *gateway selection*, which is applicable when a LAN is connected through multiple gateways to one or several ISP's. Gateway selection can boost reliability and survivability by selecting for each connection the (hopefully) best gateway. We show that gateway selection using our predictors can reduce the degradations to half of that obtained by routing all the connections through the best gateway.

I. INTRODUCTION

The Internet, to a large extent, is used for interactive communication; with many applications being highly sensitive to temporary link congestion or server overloads. The Internet is composed of thousands of Autonomous Systems (ASes) that peer with each other; and routing between different ASes is governed by the Border Gateway Protocol (BGP) version 4 [1]. BGP4 effectively scales up to support connectivity between the large number of distinct ASes and IP addresses: there are typically multiple paths between two hosts, and BGP is able to select a path and route along that path. On the flip side, since BGP path selection is guided by static heuristics and policies, the selected path is often suboptimal. BGP is known to be vulnerable to long convergence periods following link and router failures [2]; since BGP does not incorporate frequent active measurements, end-to-end performance can suffer from temporary link congestion that can be caused by time of day effects or flash crowds. Moreover, even when such "temporary" effects are discounted, since BGP selection metrics do not incorporate even infrequent measurements, it typically does not obtain a path with better *average* performance in terms of loss-rate or propagation delay – it is well known that round-trip-times (RTT) of selected paths may not satisfy the triangle inequality [3].

The focus of our study is the occurrence of service degradations (*degradations*) from the viewpoint of users contacting Web servers on the "global" Internet, where degradations are events that noticeably affect most interactive applications. We characterize degradations and use measurements from 4 different ASes to study their location and properties; We then explore the *predictability* of degradations by first developing a framework for comparing different predictors and developing prediction algorithms; Last, we explore applications for these predictions aimed at *bypassing degradations*.

* School of Computer Science, Tel Aviv University, Tel Aviv 69978, Israel.
E-mail: {haimk,natali,mansour}@cs.tau.ac.il.
** AT&T research labs, 180 park ave. Florham park, NJ, USA. Email: edith@research.att.com.

Characterizing degradations and their properties

In order to obtain a reasonable set of "representative" destinations, we used a proxy log and weighted each destination by the number of logged HTTP requests. Results on degradations were then scaled according to these weights; this weighting is important for reflecting on occurrence of degradations under normal usage, since there is a large skew in popularities of servers and there is correlation between popularity and reliability (popular servers tend to be more reliable).

We define degradations as a deviation from "normal" of the RTT (Round Trip Time) between two points. (We define "normal" according to the minimum or the median RTT measurement – we show that in most cases the difference between minimum and median is small compared to the deviations in RTT we consider as degradations). We chose to use RTT, since changes in RTT are typically correlated with other quality measures such as change in bandwidth; the RTT can be measured with very small overhead; and the RTT is feasible to obtain on paths for which we don't have control of hosts at both ends. We then discretize degradations into *levels*, that range from deviation of 50ms to packet loss. Smaller deviations do occur, but most applications (in particular Web browsing) are less sensitive to them.

We collected simultaneous periodic RTT measurements from four different Autonomous Systems (ASes) to the same set of destinations. From this data we extracted information as to the location of the cause of degradations. (e.g. if destination was unreachable from 4 of the ASes then the trouble location is likely to be close to the host and otherwise, it is likely to be on the path to the host). We also identified *blackout periods* which are continuous periods where degradations is experienced between a gateway and a destination. We observed that the majority of blackouts were not shared on all gateways, which indicates that degradations can be bypassed.

Predicting degradations

We considered the problem of predicting whether a measurement will experience degradation based on the history of previous measurements (to the same destination from the same point). We use tools commonly used in Machine Learning and Information Retrieval fields to both develop and compare different predictors, and apply predictors based on Markovian models and Hidden Markov Models.

We compare the quality of different predictors using the *precision-recall* curve. This curve captures, for each predictor, the complete tradeoff between cost of false-positives (incorrectly predicting a good measurement to be "degraded") and false-negatives (incorrectly predicting a "degraded" measurement to be good). The relative cost of false-positives and false-negatives is highly application-dependent. The tradeoff curve, however, captures the entire spectrum, and thus, ensures that our results are robust. In our context, *recall* is the fraction of the degradations which the predictor predicts correctly and *precision* is the fraction of our degradation-predictions that turned out to be correct.

We confirm the basic intuition that prediction quality improves with more measurements and with more recent measurements, and quantify the importance of recency and quantity for prediction. The design of our prediction algorithms (which we will refer to as predictors) exploited this observation. Typically, the relative performance of the predictors we evaluated was consistent across different recall values (that is, if predictor had a better precision on a particular recall value then this was often the case on all recall values). We find that simple predictors, which utilize history in a simple way, matched the performance of more complex ones. For example, the best predictors could predict level-3 degradations (RTT increase of 200ms or more) on recall of 60% and precision of 75%.

Bypassing degradations

One important application of prediction is *Performance based gateway selection* (also termed "intelligent routing"), which is applicable when a LAN is connected through multiple gateways to one or several ISP's or also for a multi-peered AS selecting the best peering link. Gateway selection can boost reliability and survivability by selecting the (hopefully) best gateway for each destination.

• LANs and proxy servers are often connected to several ISPs (to each via a different external IP address), for backup or load balancing considerations. Link selection in this case can be arbitrary: some links can be used only as backups in case of failures of other links; traffic can be partitioned using a simple load balancing; or in the best case BGP defaults can be used. Gateway selection can more fully reap the benefits of this expensive link diversity, by routing traffic through the presently better ISP link.

• ASes typically peer with several ASes, and large ASes peer with each other at several points. When routing a packet, intra-AS routing protocols can have the choice of which gateway to choose. An AS that deploys intelligent routing can improve performance for its clients by better selection of a peering link.

Development of intelligent routing products is driven by BGP limitations and the desire to carefully select, and fully utilize, costly links and peering relations [4], [5], [6], [7], [8], [9]. These devices are mostly intended for LANs that have more than a single peering point to the Internet. These products distribute traffic outgoing from the LAN between the multiple peering points so that the overall performance is optimized from the point of view of the LAN.

Even though many products are offered, predictors and the potential of gateway selection had not been carefully studied. We study the potential performance improvement of gateway selection and show that our predictors can reduce degradations to half of what can be obtained by routing all the connections through the best gateway. We also show that intelligent selection on top of two "unreliable" gateways can surpass in performance the most reliable single gateway.

Last, we consider the overhead of measurements required for these predictions, and provide some preliminary assessment of active versus passive measurements, and of the benefit of clustering destinations. To that end, we observe that static selection (selecting the same gateway per address for the whole measurement period), which utilizes much fewer measurements, can reduce the degradation rate to 75% of that of the best gateway.

Related work

There had been considerable work on increasing resilience of Internet connections. This included replicating content or Web servers (creating mirrors). The "best" destination is then selected either at the server end, as done by Content Delivery Networks (CDNs); at the client end via mirror selection; or by enhancing the network (Anycast [10]).

We complement these mirroring-based approaches. First, for scalability purposes, CDNs are able to target and serve mainly popular Web sites; we operate from the client end and thus able to tailor it to destinations used by the clients (including destinations with low overall popularity). A fundamental difference between our approach and mirroring is that we select a *better path* to the same destination rather than select a *better mirror destination.* Moreover, mirroring is often deployed mainly as a way to handle increased load of requests and resilience to server failures, with all alternative IP addresses (mirrors) lying in the same BGP prefix. In this case, mirroring does not provide resilience from BGP path failures.

Another direction of related work focused on developing an infrastructure for tracking and answering queries on path quality (e.g., the IDmaps project [11]). Our study focuses on tracking route quality from a small set of sources to a set of destinations that represents usage.

A related study explored the constancy properties of Internet routes [12]. Our study is different in that we focus on the "global" Internet (we scale our results by server popularity), whereas they focus on selected set of paths between pairs of hosts in universities and research labs (the nimi infrastructure). Our study also goes beyond path properties and looks at predictability of degradations and applications.

Another related project is RON [13] (Resilient Overlay Networks) which proposed to use routes through different peers (hosts in the network) in order to bypass degradations. The case made for RONs is rooted in the same basic issues that also promote intelligent routing (and gateway selection): essentially, BGP inability to find the best available route. Gateway selection, however, is a different application than Ron's. We focus on a fixed number of gateways and evaluate the potential gain on "normal" Web traffic.

Structure of the paper

In Section II we describe our measurements methodology and data we collected. In Section III we present and motivate our definition of degradations and present the precision-recall metrics for predictors. Section IV studies some basic properties of degradations and Section V utilizes simple predictors to learn what properties of the history are important for prediction. In Section VI we present prediction algorithms and in Section VII we evaluate their performance. Section VIII evaluates gateway selection. Section IX is concerned with the overhead of measurements used for prediction.

II. Data and Measurements

We describe how we collected the basic dataset of IP addresses and how we performed measurements.

A. Obtaining a representative set of Web servers

We downloaded a trace spanning 3 days of traffic through the UC NLANR cache from July 25-27, 2001 [14]. There were 1.96M successful requests (that is, requests with HTTP response code 200 (OK), 302 (Redirect), or 304 (Not Modified)), made to 46,030 distinct hostnames. We then performed several rounds of DNS resolutions on these hostnames, obtaining 35,124 distinct IP addresses. We associated an *importance weight* to each IP address, equal to the fraction of requests in the log directed to hostname(s) which resolved to this address: A single hostname may have multiple IP-addresses (load balancing) and a single IP-address can serve multiple hostnames (virtual hosting or aliasing); When a hostname had several IP addresses, we equally divided its weight among the addresses.

We computed basic metrics (specifically, degradation-rate) over measurements by weighting measurements associated with an IP address according to the importance weight of the IP address. This weighting resulted in a mimicked per-request calculation, since, each address "contributes" proportionally to the fraction of requests associated with it. Our measurements involved all 35K addresses or smaller subsets. When sampling addresses, we used weighted sampling based on importance weights (thus mimicking sampling requests)[1]. Weighting prevented a large fraction of very lightly-used IP addresses from dominating the metrics and resulted in numbers that better reflect actual degradation-rates.

Figure 1 shows the number of IP-addresses and the number of name-clusters responsible for a fraction of requests in the log. A name-cluster is a group of IP-addresses that serves the same set of hostnames.[2] We observed that few largest name-clusters correspond to CDNs, and these large clusters contain IP-addresses that belong to several different ASes and BGP Atoms (see Section IX-A). Most name-clusters, however, correspond to a set of mirrors which lie in the same BGP prefix or atom.

B. Measuring round-trip times.

We made periodic Round Trip Time (RTT) measurements to selected IP addresses. The RTTs were measured using the Perl socket module by sending a SYN packet to the server and waiting for the acknowledgment to come back. The measurements were recorded to the accuracy of a millisecond. We note that these "RTT" measurement include both path delay and server response time (that is, the delay between the time the host receives the SYN and sends back the ACK). Server response time, however, is typically negligible relative to path delay. Moreover, since our purpose is to monitor the end-to-end path, this combined metric is appropriate.

We performed measurements from four different locations: Two locations in California bay area (AT&T Lab in Menlo Park and ACIRI in UC Berkeley) and two locations in New Jersey (AT&T Lab in Florham Park and Princeton University). We refer to these locations as CA-1, CA-2, NJ-1, and NJ-2 *gateways*, respectively. Figure 2 demonstrates our raw data by showing the RTTs measured to two IP addresses from CA-1, and CA-2 during few consecutive hours. Subsequently we use three measurement datasets:

[1] It is possible to draw uniformly from addresses, and then apply importance weights to the metric, or use a weighted sample, and then use uniform weighting in the metric (if no repetitions occur where IP address is sampled more than once). Both have the same expectation as that of the whole set of addresses, but the second one results in smaller variance.

[2] We computed approximate name-clusters by computing the connected components of the IP-address vs. hostname graph.

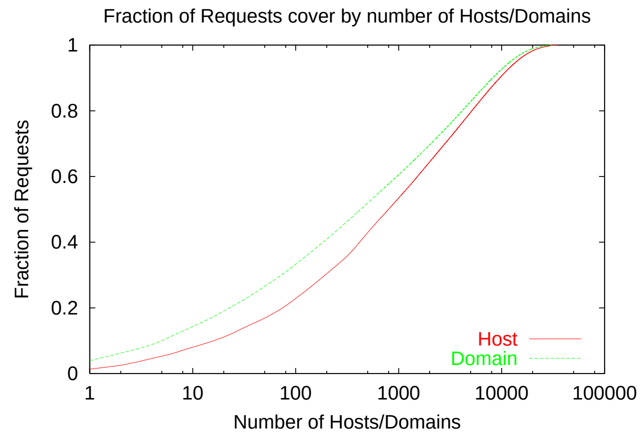


Fig. 1. Number of IP-addresses and name-clusters responsible for certain fraction of HTTP requests.

- One week of hourly measurements to all 35,124 IP addresses from CA-1 and NJ-1 gateways ($24 * 7 = 128$ measurements per gateway/address); this dataset was used mostly for median RTT statistics and evaluation of static gateway selection.
- One week of once-a-minute measurements to a weighted sample (using the weights mentioned above) of 100 addresses, from all 4 gateways[3] ($60 * 24 * 7 = 10080$ measurements per gateway/address).
- 4 hours of once-a-minute measurements from NJ-1 to 1219 addresses constituting of a weighted random sample of 430 addresses together with "similar" addresses (more details will be provided in Section IX-A).

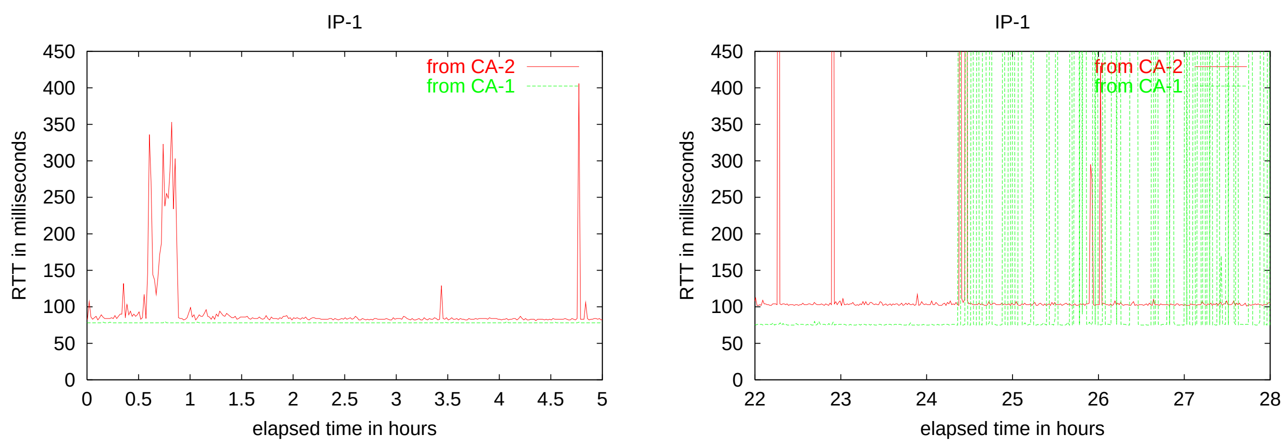


Fig. 2. RTTs measured every minute to two different IP addresses from CA-1 and CA-2. The x-axis show the elapsed time since we started the experiment. The y-axis shows the RTT in milliseconds. In these two snapshots most of long RTTs occur only in one of the gateways and therefore one can bypass them if using cleverly both gateways.

III. Model and prediction metrics

We start with our definition of an end-to-end service degradation. Intuitively speaking, we view a degradation as deviation from the "expected" (median RTT) or "best" (minimum RTT or "propagation delay") performance between the two locations. We show in subsection III-A that there is little difference if degradations are defined with respect to the median or the minimum RTT by showing that they are fairly close and typically highly stable

[3]This data set was used in the evaluation of our prediction algorithms. 8 of these 100 addresses had all "disconnects" measurements and they are not included in statistics. We also removed two addresses that had disconnects 99% of the time and another address that went and remained disconnected in the middle of the measurement duration.

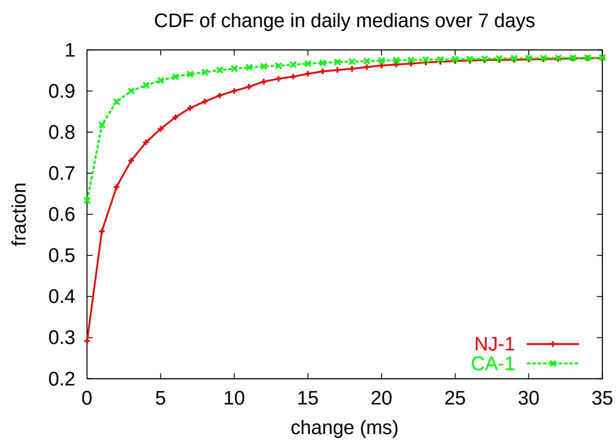
when measured over long time periods. In Subsection III-B we discretize degradations into *degradation-levels*. We then continue and define *predictors* of degradations and present the basic metrics, namely recall and precision, for comparing the performance of different predictors.

A. Median RTT versus minimum RTT

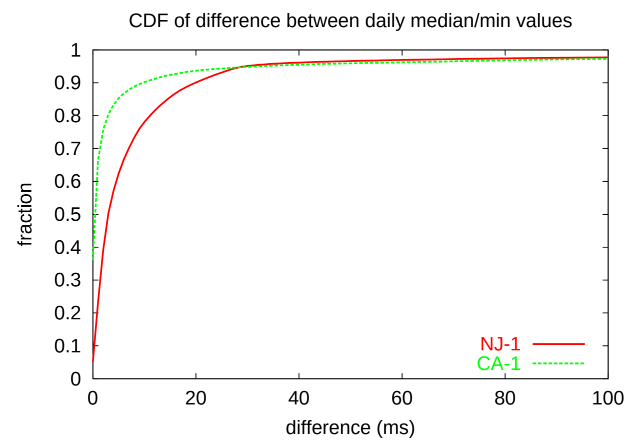
We show that for most IP addresses, the median RTT and minimum RTT are close to each other and fairly static. The fact that the two values are close to each other suggests that our definition of degradations is robust to which base parameter we choose. The fact that they are stable over time reinforces our belief that the base parameters we selected are very robust.

We used hourly measurements for seven consecutive days to each of the 35,124 IP addresses. For each IP-address and each day we computed the median and the minimum RTT. In order to measure stability of the daily medians we consider for each address the absolute difference between the (6 pairs) of medians in consecutive days. A small fraction (about 0.5%) of these transitions were such that exactly one of the days had "disconnects" as the median. We attribute these transitions to server failures rather than route failures (this is supported with the growing number of "home" Web servers and statistics in Section IV). The distribution of the remaining 99.5% of differences is shown in Figure 3(A).

The relation between the median and the minimum RTT was tested by studying the daily difference between the two for each IP address). Figure 3 (B) shows the CDF of these differences. One can see that over 90% of the differences are below 30ms. Due to the long-term server failures discussed above, a small fraction of 0.6% of the differences are between a small minimum RTT and a "disconnects" median RTT. Some of the small and intermediate differences seem to be due to BGP route changes (A route with a longer RTT "dominated" the day).



(A)



(B)

Fig. 3. (A) difference in consecutive daily medians for 35K IP addresses. (B) difference between daily median and daily minimum, for 35K addresses over 7 days.

Figure 4 shows the weighted fraction of addresses such that at least one of the 6 transitions was at least x. The figure shows that about 7% of addresses had some difference above 50ms. About half of these addresses had small differences (between 50-400ms), which is likely to be due to BGP route changes. About 3% of addresses had a change from "reachable" to "disconnected" value, which are likely to be due to a host going down. A separate statistics shows that IP addresses that had large differences typically had more than one. The average number of large differences varied from 2.2 (out of 7) for a difference of 50ms to 1.6 for a difference to/from disconnects. This is because factors such as route changes and server failures are often characteristics of certain addresses. Studies of BGP updates also verify that a large fraction of route changes is due to a smaller set of prefixes [15].

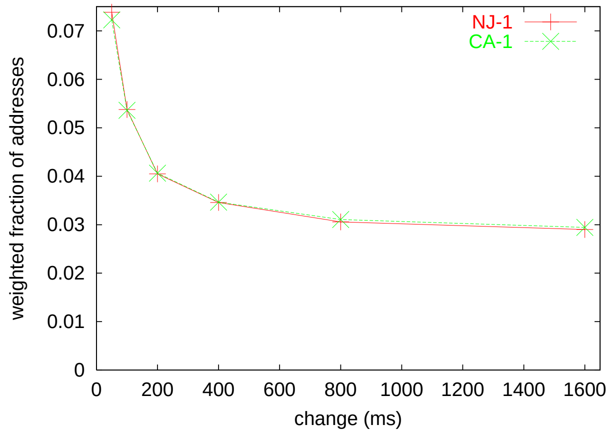


Fig. 4. Weighted fraction of addresses responsible for large differences in daily medians.

B. Defining a degradation

We classified an RTT measurement as *degraded* or *not-degraded* by considering its deviation from the minimum recorded RTT between a gateway and the IP address in a measurement period. When the route is stable, this minimum recorded RTT reflects delay when the network is not congested and we refer to it as *propagation delay*. Defining degradations with respect to the minimum rather than the median RTT is more inclusive, in that more measurements are classified as degradations. As we had seen, however, these two definitions are very similar for the vast majority of IP addresses.

We classified degradations into discrete *degradation-levels* by considering the magnitude of the difference between the RTT and the propagation delay. Each level has a threshold T, and measurements that exceed the respective propagation delay by more than T milliseconds are classified as degradations of this level. We used six levels by exponentially varying the fixed threshold T: the values $T = 50ms, 100ms, 200ms, 400ms, 800ms, 1600ms$ define degradation-levels $1, \ldots, 6$ respectively. Note that the vast majority of measurements classified as degradations of level 6 constitute packet loss. Thus, we refer to level-6 degradation as *disconnects*. To ensure robustness of our conclusions, the smallest degradation-level we worked with is 50ms deviation, which is both small enough to be noticed by end-users and is above the 90th percentile both for the differences between consecutive daily medians and the differences between daily median and daily minimum. Throughout the paper we show results for degradation-level 3 ($T = 200$ milliseconds) and for degradation-level 6 (disconnects). The results for other levels and other gateways were similar.

Figure 5(A) shows the fraction of degraded measurements at each degradation-level and gateway. The figure shows that about half of level-3 degradations were also level-6 degradations, but there are many more level-1 and level-2 degradations than level-6 degradations. Thus, most deviations in RTT are either "small" (below 100-200ms) or "disconnects." Figure 5(B) gives a picture of the total degradation-rate as a function of time from each of the 4 gateways, over a 7 day measurement period. Gateways NJ-2 and CA-2 had considerably higher degradation-rates than NJ-1 and CA-1. None of the gateways had extended durations of total (most destinations) blackouts, although NJ-2 had two bad 20-minute periods. The diurnal effects are also evident in the graphs.

C. Defining predictors

For each IP-address we obtain a sequence of measurements, $RTT_{t_1}, RTT_{t_2}, \ldots$, where RTT_t is the round trip time we recorded at time t.

Our predictors attempt to predict whether RTT_t is degraded using measurements performed before time t. For a fixed degradation-level ℓ, for each measurement RTT_t, we define the binary value $f_t \in \{0, 1\}$ to be 1 if and only if RTT_t is degraded at that level, that is, $RTT_t - P > 50 \times 2^{\ell-1} ms$, where P is the propagation delay.

The predictors we consider attempt to predict f_t using the sequence of binary values f_x for $x < t$. It may

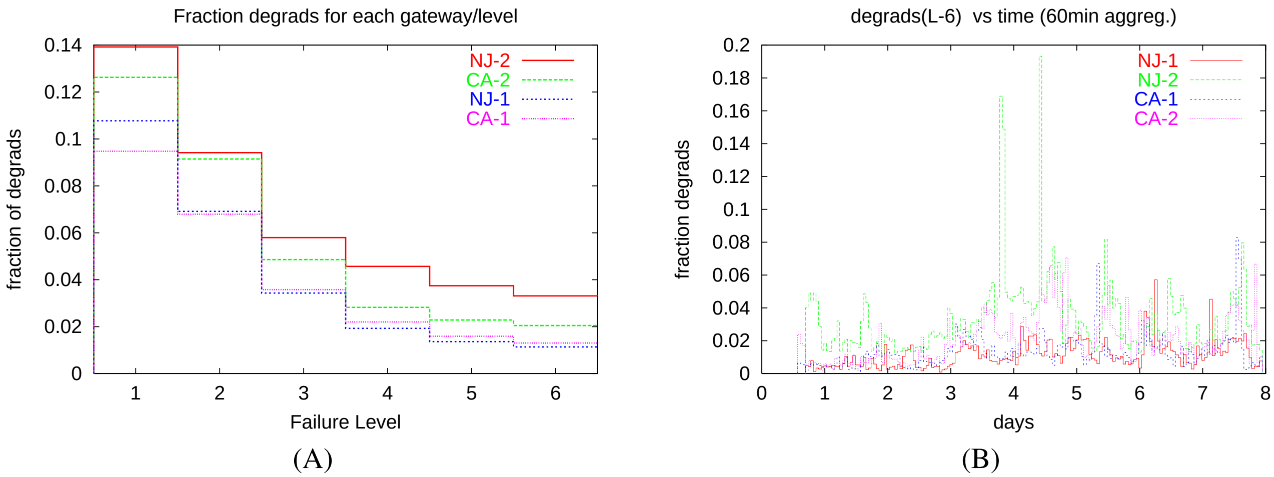


Fig. 5. (A) Histogram showing fraction of degradations of each level for each of the four gateways, using measurements on a weighted sample of 100 IP-addresses. (B) Histogram showing fraction of degradations from each gateway aggregated to hours. The graph shows that (with the exception of CA-2) there were no long duration "total gateway" degradations.

seem that restricting the predictors to use the f_t's rather than the RTT_t's would result in weaker predictors, but the exponential scale according to which levels are defined sets different levels sufficiently apart.Indeed, in Section VII we show that predictors that use information on lower-level degradations do not perform much better than similar predictors that treat lower-level degradations as no-degradations.

Our predictors associate *risk factors* with current measurements based on a history of previous measurements. The risk-factor is a real number with the following interpretation: measurements that the predictor deems to be more likely to be degraded are assigned higher risk factors. Some predictors we consider use risk factors values in $[0, 1]$ that are interpreted as the "likelihood" that the current measurement is degraded.

Formally, let H_t be a set of pairs of the form (z, f_z) $(z < t)$, corresponding to a measurement performed at time $z < t$. For a predictor h, we denote by $h(H_t, t)$ the risk-factor associated by predictor h for time t, given H_t. When H_t is clear from context, we use the shorthand $h(H_t, t) \equiv h(t)$. In order to obtain a set of binary predictions from the real-valued risk-factors, we use a threshold parameter θ. Each value of θ results in a mapping of all measurements to two sets: *predicted degraded*, and *predicted not-degraded*, where a measurement at time t is a predicted degraded if and only if its risk-factor is at least θ, that is, $h(t) \geq \theta$. By sweeping the value of the parameter θ we obtain a spectrum of predictions for the predictor h.

D. The recall and precision metrics

Two standard metrics for the quality of a binary prediction are *precision* $p(\theta)$, which is the fraction of "predicted degraded" that constitute "actual degraded," and *recall* $r(\theta)$, which is the fraction of "actual degraded" which are also "predicted degraded."

$$r(\theta) = \Pr[h(t) \geq \theta | f_t = 1] \tag{1}$$
$$p(\theta) = \Pr[f_t = 1 | h(t) \geq \theta] \tag{2}$$

For each predictor h we sweep θ to obtain a *precision-recall curve*. Lower values of the threshold θ result in more-inclusive degraded predictions, thus the recall increases as θ decreases. We generally expect prediction algorithms to be more accurate when the recall is small (to first catch the "obvious" degradations). Thus, we expect the precision to decrease as the recall increases. In particular, there are two "trivial" points on the precision-recall curve. When the recall is zero (predicted degraded measurements are the empty set), we define the precision to be 1. When the recall is 1, it is always possible to obtain precision equal to the fraction of all measurements that are degraded (by predicting all points to be degraded).

The performance of two different predictors can be compared using their precision-recall curves: The predictor which obtains higher recall for the same precision (or higher-precision with the same recall) performs better on that recall (precision) value. Our predictors result in discrete set of points each for a recall value. To facilitate a comparison we interpolated these points to a continuous curve. Note that the right interpolation is not linear. We explain how we interpolated these points.[4]

IV. Basic degradations statistics

Prior to presenting and evaluating predictors, we gain some insights by considering basic patterns. We use the once-a-minute, 4-gateway, 100 addresses dataset and determine what fraction of degradations are part of a "blackout" and to what extent are blackouts shared by several gateways

A. *Blackout durations*

We looked at continuous intervals of measured degradations, which we dubbed *blackout durations*. For each degraded measurement, we looked at the length of the blackout duration that it lies in. Figure 6 shows the cumulative distribution of measurements that lie in blackout durations of certain lengths, for each of the 4 gateways. The figure shows that most degradations occur in very short blackout durations (of 1-3 consecutive measurements), but considerable fraction occurs in longer durations. This statistics provides some hints to the performance of predictors, since degradations in a short blackout duration are harder to predict with high precision.

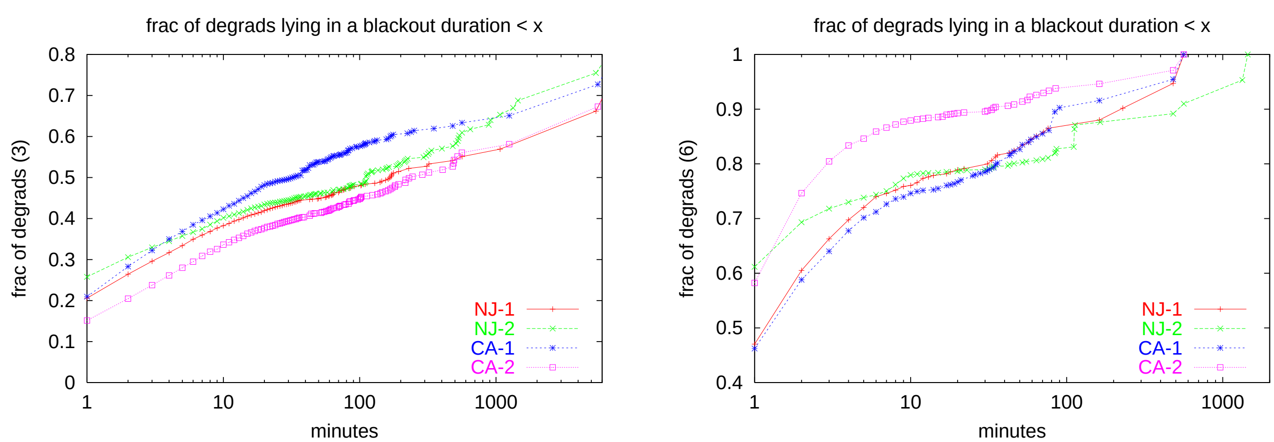


Fig. 6. Cumulative fraction of degraded measurements that lie in a blackout duration of length at most x minutes (For levels 3 and 6 for all four gateways).

B. *Correlation between gateways*

An interesting question is to what extent are blackout durations shared among gateways. The answer would also provide indication as to the location of the cause, since degradations caused by trouble closer to the destination or at the destination server itself are more likely to be shared.

In Figure 7 we considered all 4 gateways jointly. For each degraded measurement, we looked at the length of the current blackout (consecutive degraded measurements prior to current one) and the number of gateways that "shared" this blackout. The figure illustrates that longer blackout durations are more likely to be shared on

[4]Consider two threshold values $\theta_1 < \theta_2$ and the corresponding points $(r(\theta_1), p(\theta_1))$ and $(r(\theta_2), p(\theta_2))$. By definition, $r(\theta_1) \leq r(\theta_2)$. If $r(\theta_1) < r(\theta_2)$, we need to obtain points with intermediate recall values in order to complete the curve. We show how to obtain a point with recall $(1-\alpha)r(\theta_1) + \alpha r(\theta_2)$ for $0 < \alpha < 1$. Consider the respective set of predicted degraded, $R(\theta_1)$ and $R(\theta_2)$, for θ_1 and θ_2. By definition we have $R(\theta_1) \subset R(\theta_2)$. Consider the prediction where the set of predicted degradations consists of $R(\theta_1)$ and a random sample of α fraction of the measurements in $R(\theta_2) \setminus R(\theta_1)$. It is not hard to see that the expected recall of this prediction is $r(\theta_1, \theta_2, \alpha) = (1-\alpha)r(\theta_1) + \alpha r(\theta_2)$ and the expected precision is $p(\theta_1, \theta_2, \alpha) = \frac{(1-\alpha)r(\theta_1)+\alpha r(\theta_2)}{\alpha r(\theta_2)/p(\theta_2)+(1-\alpha)r(\theta_1)/p(\theta_1)}$

all 4 gateways. Thus, are more likely to be caused by failures or overload at the destination server. Particularly interesting are blackouts that were shared by 2 out of the 4 gateways; these degradations are due to the network rather than the destination host and are likely to involve a central link and have larger-scale effects. We observed that the large majority of 2-gateway blackouts involved same-coast pairs (CA-1 and CA-2 or between NJ-1 and NJ-2). 3-way blackouts were very rare, whereas 4-way blackouts included about 10% of degradations.

This statistics provides clues to the effectiveness of gateway selection in avoiding service degradations and loss of connectivity. 4-way blackout durations (durations shared by all gateways) can not be avoided with selection whereas blackouts shared by only a subset of the gateways can potentially be circumvented by selecting a suitable gateway.

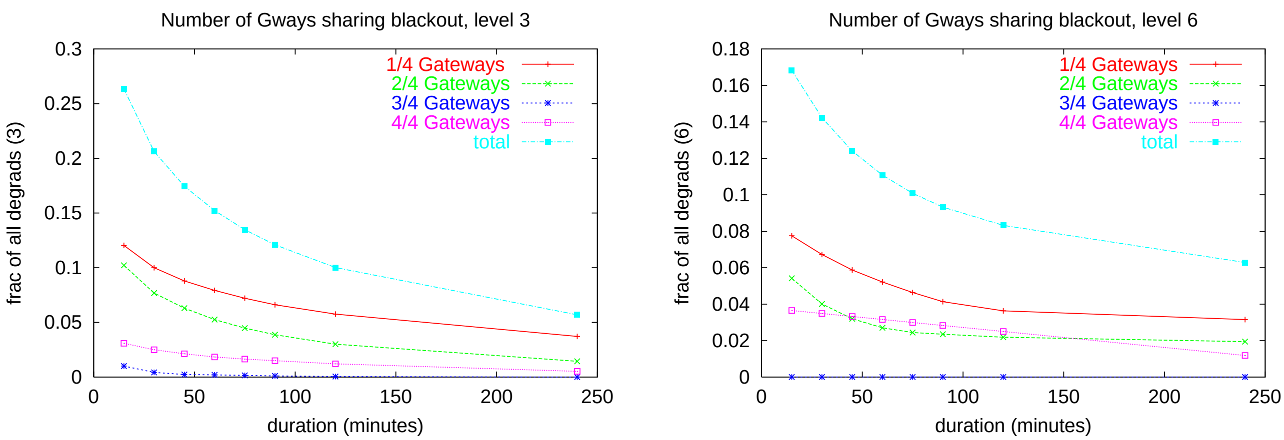


Fig. 7. Number of gateways sharing current blackout, as fraction of total number of degraded measurements. "total" is the number of degraded measurements taken in a current blackout of $x \geq 15$ minutes, as fraction of all degraded measurements. The other curves give the breakdown according to the number of gateways that shared the blackout. A degradation in blackout duration x was considered shared with another gateway, if the other gateway had a blackout duration of at least $x - 5$ just prior to the time of measurement..

V. What is important for prediction?

We learn what basic properties of the measurement history are important for prediction by evaluating the performance of simple predictors using the dataset of periodic 1-minute measurements. In particular, we look at the quality of prediction as a function of the recency and quantity of previous measurements. We also check to what extent the likelihood of a degradation depends on the particular pattern of previous degradations. These basic properties allow us to understand how measurement history can be collapsed to a smaller number of states when considering more involved predictors. We also gain insight on what measurements should be collected for effective prediction.

When predicting degradations based on past behavior, there are two intuitive guiding principles: The *recency principle* states that more recent measurements are more indicative of current performance; *The quantity principle* states that more measurements, through higher confidence levels, yield better predictions. We use two simple families of predictors to illustrate that these intuitive principles hold for our data.

1. **Single-measurement predictors** $\text{SINGLE}_n(t)$, where the risk-factor for the current measurement is based on a single measurement taken n minutes before. Formally, the risk-factor associated at time t by SINGLE_n is $\text{SINGLE}_n(t) = f_{t-n}$. Two simple observations are that (i) since the risk-factor is binary, the predictor provides a single point on the precision recall tradeoff [5]. (ii) for a given data set, the number of predicted degraded (i.e., measurements where $\text{SINGLE}_n(t) = 1$) is equal to the number of degraded (i.e., measurements for which $f_t = 1$) and thus the precision and recall of SINGLE_n are identical.

[5]This endpoint can be completed to a curve using the trivial points with recall values 0 and 1.

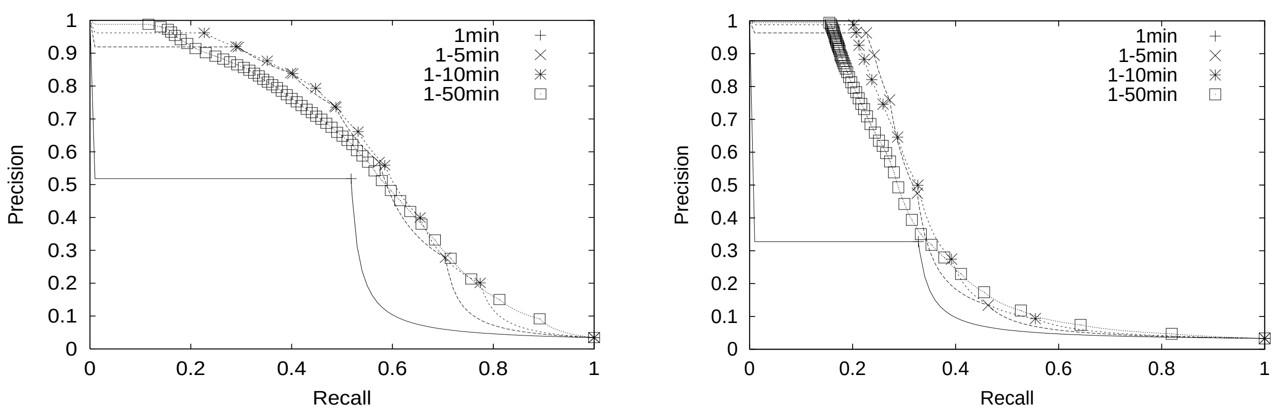
2. **Fixed-window-count** F-WINDOW$_n(t)$ The risk-factor returned by fixed-window-count is the average of the n most recent measurements. F-WINDOW$_n(t) = \sum_{i=t-n}^{t-1} f_i/n$.

The recency principle is illustrated through single-measurement predictors. Table I shows the performance of single-measurement predictors on our data while varying the recency n. As one can see, the more recent the measurement, the more accurate the prediction. The table also shows that although prediction quality does deteriorate with n, the total difference in recall and precision between the most and the least recent measurement that we considered ($n = 1$ and $n = 15$) is 15%.

Minute	NJ-2, level 6 recall (=precision)	NJ-1, level 3 recall (=precision)
1	0.33	0.52
2	0.31	0.50
4	0.30	0.48
7	0.28	0.46
10	0.27	0.45
15	0.26	0.44

TABLE I
DEMONSTRATING THE RECENCY PRINCIPLE THROUGH SINGLE-MEASUREMENT PREDICTORS.

The quantity principle is illustrated in Figure 8 through Fixed-window-count predictors. Figure 8 shows the performance of these predictors for $n = 1, 5, 10, 50$. Fixed-window count predictors for larger values of n rely more on "quantity" and less on "recency." The F-WINDOW$_{50}$ predictor outperforms others in high recall and is outperformed by F-WINDOW$_{10}$ for lower recall. In turn, F-WINDOW$_{10}$ is outperformed by F-WINDOW$_5$ for lower recall. The F-WINDOW$_1$ predictor (which is the same as SINGLE$_1$) is outperformed by all other illustrated window predictors. The F-WINDOW$_{50}$ predictor emerges as the strongest predictor among the four. Even though it achieves slightly worse precision (about 10% less) for low recall, it achieves better precision for high recall values. These results suggest that quantity is in a sense a stronger performance factor of a predictor.



NJ-1, level 3 NJ-2, level 6

Fig. 8. Demonstrating the quantity principle through the fixed-window-count predictors F-WINDOW$_1$, F-WINDOW$_5$, F-WINDOW$_{10}$, and F-WINDOW$_{50}$.

The importance of quantity is further illustrated by Figure 9, by looking at the degradation-rate dependence on certain patterns of degraded and not-degraded in previous measurements. That is, we considered some fixed window of measurements made between j and i minutes before t. For each measurement at time t we looked at the $(j - i + 1)$-bit pattern $f_{t-j}, f_{t-j+1}, \ldots, f_{t-i}$. We then looked at conditional degradation-rates for all possible

2^{j-i+1} patterns. The results in the figure show that the number of degraded in the window is a more significant indicator for degradation-likelihood than the specific pattern of the degraded in the window. These results suggest that when considering more involved predictors, we do not loose significant prediction power by summarizing measurement history by degradation-counts in fixed windows.

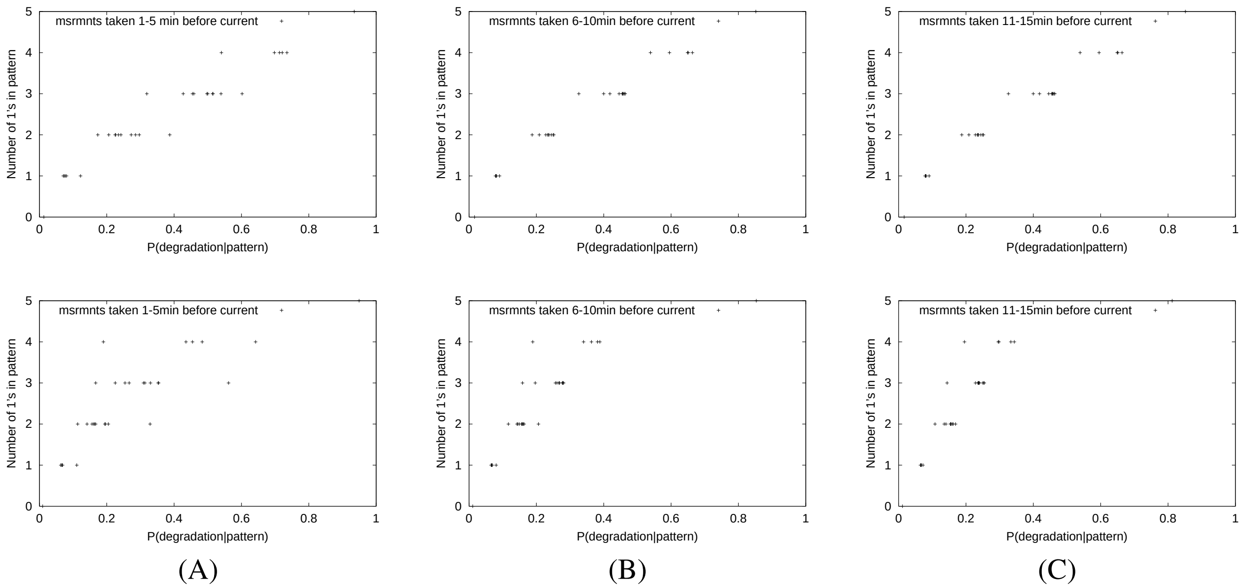


Fig. 9. For each pattern of degraded/not-degraded in a fixed window of 5 consecutive measurments, we plot the number of degraded in the window (level-3 in the upper 3 plots, and level-6 in the bottom 3 plots) as a function of $\Pr(f_t = 1|\text{pattern on previous measurements})$. For plots in column (A) we use the window of $f_{t-5}, f_{t-4}, \ldots, f_{t-1}$ (5 most recent measurements). In (B), we use the measurements $f_{t-10}, f_{t-9}, \ldots, f_{t-6}$. In (C), we use measurements $f_{t-15}, f_{t-14}, \ldots, f_{t-11}$. The output resembles step functions, showing that the number of degraded measurements in a recent window, is more dominant factor for predicting the probability of degradations than the specific history.

VI. Degradation prediction algorithms

Using the lessons from the previous section, we introduce more involved prediction algorithms. We start with the $\textsc{ExpDecay}_\lambda$ (exponential decay) and $\textsc{PolyDecay}_\lambda$ (polynomial decay) predictor families, and continue to two model-building predictors: VW-COVER and HMM.

A. *Exponential-decay predictors*

The $\textsc{ExpDecay}_\lambda(t)$ family of predictors is parameterized by $\lambda \in (0, 1)$. The relative importance of each measurement is exponentially decreasing with its recency (elapsed time since performed).

Let H_t be the set of available measurements before time t. The risk factor assigned by $\textsc{ExpDecay}_\lambda$ is

$$\textsc{ExpDecay}_\lambda(t) = \frac{\sum_{t' \in H_t} f_{t'} \lambda^{t-t'}}{\sum_{t' \in H_t} \lambda^{t-t'}} .$$

The risk-factor is the ratio of the weight of degraded to the weight of all measurements; it gives higher weight for measurements which are more recent; it is additive over measurements that are degraded. $\textsc{ExpDecay}$ predictors with λ closer to 1 put a stronger emphasis on quantity and predictor with λ closer to 0 emphasize recency. The value of λ that performs best depends on the data and can be determined experimentally.

When measurements are periodic (intervals between measurements are the same) the denominator is fixed for all measurements and the risk factor is

$$\textsc{ExpDecay}_\lambda(t) = \frac{1-\lambda}{\lambda} \sum_{i\geq 1} f_{t-i}\lambda^i \ .$$

One of the appeals of the ExpDecay family is that it requires very little bookkeeping, since the risk factor at time t can be easily obtained from the risk-factor at time $t-1$: For periodic measurements we have

$$\textsc{ExpDecay}_\lambda(t+1) = (1-\lambda)f_t + \lambda \textsc{ExpDecay}_\lambda(t) \ .$$

For arbitrary measurements we need to separately track the numerator and denominator values; namely

$$\begin{aligned} \text{N-}\textsc{ExpDecay}_\lambda(t) &= \sum_{t'\in H_t} f_{t'}\lambda^{t-t'} \\ \text{D-}\textsc{ExpDecay}_\lambda(t) &= \sum_{t'\in H_t} \lambda^{t-t'} \ . \end{aligned}$$

and the update calculation when a new measurement arrives is straightforward.

B. Polynomial-decay predictors

Polynomial-decay predictors are similar to exponential-decay predictors in that they follow the recency and quantity principles and are "additive" over degraded measurements. The decay in the weight of a measurement is polynomial instead of exponential; more precisely, the value of a measurement is inversely proportional to $t^{-\mu}$, where t is the elapsed time (in our experiments we measure elapsed time in minutes).

$$\textsc{PolyDecay}_\mu(t) = \frac{\sum_{t'\in H_t} f_{t'}(t-t')^{-\mu}}{\sum_{t'\in H_t}(t-t')^{-\mu}} \ .$$

The computation of polynomial-decay risk-factors requires maintaining more state than exponential-decay; exact computation requires remembering the complete history; a $(1+\epsilon)$-approximation, however, can be maintained by tracking $O(\log_{(1+\epsilon)^{1/\mu}} T)$ values, where T is the ratio of the elapsed time for the least-recent measurement we consider to the minimum measurement gap. Our experimental results are based on an approximate computation with $\epsilon = 0.01$.

Properties of ExpDecay *and* PolyDecay

The value of a measurement declines much more slowly with polynomial-decay than with exponential-decay. A basic qualitative difference is that with exponential decay, the relative value of measurements is fixed over time and depends on the absolute time difference between them: Exponential decay predictors assign the same relative value to measurements performed 1 and 5 minutes ago as to measurements performed 101 and 105 minutes ago. With polynomial-decay predictors, the relative value of two measurements depends on the ratio of their elapsed times, and thus the relative difference in weight of two measurements decreases as time progresses.

The ExpDecay and PolyDecay predictors are parameterized by a decay parameter that determines the importance of a measurement as a function of its recency. Smaller values of λ with ExpDecay and larger values of μ with PolyDecay correspond to faster decay in time. At the extremes of fast and slow decay speeds, these predictors degenerate to the following:

- **Lexicographic predictor** The lexicographic predictor assigns risk factors according to the lexicographic order on the measurements vectors; this is equivalent to having risk factor values of $0.f_1f_2f_3\ldots$. If measurements are spaced at least 1 time-unit apart, all values of the ExpDecay parameter $\lambda \leq 0.5$ result in the lexicographic predictor. As λ approaches (is decreased) to 0.5, predictions converge to the lexicographic predictor. In contrast, PolyDecay predictors do not degenerate to the lexicographic predictor as μ increases, but the lexicographic ordering prevails on *some prefix of the most recent measurements*; this prefix is longer for larger values of μ.

- **Count predictor** The count prediction assigns risk factor value which is the fraction of all measurements that were degradations. With very slow decay, EXPDECAY and POLYDECAY approach count predictions.

We generally expect the performance of the parameterized EXPDECAY and POLYDECAY predictors to be monotonic or bitonic between these two limits.

Model-building predictors

We now discuss predictors that build a model for the data, and then use that model for predictions. The single-measurement and fixed-window-count predictors are special cases of *Markovian* predictors that assume that for the sake of prediction, the entire history can be effectively summarized by a certain number, say n, of recent measurements of RTT to the particular IP address (in which case it is called an *nth order Markov model*).

Since our measurements are casted to be binary (degraded vs. not-degraded), an nth order Markov model for them has 2^n potential states. One way to overcome the explosion in the number of states in an nth order Markov model is to make additional assumptions about the states. For example, use the principles observed on the simple predictors to cluster the states into fewer clusters of "similar" states.

C. *The* VW-COVER *predictor*

The fixed-window count predictors introduced earlier associated higher risk when the window contained more degraded measurements. We had seen, however, that predictors with different window sizes, say F-WINDOW$_{50}$ and F-WINDOW$_5$ are not comparable; which suggests that prediction algorithms that use a mix of different window sizes may outperform the fixed-window count predictors.

The VW-COVER (Variable Window Cover) algorithm try to predict based on counting degraded measurements in windows of varying sizes. This predictor uses test data to construct a prediction model; predictions are then made using this model.

The prediction model is a sequence of pairs of the form (a_i, b_i) where $a_i \leq b_i$. The b_i specifies the window size and the a_i the number of degradations in the window. The model is then used to obtain risk-factors using the formula

$$\text{VW-COVER}(t) = \arg\min_i \sum_{k=1}^{b_i} f_{t-k} \geq a_i \ .$$

That is, the risk-factor at time t is the minimum index i such that at least a_i of the b_i measurements taken before t are degraded.

Next we explain how we construct the model (i.e. the sequence of pairs (a_i, b_i)). Our algorithm is greedy[6] choosing the "best" pair at each step. Suppose we have already chosen $p_1 = (a_1, b_1), \cdots, p_j = (a_j, b_j)$ and we are about to choose the $(j+1)^{st}$ pair. We define T_j, the set of predicted-degraded by $p_1, \cdots, p_j$, as the set of all t such that for some $i \leq j$, $\sum_{k=1}^{b_i} f_{t-k} \geq a_i$. For a pair $p = (a_m, b_m)$ not yet chosen we associate the set of *newly predicted degraded* $N(p)$ to be all $t \notin T_j$ such that $\sum_{k=1}^{b_m} f_{t-k} \geq a_m$. The *utility* of the pair p is defined as

$$w(p) = \frac{|\{t \in N(p) \mid f_t = 1\}|}{|N(p)|} \ .$$

We pick the $(j+1)^{st}$ pair to be the yet unpicked pair with maximum utility. [7]

An example of a prediction model build by VW-COVER is given in Appendix A.

[6] It is similar to the greedy approximation algorithm for the weighted set cover problem. Theoretical results for the classic weighted set cover algorithm establish that the greedy algorithm produces a cover that is a reasonable approximation of the optimal cover.

[7] For gateway selection we used the values $1 - w(p_i)$ as the risk factor (rather than the index i). Typically (but not necessarily) utility values are decreasing with i, and performance is the same whether we use i or $1 - w(p_i)$ as the risk factor.

D. Hidden Markov Model

In contrast to Markovian models, which assume that the entire history can be summarized by a certain number, n, of recent observations, the Hidden Markov Model (HMM) assumes that the system has a state which we do not observe. Formally, the model is an automata, with a finite set of states S, and for each state s there is an output probability $\alpha(s)$ (the probability that we output either 0 or 1). In addition there is a probabilistic transition function, that determines the probability distribution of the next state, given the current state. Intuitively, the model specifies how, in a probabilistic way, the output string is generated. An Example of a model is shown in Figure 10.

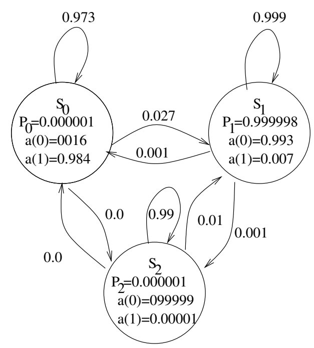


Fig. 10. The HMM model for the data of gateway CA-1 for level 6. P_0, P_1, and P_2 is the initial probability distribution on the states.

The HMM can be applied for prediction. This is done by first computing the probability distribution of the current state, denoted by $p_t(s)$. To predict the probability for degradation we compute the expected value of a degradation given the current state distribution. Formally,

$$\text{HMM}(t) = \sum_{s \in S} \alpha(s) p_t(s) \; .$$

The HMM algorithm updates $p_t(s)$ using the current measurement f_t. (Note that $p_t(s)$ changes both because we have more measurements, and also because the underlying model might have changed states.)

We build a model given a test sequence by using the standard Baum-Welsh algorithm. This algorithm attempts to find a model with the maximum likelihood to produce the given sequence. In our experiments we used an HMM with three states (working with more states did not improve prediction).

The two model-building predictors we presented were tailored to periodic measurements whereas the parametric decay predictors apply when measurements are performed at arbitrary times. We comment that the HMM predictor can be applied to measurements performed at arbitrary times, by treating other times as “missing” data points; when a data point is missing, $p_t(s)$ is updated only due to the HMM automaton changes states, without taking into account the value of the data points.

VII. Experimental Evaluation

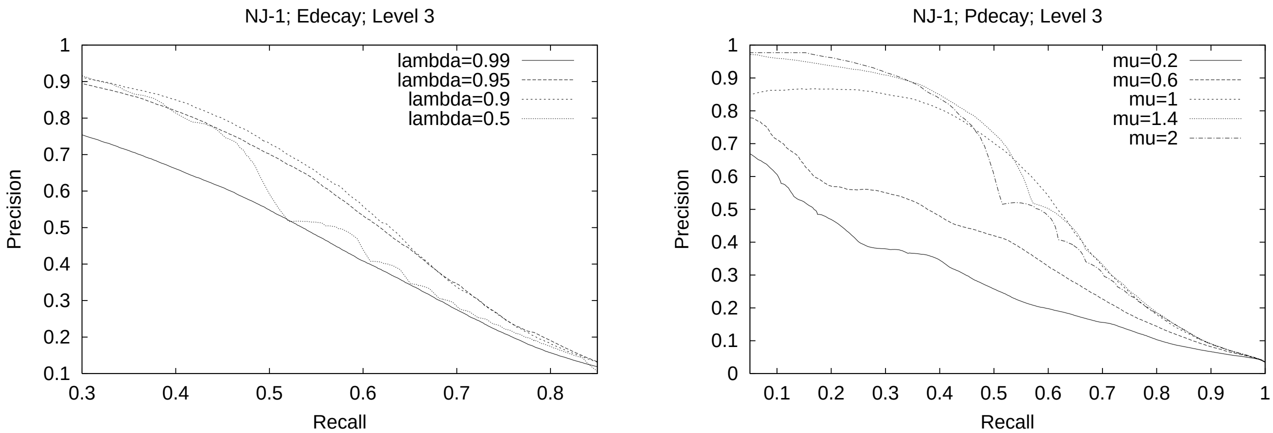
We evaluate the performance of the different predictors introduced in Section VI. We also perform experiments which supports the robustness of our findings.

The evaluation uses the 100 addresses once-a-minute 4 gateway measurements.

A. Parametric-decay predictors

Figure 11 shows the performance of the ExpDecay and PolyDecay predictors. Predictors with slower decay had worse relative performance on low-recall/high-precision area and better relative performance in the high-recall/low-precision area; and vice-versa for fast decay.[8] The PolyDecay predictor with $\mu \geq 1.8$ and ExpDecay

[8]For fast decay predictors the risk factor effectively depends on a small number of measurements. Therefore it obtains a smaller set of values. This may explain the wave pattern evident in the figure for these predictors. It is an artifact of our nonlinear interpolation.



EXPDECAY POLYDECAY

Fig. 11. Recall-Precision curves for predicting level-3 degradations on NJ-1 gateway, using EXPDECAY and POLYDECAY predictors.

with $\lambda \leq 0.65$ resulted in similar predictions. Values of $\mu \geq 1.8$ (fast decay) seemed to perform similar to each other and slightly worse than $\mu = 1.4$. On the slow-decay side, POLYDECAY with $\mu \leq 0.6$ and EXPDECAY with $\lambda \geq 0.99$ performed worse than other predictors, and progressively worse for slower decay. Our results suggest that fine tuning of the decay parameter value is not necessary. The "sweet spot" range of the decay parameter was $\lambda \in [0.75, 0.95]$ for EXPDECAY and $\mu \in [1, 1.4]$ for POLYDECAY, across gateways and degradation-levels, with relatively small variance in performance within this range.

B. Attainable prediction quality

Figure 12 shows the precision vs. recall curves of 6 predictors for degradations of levels 3 (part A) and 6 (part B). The predictors shown are F-WINDOW$_{10}$ (fixed window of 10 previous measurements), F-WINDOW$_{50}$, EXPDECAY$_{0.99}$, EXPDECAY$_{0.95}$, VW-COVER and HMM.

Our predictors achieve good precision vs. recall tradeoffs. For example, a recall of 0.5 (which means capturing 1/2 of the degradations) can be obtained with precision close to 90% (which means that only 10% of the time that we guess a degradation we are wrong). We observe that when one of these predictors dominates another predictor on one recall value it tends to dominate or be close to it on other recall values. This implies that there is a notion of a universal predictor that performs well across costs ratio of false positive error to false negative error. For degradation-level 3 the difference between the best predictor and the worse predictor for any given recall value is at most 20% in precision. The best predictor is VW-COVER, but HMM and EXPDECAY$_{0.95}$ are only slightly worse.

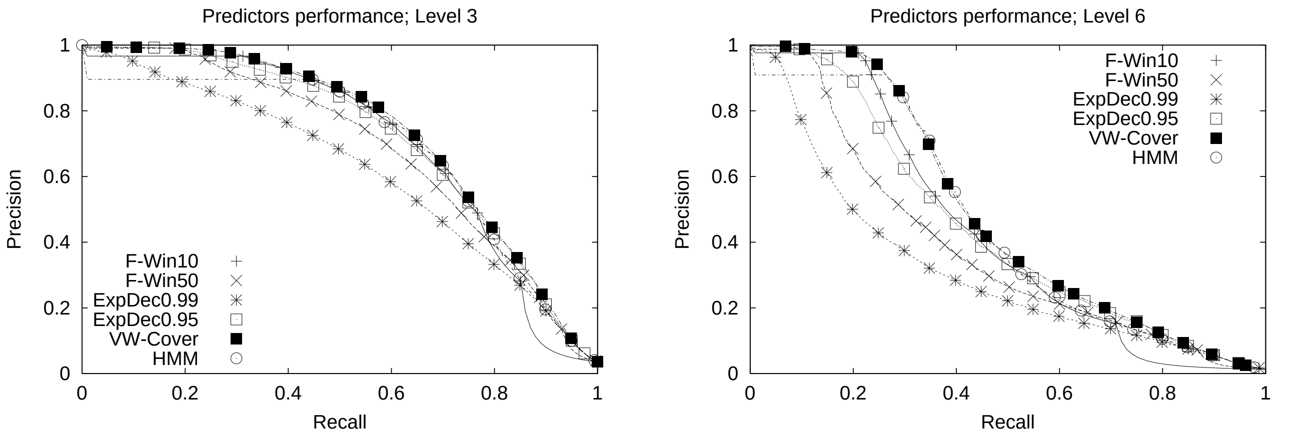
By contrasting Figure 12(B) with (A)the two plots in Figure 12 we see that degradations of level-6 are harder to predict than level-3, as all predictors have lower precision values. For recall value of 0.5, the best predictor shown has precision of about 0.4. The best level-6 predictors are again VW-COVER and HMM, with EXPDECAY$_{0.95}$ being slightly worse. The gap between the precision obtained by the best and worst predictors shown is larger than for level-3 degradations and reaches 40% for some recall values.

Our results show that the model building predictors only slightly outperform the parametric-decay predictors. This is encouraging since EXPDECAY$_{0.95}$ is considerably simpler to implement than HMM or VW-COVER.[9]

C. Robustness of VW-COVER *and* HMM *models*

We checked the universality of the models produced by the VW-COVER and HMM predictors, by applying models produced for one set of measurements to another set of measurements from a different gateway. A repre-

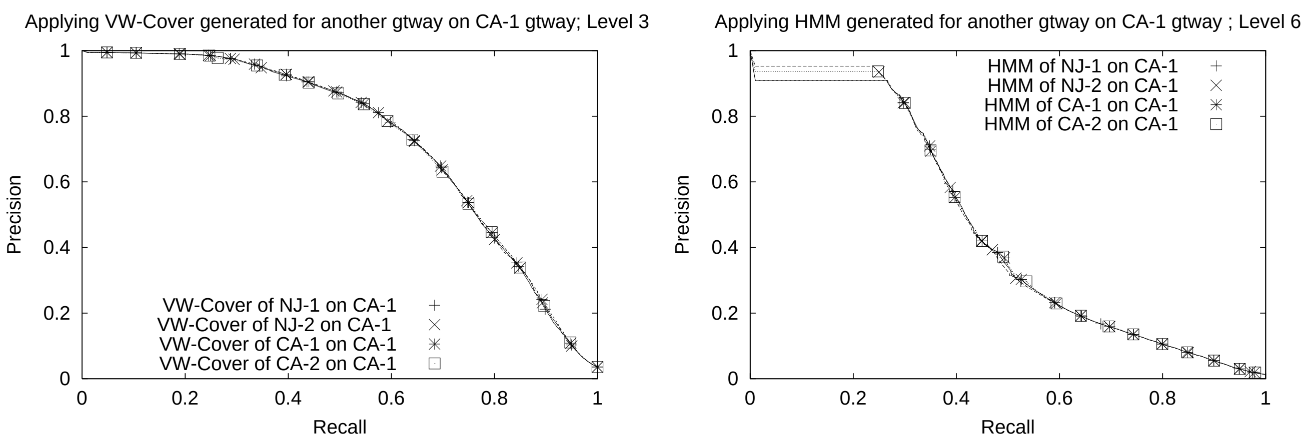
[9] HMM has a reasonably efficient implementation, and VW-COVER requires the most resources as it uses a large number of options at each step.



(A) (B)

Fig. 12. Recall-Precision curves for predicting level-3 degradations (A) and level-6 degradations (B) using F-WINDOW$_{10}$, F-WINDOW$_{50}$, EXPDECAY$_{0.95}$, EXPDECAY$_{0.99}$, VW-COVER, and HMM (CA-1 gateway).

sentative subset of these results is shown in Figure 13. These results indicate that indeed the models produced by VW-COVER and HMM are not limited to specific gateways or time periods and are generally applicable.



(A) (B)

Fig. 13. Applying a model built using one gateway to predict degradations on other gateways. (A) The VW-COVER predictor (B) The HMM predictor.

D. Correlation between degradation-rate and predictability

The predictability of degradations depends to a large extent on their pattern. Random scattered degradations are hard to predict, whereas degradations that appear for some duration. are easier to predict. We observed that a large fraction of degradations are concentrated on blackout events, but nonetheless, these events are relatively rare. Many of the addresses did not experience even one such significant blackout event in the measurement week. We therefore expect that a subset of the addresses, that did experience longer blackouts, would have high-precision predictions whereas other addresses that experiences very few scattered degradations would have low-precision predictions. This expectation is confirmed in Figure 14, which shows the fraction of IP addresses with precision smaller than x for recall 0.5 using the predictor F-WINDOW$_{50}$. The figure shows that only small subsets of IP addresses reach high precision while the majority, 80% for level-6 degradations, and 75% for level 3 degradations, obtain precision lower than 0.4.

The scatter plot of Figure 15 shows that there is some correlation (but not a strong one) between the number of degradations of an IP address and the precision of predictions for that address. The plot for level-6 degradations shows a bimodal pattern where some addresses have predictable degradations and some have "unpredictable" degradations. Interestingly, these patterns occur across degradation amounts. That is, addresses with a large number of degradations tend to have either "random" unpredictable degradations or predictable degradations.

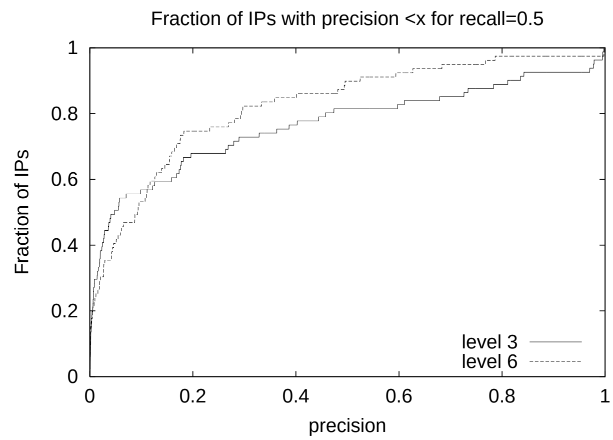


Fig. 14. The cumulative fraction of IP addresses that have precision of at most x (for recall value of 0.5).

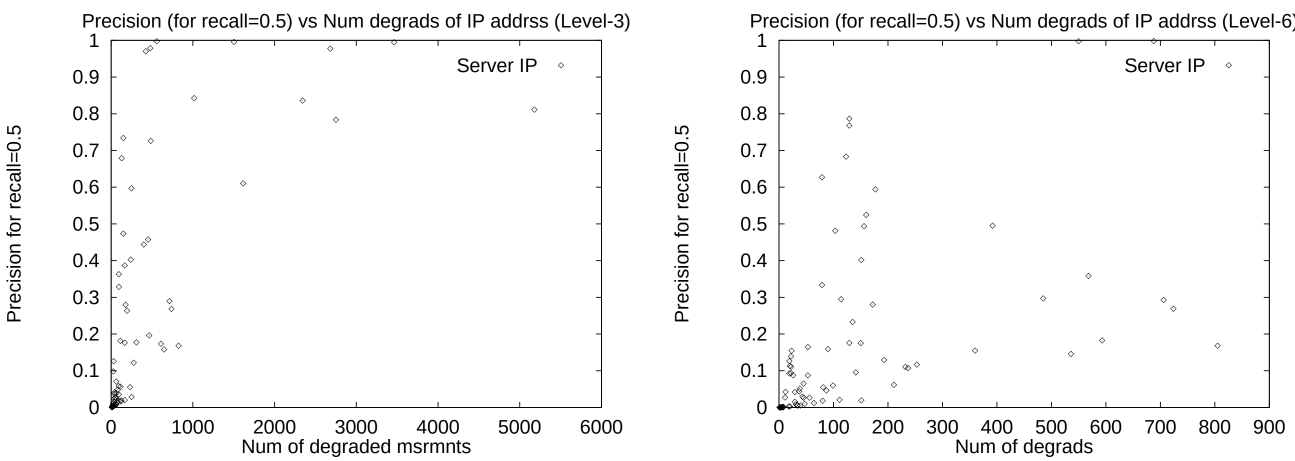


Fig. 15. Scatter plot showing for each IP address the precision (for recall value of 0.5) and the number of degraded measurements.

VIII. Gateway selection

.1 Bypassing degradations

We evaluate the potential of gateway selection for bypassing degradations using simultaneous measurements from the 4 gateways. Table II shows the degradation-rates from each gateway separately and for various selection algorithms. The *Best* predictor shows the omnipotent selection of the minimum degradation-rate of all 4 gateways; that is, selecting (for each measurement time) the gateway with the best measurement. This selection constitutes a lower bound on the performance of gateway selection.

For the $\text{ExpDecay}_{0.95}$ and VW-cover predictors, the *Predicted Best* column shows the degradation-rate when selecting the gateway with the smallest risk-factor for the current measurement time (and the resulting breakdown of what fraction of time each gateway is used in the selection). The $\text{ExpDecay}_{0.95}+$ (and VW-cover+) selections are based on the $\text{ExpDecay}_{0.95}$ (respectively, VW-cover) risk-factors, but in times where a subset of the

level	NJ-1	NJ-2	CA-1	CA-2	Predictor	Predicted Best (breakdown)	
6	1.15%	3.29%	1.34%	2.07%	Best	0.08%	
6					EXPDECAY$_{0.95}$	0.52%	(42%,2%,38%,18%)
6					EXPDECAY$_{0.95}$+	0.53%	(37%, 2%, 38% ,23%)
6					VW-COVER	0.49%	(61%, 1%, 28% ,10%)
6					VW-COVER+	0.51%	(39%, 1%, 39% ,21%)
6					Static	0.86%	(54%, 0%, 35%, 11%)
6					Static 1stD	1.05%	(63%, 0%, 29%, 8%)
6					Static 15% samp	0.88%	(55%, 0%, 33%, 13%)
3	3.45%	5.77%	3.64%	4.90%	Best	0.45%	
3					EXPDECAY$_{0.95}$	1.56%	(30%,5%, 45%, 20%)
3					VW-COVER	1.50 %	(33%, 2%, 51% ,14%)
3					Static	2.41%	(18%, 2%, 66%, 14%)
3					Static 1stD	2.84%	(61%, 0%, 30%, 9%)
3					Static 15% samp	2.44%	(26%, 3%, 58%, 13%)

TABLE II

DEGRADATION-RATES FROM EACH GATEWAY WITH OR WITHOUT SELECTION.

gateways have very low risk-factors, a gateway is selected uniformly at random. This variant results in a more balanced usage of gateways.

The table also shows the results of *static gateway selection* where the same gateway is used throughout the whole period for each IP-address. We evaluated 3 flavors of static selection: The first, *static*, selects the gateway with minimum degradations over the whole duration. The second, *Static 15% sample* selects the gateway with minimum degradations in a sample of 15% of the measurements (this is to gauge "gain" due to variance). The third, *static 1st day*, selects for each IP address the gateway with minimum degradations in the first day. The *static* selection gives a bound on the best possible performance of fixed selection. The *static 1st Day* shows the limitation of static selection for longer durations (especially for level-6 degradations where performance of selection based on measurements of the 1st day is considerably worse than measurements made throughout.

The results shows that active gateway selection resulted in 50% reduction in the degradation-rate with respect to the *best* single gateway. Static gateway selection can avoid at most 25% of degradations. The results also show that it is possible to significantly reduce degradation-rate while maintaining reasonable balance of the load on the different gateways.

Table III lists the results of gateway selection when it is applied to two gateways at a time. Interestingly, when applied to the two worst-performing gateways, CA-2 and NJ-2, the resulting degradation-rate with gateway selection outperforms the best single gateway (NJ-1). The best of these two gateways had level-6 degradation-rate of 2.07% and level-3 degradation-rate of 4.90%. By using a predictor to choose a gateway we got degradation-rates of 1.11% and 2.54%, respectively, which are below the degradation-rates 1.15% and 3.45%, respectively, of the best gateway.

These results also shows the stronger correlation between gateways that are located on the same coast, even though each such gateway used a different autonomous system. Gateway selection on same-coast pair resulted in only 10% reduction in degradations (over the best gateway). This suggest that it is important to select sufficiently "independent" gateways in order to obtain good performance with gateway selection. The methodology we developed can be applied to first select a base set of "complementary" gateways, and then apply gateway selection within the selected set. This behavior also suggests that long hauls over Internet backbones are a primary culprit of path degradations.

Last, these correlations between pairs of gateways reinforce the fact that the improvement we see with selection is not simply an artifact of avoiding "gateway blackouts." First, we observed that long "gateway blackouts" did not occur in our measurement period. Second, gateway blackouts can be bypassed using any pair of reasonably-reliable gateways which utilize different ASes. The fact that only some pairs are effective suggest that the bulk of degradations occur closer to the Internet core.

level	Best	BestGW	PBest	Best	BestGW	PBest	Best	BestGW	PBest
		NJ-1 , NJ-2			NJ-2 , CA-2			NJ-1 , CA-1	
6	0.42%	1.15%	1.05%	0.21%	2.07%	1.11%	0.12%	1.1%	0.54%
3	1.7%	3.45%	3.05%	1.15%	4.90%	2.54%	0.90%	3.45%	1.78%

TABLE III

Degradation-rates when selecting between 2 gateways. The *Best* column shows rate when the gateway with best measurement is selected; The *P.Best* column shows selection of the gateway with the best prediction (using $\text{ExpDecay}_{0.95}$).

An important factor in the practicality of gateway selection is how sensitive it is to the recency of measurements. The predictions we evaluated so far were performed on measurements taken as recently as 1 minute before the prediction. Figure 16 shows that performance of gateway selection when the $\text{ExpDecay}_{0.95}$ predictor uses older measurements. The figure shows that performance is not very sensitive to recency: even when the most recent measurements used is 30 minutes before the prediction is made, gateway selection is still able to significantly outperform the best single gateway. Interestingly, the performance gain with 30 minute old data is close to that of static selection. These results quantify by what amount availability of recent measurements boost the effectiveness of selection.

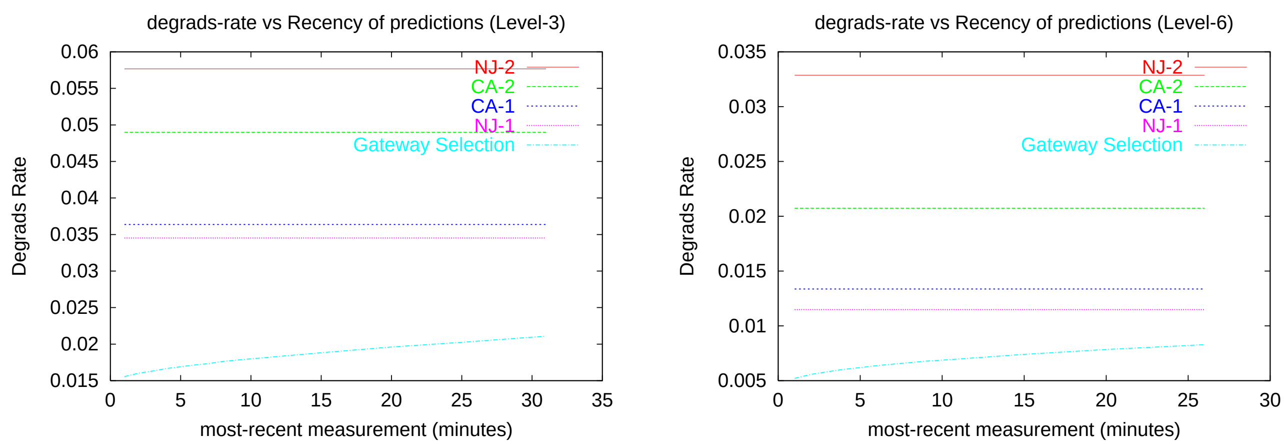


Fig. 16. Performance of gateway selection (using $\text{ExpDecay}_{0.95}$ predictor) as a function of recency of measurements.

.2 Improving median RTT

So far we considered gateway selection potential for avoiding degradations. Since different gateways often have different IP paths to the destination, selection can also be used to reduce the median RTT. We observed earlier that the median RTT is fairly static, and thus, the median can be determined using fewer and less-recent measurements than what we used for avoiding degradations. The selection itself can also be static, at least within a time-frame of days, with each IP address being mapped to the gateway that obtains the smallest median RTT value for it.

We evaluated the performance of median-based selection on the CA-1 and NJ-1 gateways. These two gateways have a T1 line connecting them with typical Round Trip Time of 62ms. For each of the 35,124 IP addresses, and each gateway, we computed the median value of hourly measurements taken over a period of 7 days. The IP-addresses were weighted by the corresponding number of requests, and a weighted fraction of 1.7% of the addresses had "disconnects" median from at least one of the gateways.

Interestingly, the difference between the medians of the two location, shown in Figure 17, often exceeded the RTT we observed on the T1 line (the triangle inequality clearly does not hold for IP routes). This phenomenon of the triangle inequality not holding for IP-routes, was observed earlier [13], [3].

These results can be used to estimate the potential benefit of median-based gateway selection. Table IV sum-

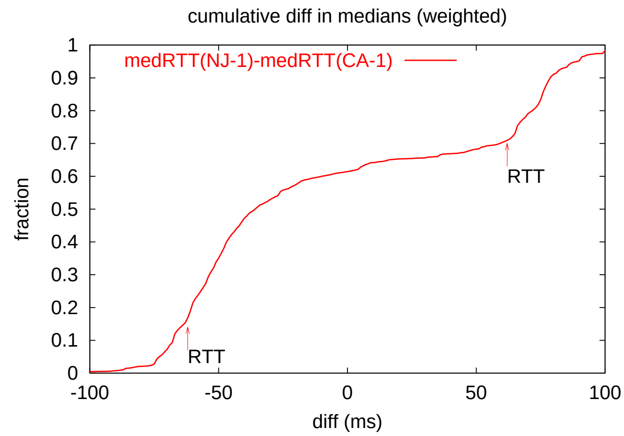


Fig. 17. Difference in median between NJ-1 gateway and CA-1 gateway

gateway	no selection	selection	reduction
NJ-1	83.5ms	78.5ms	6%
CA-1	90.0ms	88.2ms	2%

TABLE IV

POTENTIAL REDUCTION IN TOTAL LATENCY USING MEDIAN-BASED STATIC GATEWAY SELECTION

marizes the potential reduction in latency using gateway selection across all IP addresses such that the median measurement was "reachable" from both gateways (this accounted for weighted 98.2% of addresses). The total latency values with and without selection are computed assuming all RTT are equal to their median value. The "no selection" column in the table states the average (over HTTP requests) of the median RTT value from the gateway to the destination, when all requests are routed out through the gateway. The "selection" columns shows the same average when for each destination, we use the minimum of the RTT through the gateway, and the T1-line RTT plus the RTT through the other gateway. The "reduction" column shows that reduction in this average when using selection.

IX. CONTROLLING PREDICTION OVERHEAD

An important issue for intelligent routing products is the overhead of measurements. *Active-measurements* products initiate probes (SYN,ping,HTTP request) to the destinations. Active measurements may suffer from scalability issues, since they consume network resources and can potentially annoy network operators of the probed targets [16]. *Passive measurements* are collected on regular traffic, and avoid overhead altogether; they are more effective for popular and recently-used destinations, and can be complemented with active measurements.

We argue that the necessary measurement overhead for good predictions can be balanced against per-destination traffic: Our gateway selection evaluation showed that a considerable part of the benefit can be obtained using less-recent measurements, which can be obtained via passive measurements alone. Since the distribution of requests per destination has large skew, with most traffic being directed to a small set of destinations, active probing of only the more popular destinations would provide most of the benefits of gateway selection. This limited probing also ensures that the amount of measurement probes each destination receives is a fraction of the normal traffic sent to the destination.

Another approach to reduce the number of active measurements while still benefiting from recent measurements is to cluster destinations, so that measurements of one destination can be used to predict another. This approach is further investigated and some preliminary results are provided below.

ner-routes.bbnplanet.net	route-server.cerf.net	route-server.gblx.net
route-server.exodus.net	route-server.as5388.net	route-server.ip.att.net
route-server.colt.net	route-server-eu.exodus.net	route-server.gt.ca
route-server-ap.exodus.net	route-views.oregon-ix.net	144.85.10.248
route-server.cbbtier3.att.net		

TABLE V

ROUTING TABLES USED FOR OBTAINING BGP ATOMS.

A. Clustering destinations

A way to predict degradations using fewer measurements is by clustering destinations. The desirable properties from such a clustering is that it is easy to map a destination to other destinations that tend to experience degradation together.

Clustering of all destination would be effective in reducing the number of probes if (i) the number of clusters is small with respect to the number of destination, and (ii) members of the same clusters share most "by-passable" degradations. We performed some preliminary measurements to determine the feasibility of such clustering.

We consider several natural clustering methods based on information drawn for router BGP table dumps (in the spirit of [17]).

1. AS clusters, where all destinations that belong to the same AS are placed in the same cluster.
2. BGP atoms, where all destinations that belong to the same BGP atom are placed in the same cluster. (The term BGP Atom was coined in [18] and we use the same definition.) To obtain atoms we took snapshots of 13 large BGP routers (see Table V) and downloaded their routing tables. Each of these routers has a route (at the AS level) to each particular destination, two IP addresses are placed in the same BGP atom if and only if they have the same route in each snapshot of each of the routing tables.
3. Routing prefix: We took the union of all routing prefixes that occurred in the snapshots of the routing tables of the routers listed in Table V. We associated with each destination the longest prefix that matches in the union of the prefixes.

The first requirement for feasibility of clustering is addressed in Figure 18, which shows the number of representatives needed to cover a certain fraction of logged HTTP requests. The "Host" line shows the number of representatives one needs to monitor in order to cover a certain fraction of requests if no clustering of addresses is used. The other lines show this tradeoff if one representative is used per cluster when the clustering is performed according to BGP prefix, BGP Atom, and AS, respectively.

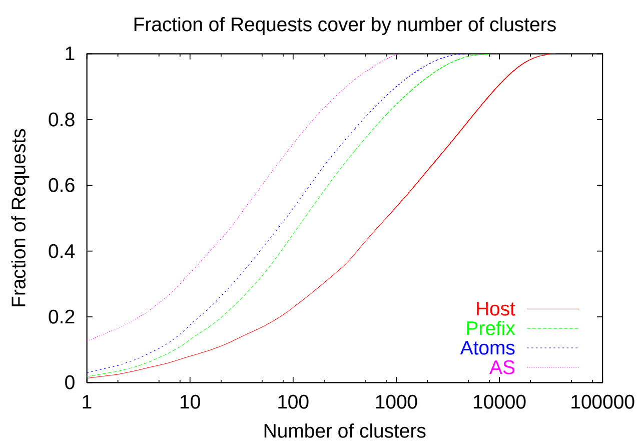


Fig. 18. Number of representatives required to cover fraction of HTTP requests.

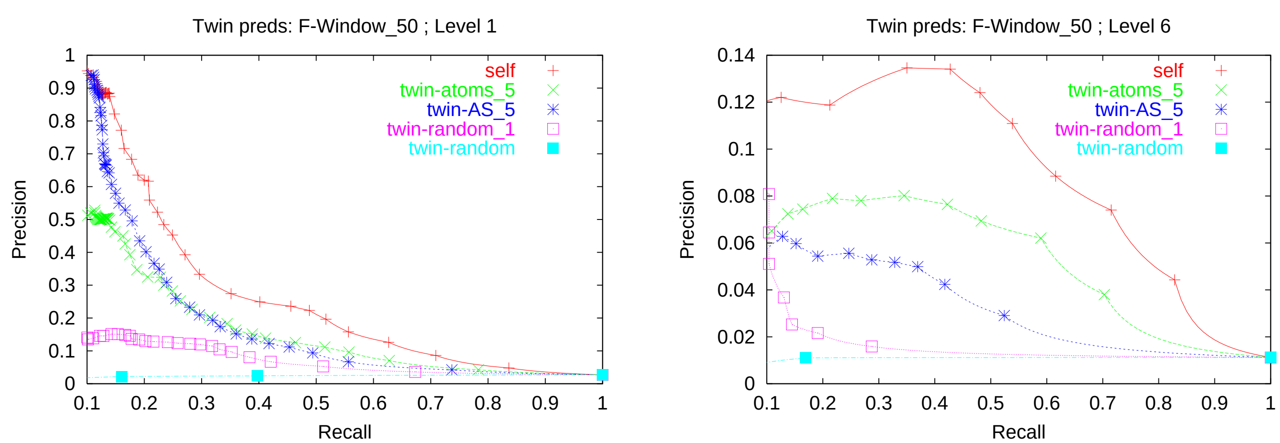
These clusters were further refined using the median of the measurements to each IP address. Where within each cluster, the predictor for each address used only measurements on addresses where the median value was not to far from that of the predicted address. This heuristic was meant to refine the clustering for very large ASes. We then tested the prediction quality when different "representatives" are used for each address.

We picked an initial weighted sample S of 430 IP addresses. From these addresses we focused on those that had at least one representative (other than the address itself) belonging to the same "refined category." For each address, within each refined category, we then sampled one of the appropriate representatives uniformly and random and we refer to it as a *twin* of the address in this refined category We considered the following refined categories (defined for an address A.)

1. twin-atoms-5: Address belonging to the same BGP atom as A with a median RTT that is within 5 milliseconds of the median RTT of A.
2. twin-AS-5: Address belonging to the same AS as A with a median RTT that is within 5 milliseconds of the median RTT of A.
3. twin-random-1: A random address with median within 1 millisecond from the median of A.
4. twin-random: A random address, for control purposes.

The most restrictive category is the twin-atom-5 category. Only about 200 of the 430 IP addresses had a twin in this category. It turns out that all these 200 addresses had twins in other categories. We then conducted measurements on these 200 addresses and all their twins in the 4 different categories. We applied the F-WINDOW$_{50}$ predictor on these measurements, where for each address we used its own historic measurements and the historic measurements of the twin addresses in the different categories. Figure 19 shows the results for level-1 and level-6 degradations. As expected, predictions based on the history of the address itself are the most accurate, the twin-atom-5 predictions come second, and then the twin-AS-5. The "random" twins come last, showing that indeed the BGP twins do capture a significant part of degradations. For example, for a recall value of 0.5 and level-6 degradations, the precision is 11% for the self prediction, degrades to 7% if we use a twin from the same atom, but is only 1% if we use just use a random IP address or even a random IP address with the same median. The gap between 11% (self predictions) and 7% (atom twin predictions) reflects host degradations and degradations very close to the end host.

Note that the precision recall on this figure are much lower than observed earlier. The reason for that is that the measurement period was 4 off-peak hours, and included very few blackouts (the majority of degradations were isolated degradations and the total number of degradations of all addresses in the measurement period was several hundreds). Since we focus only on the 200 addresses of the 430 in the sample that had a twin-atom-5 twin, the sample is also biased.



level 1 level 6

Fig. 19. Quality of twin predictions (NJ-1 gateway; Degradation levels 1 and 6.)

We note that our experiments on using BGP clusters are still preliminary. In the future we intend to use these clusters as another way to pin point location of cause. We would also like to combine BGP clusters with measurements from several gateways and correlate the fraction of degradations that are shared amongst all gateways with the prediction gap between "self" and "atom-twin", and thus learn the true effectiveness of each twin for gateway selection by learning what fraction of "by-passable" degradations are missed by predictions based on the "atom twin."

SUMMARY

The BGP4 protocol, which governs inter-AS routing, remarkably manages to maintain connectivity between large number of Internet hosts. The well-known deficiencies of BGP4 are its suboptimal path selection and inability to cope with temporary link congestion. We show, however, that by and large, these deficiencies can be compensated through local gateway diversity at the source combined with intelligent *gateway selection*.

Specifically, we first learn that a large fraction of end-to-end service degradations are path degradations which can be bypassed if a request is routed through a different AS; we then show that small (but carefully selected) gateway diversity combined with simple prediction algorithms can reduce the degradation-rate to half of that of the best single gateway.

Gateway selection represents an important direction of emerging applications that allow multi-peered ASes and LANs with multiple connections to better choose and utilize expensive link diversity to increase QoS. We find gateway selection to be a viable approach for increasing QoS and develop and evaluate different algorithms. Our methodology is generally applicable for comparing performance of different gateway selection products, and for selecting a combination of gateways that would best balance resilience and cost.

-0.03in

REFERENCES

[1] Y. Rekhter and T. Li, "A border gateway protocol 4 (BGP4)," Tech. Rep., IETF, 1999, draft-ietf-idr-bgp4-09.txt.

[2] C. Labovitz, A. Ahuja, A. Bose, and F. Johanian, "Delayed Internet routing convergence," in *Proc. ACM SIGCOMM 00*, 2000.

[3] S. Savage, A. Collins, and E. Hoffman, "The end-to-end effects of Internet path selection," in *Proc. ACM SIGCOMM 99*, 1999.

[4] "Route science company," http://www.routescience.com/.

[5] "Sockeye networks," http://www.sockeye.com/.

[6] "Opnix inc.," http://www.opnix.com/.

[7] "Netvmg," http://www.netvmg.com/.

[8] "Radware company," http://www.radware.com/.

[9] "Internap company," http://www.internap.com/.

[10] Z. Fei, S. Bhattacharjee, E. W. Zegura, and M. H. Ammar, "A novel server selection technique for improving the response time of a replicated service," in *Proceedings of the IEEE INFOCOM Conference*, 1998.

[11] P. Francis, S. Jamin, C. Jin, Y. Jin, D. Raz, Y. Shavitt, and L. Zhang, "IDMaps: a global Internet host distance estimation service," *IEEE/ACM Transactions on Networking*, vol. 9, pp. 525–540, 2001.

[12] Y. Zhang, N. Duffield, V. Paxson, and S. Shenker, "On the constancy of Internet path properties," in *Proc. ACM-SIGCOMM Internet Measurement Workshop*, 2002.

[13] D. G. Andersen, H. Balakrishnan, M. F. Kaashoek, and R. Morris, "Resilient overlay networks," in *Proc. SOSP 2001*, 2001.

[14] "A Distributed Testbed for National Information Provisioning," `http://www.ircache.net`.

[15] Timothy G. Griffin, "Tutorial: An introduction to interdomain routing and BGP," August 2001.

[16] "Nanog digest 900," http://www.nanog.org/.

[17] J. Wang and B. Krishnamurthy, "On netowrk aware clustering of Web clients," in *Proc. ACM SIGCOMM 00*, 2000.

[18] A. Broido and kc claffy, "Global patterns and efficiency of the routing system," `http://www.caida.org/outreach/isma/0012/talks/kcandre/`.

APPENDIX

I. VW-COVER OUTPUT

Tables VI and VII show two representations of the prediction models constructed by VW-COVER for level-6 degradations on the CA-1 measurements. the table shows only the first 25 pairs (out of total of 90 pairs) produced by the greedy selection, along with the respective (recall, precision) values. Table VI provides the raw representation. In this form, in order to reach the (recall,precision) values of step i, one should use all the windows and the correspond number of degradations of steps $1 \dots i-1$. The second more refined representation is obtained when noticing that many of the windows are overlapping. Hence, the predictions (and same (recall,precision) values) can be obtained via a subsets of previous windows. This reduced set of windows is shown in Table VII. For each step i, we list the correspond reduced set of windows that one should use in order to reach the (recall,precision) values of that step.

Step	Window	Num Degradations	Recall	Precision
1	120	120	0.068	0.996
2	53	53	0.105	0.989
3	27	27	0.152	0.983
4	22	22	0.165	0.982
5	13	13	0.197	0.980
6	12	12	0.200	0.979
7	11	11	0.205	0.978
8	9	9	0.214	0.975
9	8	8	0.220	0.970
10	6	6	0.235	0.957
11	5	5	0.246	0.941
12	4	4	0.261	0.913
13	3	3	0.285	0.871
14	11	10	0.288	0.861
15	2	2	0.330	0.751
16	6	5	0.334	0.738
17	5	4	0.338	0.724
18	4	3	0.346	0.698
19	5	3	0.364	0.636
20	49	45	0.365	0.633
21	8	5	0.368	0.620
22	7	4	0.372	0.608
23	6	3	0.383	0.577
24	7	3	0.401	0.528
25	15	5	0.435	0.456

TABLE VI
RAW REPRESENTATION OF VW-COVER PREDICTION MODEL

Step	Recall	Precision	Window	Num Degradations
1	0.068	0.996	120	120
2	0.105	0.989	53	53
3	0.152	0.983	27	27
4	0.165	0.982	22	22
5	0.197	0.980	13	13
6	0.200	0.979	12	12
7	0.205	0.978	11	11
8	0.214	0.975	9	9
9	0.220	0.970	8	8
10	0.235	0.957	6	6
11	0.246	0.941	5	5
12	0.261	0.913	4	4
13	0.285	0.871	3	3
14	0.288	0.861	3	3
			11	10
15	0.330	0.751	11	10
			2	2
16	0.334	0.738	2	2
			6	5
17	0.338	0.724	2	2
			5	4
17	0.338	0.724	2	2
			5	4
18	0.346	0.698	2	2
			4	3
19	0.364	0.636	2	2
			5	3
20	0.365	0.633	2	2
			5	3
			49	45
21	0.368	0.620	2	2
			5	3
			49	45
			8	5
22	0.372	0.608	2	2
			5	3
			49	45
			7	4
23	0.383	0.577	2	2
			49	45
			6	3
24	0.401	0.528	2	2
			7	3
25	0.435	0.456	2	2
			7	3
			15	5

TABLE VII

REDUCED REPRESENTATION OF VW-COVER PREDICTION MODEL