Detecting ICMP Rate Limiting in the Internet

Hang Guo John Heidemann

USC/Computer Science Dept. and Information Sciences Institute

Abstract. ICMP active probing is the center of many network measurements. Rate limiting to ICMP traffic, if undetected, could distort measurements and create false conclusions. To settle this concern, we look systematically for ICMP rate limiting in the Internet. We create *FADER*, a new algorithm that can identify rate limiting from user-side traces with minimal new measurement traffic. We validate the accuracy of FADER with many different network configurations in testbed experiments and show that it almost always detects rate limiting. With this confidence, we apply our algorithm to a random sample of the whole Internet, showing that *rate limiting exists* but that *for slow probing rates, rate-limiting is very rare*. For our random sample of 40,493 /24 blocks (about 2% of the responsive space), we confirm 6 blocks (0.02%!) see rate limiting at 0.39 packets/s per block. We look at higher rates in public datasets and suggest that fall-off in responses as rates approach 1 packet/s per /24 block is consistent with rate limiting. We also show that even very slow probing (0.0001 packet/s) can encounter rate limiting of NACKs that are concentrated at a single router near the prober.

1 Introduction

Active probing with pings and traceroutes (both often using ICMP echo requests) are often the first tool network operators turn to assess problems and are widely used tools in network research. Studies of Internet address usage [10,4], path performance [13], outages [15,19], carrier-grade NAT deployment [18], DHCP churn [14] and topology [3,12] all depend on ICMP.

An ongoing concern about active probing is that network administrators rate limit ICMP. Administrators may do *traffic policing*, limiting inbound ICMP, and routers often *rate-limit generation of ICMP error messages* (ICMP types 3 and 11, called here ICMP NACKs). However recent work has emphasized probing as quickly as possible. For IPv4 scanning, ISI Internet Censuses (2008) send 1.5k probe/s [10], IRLscanner (2010) sends 22.1k probe/s [11], Trinocular (2013) sends 20k probes/s [15], ZMap (2013) sends 1.44M probes/s [5], or 14M probes/s in their latest revision [2], and Yarrp (2016) sends 100k probes/s or more [3]. Interest in faster probing makes rate limit detection a necessary part of measurement, since undetected rate limiting can silently distort results.

Although rate limiting is a concern to active probing, we know only two prior studies that explicitly look for rate limiting in the general Internet [17,6]. Both of their mechanisms are expensive (requiring hundreds of vantage points or server-side traffic of Google's CDN) and neither of them look at rate limiting to ICMP echo requests in forward path (§6). Unlike this prior work, we want to study forward-path ICMP rate limiting in global scale without intensive traffic probing or extensive sever-side data.

Our first contribution is to provide FADER (Frequent Alternation Availability Difference ratE limit detector), a new lightweight algorithm to detect and estimate forward-path ICMP rate limit across the Internet. Our approach works from a single vantage point, and requires two scans at different rates, detecting rate limits that take any value between those rates.

Our second contribution is to re-examine two existing public datasets for signs of ICMP rate limiting in the whole Internet. First, we use random samples of about 40k /24 blocks to show that *ICMP Rate limiting is very rare in the general Internet* for rates up to 0.39 packets/s per /24: only about 1 in 10,000 /24 blocks are rate limited. Second, we look at higher rate scans (up to 0.97 packets/s per /24) and show the response fall-off in higher rates is consistent with rate limiting from 0.28 to 0.97 packets/s per /24 in parts of the Internet.

Finally, although low-rate scans do not usually trigger rate limiting, we show that rate limiting explains results for error replies when Internet censuses cover non-routed address space.

2 Modeling Rate Limited Blocks

Our detection algorithm uses models of rate limiting in commercial routers.

2.1 Rate Limit Implementations in Commercial Routers

We examined Cisco and Juniper router manuals and two router models (Cisco ME3600-A and Cisco 7204VXR); most routers implement ICMP rate limiting with some variation on a token bucket.

With a token bucket, tokens accumulate in a "bucket" of size B tokens at a rate of L tokens/s. When a packet arrives, it consumes one token and is forwarded, or the packet is discarded if the token bucket is empty (assuming 1 token per packet). Ideally (assuming smooth traffic), for incoming traffic of P packets/s, if $P < L$, the traffic is below rate limit and will be passed without loss. When $P > L$, initially all packets will be passed as the bucket drains, then packet loss and transmission will alternate as packets and tokens arrive and are consumed. In the long run, when $P > L$, egress traffic exits at rate L packets/s.

We only model steady-state behavior of the token bucket because our active probing (§4) lasts long enough (2 weeks, 1800 iterations) to avoid disturbance from transient conditions.

2.2 Modeling Availability

We first model *availability* of a rate limited block—the fraction of IPs that respond positively to probing. We consider both the *true* availability (A), ignoring rate limiting, and also the *observed* availability ($\hat{A}$) affected by rate limiting.

Two observations help model availability. From §2.1, recall that L packet/s pass when P packet/s enter token bucket. Therefore L/P is the proportion of probes that pass. Second, if N IPs in target block are responsive, a non-rate-limited ping hits a responsive IP with probability N/n_B (n_B represent number of IP in a /24 block: 256). Combining above two observations gives us Equation 1.

$$A = \frac{N}{n_B} \quad \text{and} \quad \hat{A} = \begin{cases} A(L/P), & \text{if } P > L \\ A, & \text{otherwise} \end{cases} \tag{1}$$

$$R = \frac{N}{n_B}P \quad \text{and} \quad \hat{R} = \begin{cases} R(L/P), & \text{if } P > L \\ R, & \text{otherwise} \end{cases} \tag{2}$$

2.3 Modeling Response Rate

Response rate is the positive responses we receive from target block per second. In our model (Equation 2), we consider both the *true* value (R), ignoring rate limit, and the *observed* value ($\hat{R}$), affected by rate limit.

2.4 Modeling Alternation Count

Response Alternation is defined as the transition of an address from responsive to non-responsive or the other way around. Rate limits cause *frequent alternation* between periods of packet response and drops as the token bucket fills and drains. Frequent alternation helps distinguish rate limiting from other sources of packet loss such as networks outages (since outages are long-lived). Frequent alternation is, however, less effective in distinguishing rate limiting from transient network congestion because congestion losses are randomized and create frequent alternation. An additional round of probing ensures the detection results are robust against transient network congestion.

We model the count of observed response alternations, $\hat{C}$, both accurately and approximately. The accurate model (in our technical report [9] due to space) fits measured values precisely but are not computable because our data has $r = 1800$ iterations and the number of states scales as 2^r. The approximate model (Equation 3) provides single expression covering all r but fits only when $P \gg L$ (so that consecutive packets from same sender are never passed by token bucket). We use it in our evaluation since it is feasible to solve when $r = 1800$.

$$\hat{C} = 2(L/P)Nr, \quad \text{when } P \gg L \qquad (3)$$

$$\hat{L} = \frac{n_B \hat{A}_H P_H}{\hat{N}_L} \qquad (4)$$

3 Detecting Rate Limited Blocks

The models (§2) assist our detection algorithm.

3.1 Input for Detection

Our detection algorithm requires *low-* and *high-rate* measurements as input.

Low-rate measurements must be slower than any rate limit that are detected. Fortunately the routers we study have minimal values for rates, and we believe our low-rate, at 0.0001 pings/s per block, is below that in most cases (§4.4 describes one exception). Low-rate captures the true availability (A) of target blocks.

High-rate measurements must exceed the target rate limit. It sets the upper bound for FADER's detection range. In addition, high-rate measurements must be repeated to use our alternation detection algorithm (Algorithm 1). Validation in §5.1 shows that 6 repetitions is sufficient but our data include 1800 repetitions.

Both low- and high-rate measurements need to last a multiple of 24 hours to account for regular diurnal variations in address usage [16].

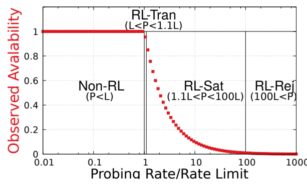


Fig. 1: Four Phases of ICMP Rate Limiting

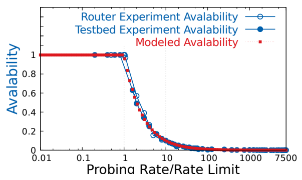


Fig. 2: Availability Model Validation Results

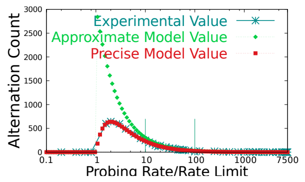


Fig. 3: Alternation Count Model Validation Results

Algorithm 1: Frequent Alternation Test

```
Input:
    $\hat{C}$: observed response alternation count in fast scan
    $r$: number of probing rounds in fast scan
    $\hat{N_L}$: responsive IP count observed in slow scan
    $\hat{N_H}$: responsive IP count observed in each round of fast scan (where responsive IPs observed at $i$th round is $\hat{N_{H_i}}$)
Output:
    $O_{fat}$: results of frequent alternation test

1:  if $\hat{C} > (2\hat{N_L}r)/T_{rej}$ and NotDirTmpDn($\hat{N_H}, \hat{N_L}, r$) then
2:      $O_{fat} \leftarrow$ Passed   // has freq alternations
3:  else
4:      $O_{fat} \leftarrow$ Failed   // no freq alternations
5:  end if

6:  function NotDirTmpDn($\hat{N_H}, \hat{N_L}, r$)
7:      for $i = 1$ to $r$ do
8:          if $\hat{N_{H_i}} \geq \hat{N_L}$ then
9:              return $false$
10:         end if
11:     end for
12:     return $true$
13: end function
```

Algorithm 2: FADER

```
Input:
    $\hat{A}_L / \hat{A}_H$: measured block availability in slow/fast scan
    $\hat{N_L}$: responsive IP count in slow scan
    $T_{rej}$: lower bound of RL-Rej phase
    $O_{fat}$: result of frequent alternation test
Output:
    $O_{fader}$: detection result of FADER
1:  if $\hat{A}_L = 0$ or $\hat{A}_H = 0$ or $\hat{N_L} < 10$ then// blk down
2:      $O_{fader} \leftarrow$ Can-Not-Tell
3:  else if $(\hat{A}_L - \hat{A}_H)/\hat{A}_L > 0.1$ then// significant $\hat{A}$ drop in faster probing
4:      if $\hat{A}_H/\hat{A}_L < 1/T_{rej}$ then // in RL-Rej
5:          $O_{fader} \leftarrow$ Can-Not-Tell
6:      else
7:          if $O_{fat}$ = Passed then
8:              $O_{fader} \leftarrow$ Rate-Limited
9:          else // no freq alternations
10:             $O_{fader} \leftarrow$ Can-Not-Tell
11:         end if
12:     end if
13: else// no significant $\hat{A}$ drop in faster probing
14:     $O_{fader} \leftarrow$ Not-Rate-Limited
15: end if
```

3.2 Four Phases of ICMP Rate Limiting

The models from §2 allow us to classify the effects of ICMP rate limiting into four phases (Figure 1). These phases guide our detection algorithm:

1. Non-RL ($P < L$): before rate limiting takes effect,
2. RL-Tran ($L < P < 1.1L$): rate limiting begins to reduce $\hat{A}$ with alternating responses.
3. RL-Sat ($1.1L < P < 100L$): significant $\hat{A}$ drop and frequent alternation.
4. RL-Rej ($P > T_{rej}L$, $T_{rej} = 100$): most packets are dropped ($\hat{A} < 0.01N/n_B$) and response alternations are rare.

These phases also identify regions where no algorithm can work: rate limits right at the probing rate (RL-Tran phase, due to not enough change in response), or far above it (RL-Rej phase, because the block appears completely non-responsive, giving little information). We use empirical thresholds $1.1L$ and $100L$ to define these two cases.

In §5.2 we show that our algorithm is correct in the remaining large regions (Non-RL and RL-Sat), provided $P < 60L$.

3.3 Detecting Rate Limited Blocks

FADER is inspired by observations that the RL-Tran phase is narrow, but we can can easily tell the difference between the Non-RL and RL-Sat phases. Instead of trying to probe at many rates, we probe at a slow and fast rate, with the hope that the slow probes observe the Non-RL phase and the high-rate brackets the RL-Tran phase. If the target block shows much higher availability in slow probing, we consider the block a rate limit *candidate* and check its traffic pattern for signs of rate limiting: consistent and randomized packet dropping and passing.

We first introduce *Frequent Alternation Test* (Algorithm 1). This subroutine identifies the consistent and randomized packet dropping caused by rate limiting (by looking for large number of responses alternations).

Threshold $(2\hat{N_L}r)/T_{rej}$ is derived from our approximate alternation count model (Equation 3). As low-rate measurement is assumed non-rate-limited, we have $\hat{N_L}$ (responsive IPs count in low-rate measurement) $= N$ (responsive IP count when non-rate-limited). Recall we give up detection in RL-Rej phase (§3.2), we have $P < T_{rej}L$. Substituting both into alternation count model, for a rate limited block, there must be at least $(2\hat{N_L}r)/T_{rej}$ response alternations.

Function NotDirTmpDn filters out diurnal and temporarily down blocks, which otherwise may become false positives because their addresses also alternate between responsive and non-responsive. NotDirTmpDn checks if any round of the high-rate measurement looks like the daytime (active period) of diurnal block or the up-time of temporarily down blocks, satisfying $\hat{N_{H_i}} \geq \hat{N_L}$.

Next, we describe our detection algorithm FADER (Algorithm 2). FADER detects if target block is rate-limited, producing "cannot tell" for blocks that are non-responsive or respond too little. No active measurement system can judge the status of non-responsive blocks; mark such block as cannot-tell rather than misclassifying them as rate limited or not.

In our experiments we see cannot-tell rates of 65% when $P = 100L$ and in average only 2.56 IPs respond in each target block (§5.2); these rates reflect the fundamental limit of any active probing algorithm rather than a limit specific to our algorithm.

Threshold $\hat{N_L} < 10$ used in line 1 is empirical, but chosen because very sparse blocks provide too little information. Test $(\hat{A}_L - \hat{A}_H)/\hat{A}_L > 0.1$ (line 3) is derived by substituting $P > 1.1L$, the lower bound of RL-Sat phase (where we start to detect rate limit), into availability model (Equation 1). Test $\hat{A}_H/\hat{A}_L < 1/T_{rej}$ (line 4) is derived by substituting $P > T_{rej}L$ (RL-Rej phase, where we give up detection), into availability model (Equation 1).

Once a target block is detected as rate limited, we estimate its rate limit $(\hat{L})$ by Equation 4 which is derived by inverting our availability model (§2.2). (We estimates the effective rate limit at each target /24 block, or the aggregate rate limit of intermediate routers across their covered space. We do not try to differentiate between individual hosts in a /24 block because two scans provide too few information about host).

4 Results: Rate Limiting in the Wild

We next apply FADER to existing public Internet scan datasets to learn about ICMP rate limiting in the Internet. (We validate the algorithm later in §5.)

Start Date (Duration)	**Size** (/24 blocks)	**Alias**	**Full Name**
2016-08-03 (32 days)	14,460,160	it71w census	`internet_address_census_it71w-20160803`
2016-08-03 (14 days)	40,493	it71w survey	`internet_address_survey_reprobing_it71w-20160803`
2016-06-02 (32 days)	14,476,544	it70w census	`internet_address_census_it70w-20160602`
2016-06-02 (14 days)	40,493	it70w survey	`internet_address_survey_reprobing_it70w-20160602`

Table 1: Datasets Used in This Paper

blocks studied	40,493	(100%)	
not-rate limited	24,414	(60%)	
cannot tell	15,941	(39%)	
rate limited	111	(0.27%)	*(100%)*
false positives	105	(0.25%)	*(95%)*
true positives	6	(0.015%)	*(5%)*

Table 2: it71w Detection Results

	Number of Blocks (Ratio)		
Test Name	Input	Passed	Filtered
Availability Diff	40,403	2,088 (5.2%)	38,315 (94.8%)
Freq Alternation	2,088	111 (5.3%)	1,977 (94.7%)
Re-probing	111	5 (4.5%)	106 (95.5%)

Table 3: Effects of Each FADER Step

4.1 How Many Blocks are Rate Limited in the Internet?

We first apply FADER to find rate limited blocks in the Internet, confirming what we find with additional probing.

Input data: We use existing Internet censuses and surveys as test data [10]. Reusing existing data places less stress on other networks and allows us to confirm our results at different times (§4.2). Table 1 lists the public datasets we use [8].

Censuses (0.0001 pings/s per block) and surveys (0.39 pings/s per block) define the low- and high-rates that bound rate limits detected by our algorithm. We could re-run FADER with higher rates to test other upper bounds; we report on existing higher rate scans in §4.3.

Surveys probe about 40k blocks about 1800 times over two weeks, supporting frequent alternation detection. Censuses cover almost the entire unicast IPv4 Internet, but we use only the part that overlaps the survey. With a 2% of the responsive IPv4 address space, randomly chosen, our data provides a representative of the Internet.

Initial Results: Here we apply FADER to it71w, the latest census and survey datasets, in Table 2. We find that *most blocks are not rate limited* (60%), while a good number (39%) are "cannot tell", usually because they are barely responsive and provide little information for detection (without additional information, no one could tell if these blocks are rate limited or not). However, *our algorithm classifies a few blocks* (111 blocks, 0.27%) as apparently rate limited.

Validation with additional probing: To confirm our results, we next re-examine these likely rate-limited blocks, We re-probe each block, varying probing rates from 0.01 to 20 ping/s per block to confirm the actual rate limiting. Our additional probing is relatively soon (one month) after our overall scan.

Figure 4 shows this confirmation process for one example block. Others are similar. In this graph, red squares show modeled availability assuming the block is rate limited (given the rate limit estimation from FADER in Table 4). The green line with diamonds shows the availability if the block is not rate limited. As Figure 4 shows, this block's measured availability (blue dots) tightly matches the modeled value with rate limiting while diverging from values without rate limiting. We also apply similar confirmation process to this block's measured response rate (omitted, but details in our technical report [9]). These data show that this block, 182.237.200.0/24, is rate limited.

/24 Block	Response Rate (measured, pkts/s)	Availability ($\hat{A}_L$, %)	Rate Limit (ping/s per blk) (measured)	(estimated)
124.46.219.0	0.009	9.77	0.09	0.09
124.46.239.0	0.08	53.13	0.15	0.12
182.237.200.0	0.06	58.98	0.10	0.12
182.237.212.0	0.04	27.34	0.15	0.10
182.237.217.0	0.06	49.61	0.12	0.13
202.120.61.0	0.35	17.58	1.99	0.32

Table 4: True Rate Limited Blocks in it71w Census and Survey.

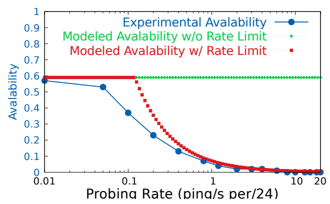
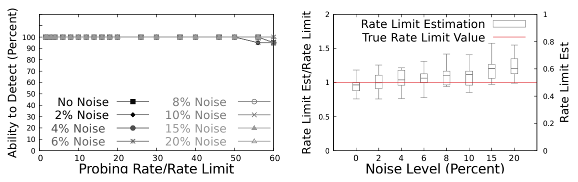


Fig. 4: Confirming Block 182.237.200/24 is Rate Limited with Additional Probing.



(a) Detection Correctness (b) Rate Limit Estimation ($P = 26L$)

Fig. 5: FADER Validation: with Packet Loss

Although this example shows a positive confirmation, we find that most of the 111 blocks are false positives (their availabilities and response rates in re-probing do not match rate limit models). Only the 6 blocks listed in Table 4 are indeed rate limited. We design our algorithm to favor false positives for two reasons. First, favoring false positives (by using necessary conditions as detection signals) avoids missing rate-limited blocks (false negatives). Second, this trade-off (favoring false positives over false negatives) is required to confirm the near-absence of rate limiting we observe. We rule out the possibility that these false positives are caused by concurrent high-rate ICMP activities at our target blocks by observing over long duration and at different times (§4.2).

We use additional verification to confirm true positives. Among the 6 rate limited blocks, 5 belong to the same ISP: Keumgang Cable Network in South Korea, while the last block is from Shanghai Jiaotong University in China. We have contacted both ISPs to confirm our findings, but they did not reply.

Our first conclusion from this result is *there are ICMP rate-limited blocks, but they are very rare*. We find only 6 blocks in 40k, less than 0.02%. Thus it is almost always safe to probe in this range (up to 0.39 packets/s per block).

Second, we see that each of FADER's steps rule out about 95% of all the blocks entering that rule (as in Table 3). However, even after two phases of filtering, there is still a fairly high false positive rate in the remaining blocks, since only 6 of 111 (5.4%) are finally confirmed as rate limited.

Finally, we show that *when* we detect rate limiting, our *estimate of the rate limit* are correct in general. Table 4 shows this accuracy: five out of six rate limits observed in re-probing (which is estimated by measuring $\hat{R}$, $\hat{A}_L$ and inverting our response-rate model Equation 2) closely match FADER's estimates.

However our rate limit estimation (0.32 ping/s per block) for block 202.120.61/24 is 5 times smaller than the rate limit (1.99 pings/s per block) observed in re-probing. When we review the raw data, we believe the rate limit for this block *changed* between our measurements.

4.2 Verifying Results Hold Over Time

To verify our approach works on other datasets, we also apply FADER to it70w census and survey data. This data is taken two months before it71w and sharing 76% of the same target blocks. Detection results of it70w data agrees with our previous conclusion, resulting in about the same number of blocks identified as rate limited (0.3%, 138 of 40,493), and the same fraction as actually limited (0.012%, 5). Of blocks that we confirm as rate limited after re-probing, four also are detected and confirmed in it71w. The fifth, 213.103.246.0/24, is from ISP Swipnet of Republic of Lithuania and is not probed in it71w.

We observe inconsistencies between it70w and it71w for two blocks: 124.46.219.0/24 and 202.120.61.0/24 (detected as rate-limited in it71w, but as Can-Not-Tell and Not-Rate-Limited respectively in it70w) We believe the former block is hard to measure: with only 25 (9.8%) responsive addresses, and the latter actually changed its use between the measurements (supporting details in our technical report [9]).

4.3 Is Faster Probing Rate Limited?

Having shown that rate-limited blocks are very rare up to 0.39 packets/s, we next evaluate if *faster* probing shows signs of rate limiting, as advocated by ZMap [2] and Yarrp [3].

We study Zippier ZMap's 50-second TCP-SYN probing datasets (private dataset obtained from the authors [1], the public ZMap datasets are lower rates), from 0.1M to 14M packet/s, which we estimate as 0.007 to 0.97 packets/s per /24 block. We show rate limiting could explain the response drop-off at higher rates. Although both our models and FADER are originally designed for ICMP rate limiting, they also detect TCP-SYN rate limiting because they detect the actions of the underlying token bucket.

ZMap performs a series of 50-second experiments from 0.1M to 14M packets/s [2]. Each experiment targets a different random sample of a 3.7 billion IP pool. Their results show overall availability (the fraction of positive responses of all hosts that are probed) is roughly stable up to 4M packets/s. However, when probing rates exceed 4M packets/s, the availability starts to decline linearly (the blue dots in Figure 6, from their paper [2]). They state that they do not know the exact reason for this decline.

We believe rate limiting explains this drop—once rate limits are exceeded, as the packet rate increases, availability drops. We also believe that there are roughly the same amount of rate limiting at each packet rate between 4M and 14M packets/s in the Internet, causing the overall availability drop to be linear.

We would like to apply FADER directly to Zippier ZMap's 50-second probing results. Unfortunately we cannot because the target IPs are not known for each run (they do not preserve the seed, so we do not know addresses that do not respond), and they do not repeat addresses, so we cannot test response alternation. (We chose not to collect new, high-rate ZMap data to avoid stressing target networks.) However, we can statistically estimate how many addresses do no not respond, allowing us to evaluate rate-limiting for high-rate scans (up to 14M packets/s).

We create a model of their measurement process and show rate limiting can explain their drops in response rate. Full details of this model are in our technical report [9]. We show availability of many ZMap target blocks matches our expectation of rate limiting by statistically estimating the number of IPs probed in each target block.

Potential Limiting at High Probe Rates: Figure 6 compares our model of ZMap measurement process (the red squares) against reported Zippier ZMap experiments (blue circles). Observing that rate limits are consistent with the drops in response of ZMap at high speeds, we next apply FADER (without the frequent alternation test) to ZMap data, looking for blocks that appear to be rate limited.

We statistically estimate the number of IPs probed in each block. Recall each 50-second scan send pseudo-random probes into the same 3.7 billion IPv4 pool, assuming uniform sampling, about same number of IP will be sampled from each /16 block in the pool. (Here we look at /16 blocks instead of /24 blocks because larger blocks decrease the statistical variance.) As a consequence, for a 50-second ZMap scan of P packets/s, approximately $50P/(3.7 \times 10^9) \times 2^{16}$ IPs are probed in each /16 block, given $50P/(3.7 \times 10^9)$ as the fraction of addresses probed in 50s, against a target 2^{16} addresses in size. We then estimate availability of each /16 block as the fraction of target IPs that respond positively to probes.

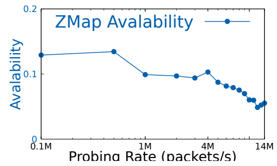
We next apply FADER to detect rate limiting (assuming all blocks pass Frequent Alternation Tests). For each ZMap target block, we use slowest 50-second scan (0.1M packets/s) as the low-rate measurement and test each of the other 15 faster scans as high-rate measurement. This gives us 15 test results (each at a different high rate), for each target block. We consider a block as potentially rate limited if it is detected as rate limited in at least one test. We do not consider the other blocks (cannot tell or not-rate limited) further.

Table 6 shows detection results. Most ZMap target blocks (53,149 blocks, 93.99%) are cannot tell in all 15 FADER tests (43,067 of them due to target block went dark during low-rate measurement and provide no information for detection). A good number of them (3,090 blocks, 5.46%) are classified as rate-limited in at least one FADER test and are considered potentially rate-limited. It is worth noting that most (69.68%) of these potentially rate-limited blocks are consistently classified as rate-limited in most FADER tests (at least 13 out of 16 tests), supporting our claim that those blocks are potentially rate-limited.

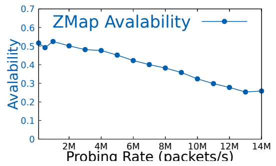
Since we omit Frequent Alternation Tests and our algorithm is optimized to avoid false negatives, we know many of these potential rate limited blocks may be false positives. To further filter out false detection,we manually check a 565 (1%) random sample of 56,500 ZMap target blocks. Of these sample blocks, 31 are detected as rate-limited in at least one FADER test and are considered potentially rate-limited.

We find the other 534 blocks (cannot tell or not-rate limited) to be true negative. They either have almost zero $\hat{A_L}$ or $\hat{A_H}$ (providing no information for detection) or become more available at higher probing rate (opposing our expectation of reduced availability at faster scan)

All 31 potential rate-limited blocks show reduced availability at higher probing rates (regardless of jitters caused by probing noises and distortions introduced by our statistical estimation), matching our expectation of rate limited blocks. We also find 7 of them appear to have more than one rate limits. For



(a) 125.182/16 in Log



(b) 50.62/16 in Linear

Table 5: 2 ZMap Blocks Showing Multiple Rate Limits

blocks studied	56,550	(100%)	
0 rate limited	53,460	(94.54%)	
$\geq$ 1 rate limited	3,090	(5.46%)	*(100%)*
$\geq$ 13 rate limited	2,153	(3.81%)	*(69.68%)*
$<$ 13 rate limited	937	(1.66%)	*(30.32%)*

Table 6: Applying 15 FADER Tests to ZMap /16 Blocks

example, block 125.182/16 in 5a looks like a superposition of $\hat{A}$ curves of two rate limits: one at 0.5M packets/s, the other at 4M packets/s (recall the ideal $\hat{A}$ curve of rate limited block in Figure 1). Block 50.62/16 in 5b, on the other hand, show nearly linear drops in availability as probing rates get higher, suggesting it consists of multiple rate limits (reasons are similar as in our modeling of ZMap experiments). We manually check each /24 blocks in those two /16 blocks, and it appears that those /24 blocks indeed have multiple rate limits. This observation supports our claim that different parts of the /16 have different rate limits.

4.4 Rate Limiting of Response Errors at Nearby Routers

Having shown that probing rates up to 0.39 pings/s trigger rate limits on almost no *target* blocks, we also observe a case where even slow probing (0.0001 ping/s per block) can trigger rate limits in reverse path because *traffic to many targets is aggregated at a nearby router*. Our technical report gives details of this case [9].

5 Validation

We validate our model against real-world routers and our testbed, and our algorithm with testbed experiments. (We also tried to contact two ISPs about ground truth, but we got no response §4.1)

5.1 Does the Model Match Real-World Implementations?

We next validate our models for availability, response alternation, and response rate of rate-limited blocks. We show they match the ICMP rate limiting implementations in two carrier-grade, commercial routers and our testbed.

Our experiments use two commercial routers (Cisco ME3600-A and Cisco 7204VXR) and one Linux box loaded with Linux filter iptables as rate limiters. Our measurement target is a fully responsive /16 block, simulated by one Linux box loaded with our customized Linux kernel [7]. In each experiment, we run a 6-round active ICMP probing, with the rate changing from below the limit to at most 7500× the rate limit (while fixing rate limit).

We begin with validating our *availability model* from Equation 1. Figure 2 shows model predicted availability (red line with squares) closely matches router experiments (blue line with circles) and testbed experiments (blue line with dots) from below to above the rate limit.

We validate our *response rate model* from Equation 2. We omit this data due to space limitations, but our response rate model is accurate from a response rate of 0.01 to 90× the rate limit.

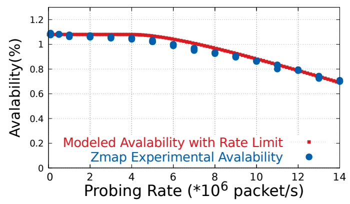
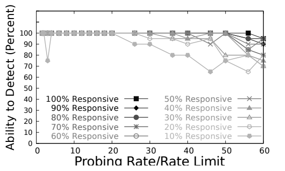
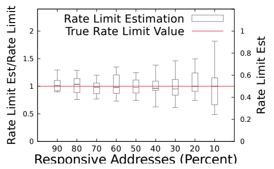


Fig. 6: Modeled Availability (Red) Matches ZMap Probing Results (Blue)



(a) Detection Correctness



(b) Rate Limit Estimation ($P = 26L$)

Fig. 7: FADER Validation: with Partially Responsive Target Blocks

We next validate our models of alternation counts (Equation 3) with testbed experiments. Figure 3 shows our precise model (red square) fits perfectly from below the rate limit up to 7500× the rate limit while our approximate model (green diamond) fits when $P \gg L$ (in our case when $P > 10L$).

We are unable to validate alternation count model with commercial routers; the routers are only available for a limited time. But we believe testbed validations shows the correctness of our alternation counts models since we have already shown rate limiting in testbed matches that of two commercial routers.

5.2 Correctness in Noise-Free Testbed

We next test the correctness of FADER in a testbed without noise (supporting graphs in our technical reports [9]). For noise-free experiment, we run high-rate probing from $1.6L$ to $240L$ stressing FADER beyond its designed detecting range $P < 60L$. FADER detection is perfect for $P < 60L$. However, as we exceed FADER's design limit ($60L$), it starts marking blocks as can-not-tell. The fraction of can-not-tell rises as P grows from $60L$ to $144L$ (when $P = 100L$, 65% blocks are marked as can-not-tell). Fortunately, even when the design limit is exceeded, FADER is *never incorrect* (it never gives a false positive or false negative), it just refuses to answer (returning can-not-tell).

In addition to detecting rate limiting, FADER gives an estimate of what that rate limit is. Varying P from L to $144L$, FADER's rate limit estimate is within 7% (from -4.2% to $+6.9\%$) when $P < 60L$, and it drops gradually as the design limit is exceed.

5.3 Correctness in the Face of Packet Loss

We next consider FADER with packet loss which could be confused with loss due to rate limiting. We vary the amount of random packet loss from 0 to 60%.

Figure 5a shows FADER's detection as packet loss increases. There is almost no misdetection until probe rates become very high. At the design limit of $P = 60L$, we see only about 4% of trials are reported as cannot tell.

While ability to detect is insensitive to noise, our estimate of the rate limit is somewhat less robust. Figure 5b shows that packet loss affects our estimate of the value of the rate limit (here we fix $P = 26L$, but we see similar results for other probe rates). Error in our rate limit is about equal to the dropping rate (at 20% loss rates, the median estimate of rate limit is 20.72% high).

5.4 Correctness with Partially Responsive Blocks

We next consider what happens when blocks are only partially responsive. Partially responsive blocks are more difficult for FADER because probes sent to non-responsive addresses are dropped, reducing the signal induced by rate limiting. Here we vary probe rate for different density blocks. (We hold other parameters fixed and so do not add packet loss.)

In Figure 7a we vary the relative probing rate and plot separate lines for each level of block responsiveness. In general, the number of can-not-tell increase as block responsiveness falls, but only when the probe rate is also much greater than the rate limit. In the worst case, with only 10% of IPs responding at a probe rate 60× the rate limit, 35% of tries report can-not-tell and no wrong answer is given.

Figure 7b shows the rate limit output by FADER as the block density changes. We show median and quartiles with box plots, and minimum and maximum with whiskers. The median stays at the true value, but the variance increases, as shown by generally wider boxes and whiskers. Here $P = 26L$; we see similar results at other probing rates.

6 Related Work

Two other groups have studied detecting rate limits in the Internet.

Work from Universite Nice Sophia Antipolis studies rate limiting for traceroutes [17]. Specifically, they study ICMP, Type 11, Time exceeded replies on reverse paths. They detect rate limits by sending TTL-limited ICMP echo requests from 180 vantage points, varying the probing rate from 1 to 4000 ping/s and looking for constant response rates as a sign of rate limits. They studied 850 routers and found 60% to do rate limiting. Our work has several important differences. The overall result is quite different: they find 60% of reverse paths are rate limited in 850 routers, measured up to 4000 ping/s, while we find only 0.02% of forward paths are rate limited in 40k /24 blocks, measured up to 0.39 pings/s per /24.

We believe that both their results and ours are correct. Many routers have reverse-path rate limiting on by default, consistent with their results. Our approach provides much broader coverage and generates less additional traffic by reusing existing data. Our work uses different signals (availability difference and frequent alternation) for detection. Finally, we focus on forward path, so our results apply to address allocation information, while they focus on reverse path, with results that apply to fast traceroutes.

Google recently examined traffic policing, particularly in video traffic [6]. Their analysis uses sampled measurement from hundreds of Google CDNs to millions of users of YouTube. They provide a thorough analysis on the prevalence of policing and the interaction between policing and TCP. They also provide suggestions to both ISP and content providers on how to mitigate negative effect of traffic policing on user experience. Their focus on TCP differs from ours on ICMP rate-limiting. Their coverage is far greater than ours, although that coverage is only possible because Google is a major content provider. They find fairly widespread rate limiting of TCP traffic, but their subject (TCP video) is much faster than ours (ICMP) that such differences in results are not surprising.

7 Conclusion

Undetected rate limiting can silently distort network measurement and bias research results. We have developed FADER, a new, light-weight method to detect ICMP rate limiting. We validated FADER against commercial routers and through sensitivity experiments in a testbed, showing it is very accurate at detecting rate limits when probe traffic is between 1 and 60× the rate limit.

We applied FADER to a large sample of the Internet (40k blocks) on two separate dates. We find that only a tiny fraction (0.02%) of Internet blocks are ICMP rate limited up to 0.39 pings/s per /24. We also examined public high-rate datasets (up to 1 ping/s per /24) and showed their probing results are consistent with rate limitings. We only see significant rate limiting on reverse path when routers near the prober see a large amount of traffic. We conclude that low-rate ICMP measurement (up to 0.39 ping/s per block) are unlikely to be distorted while high-rate measurement (up to 1 ping/s per block) risks being rate limited.

References

1. D. Adrian, Z. Durumeric, G. Singh, and J. A. Halderman. 50-second scans dataset in paper "Zippier ZMap: Internet-Wide Scanning at 10 Gbps", obtained from David Adrian by request.
2. D. Adrian, Z. Durumeric, G. Singh, and J. A. Halderman. Zippier ZMap: Internet-wide scanning at 10 Gbps. In *USENIX Workshop on Offensive Technologies*, 2014.
3. R. Beverly. Yarrp'ing the Internet: Randomized high-speed active topology discovery. In *ACM Internet Measurement Conference*. ACM, Nov. 2016.
4. A. Dainotti, K. Benson, A. King, kc claffy, M. Kallitsis, and E. Glatz. Estimating Internet address space usage through passive measurements. *ACM Computer Communication Review*.
5. Z. Durumeric, E. Wustrow, and J. A. Halderman. ZMap: Fast Internet-wide scanning and its security applications. In *USENIX Security Symposium*, 2013.
6. T. Flach, P. Papageorge, A. Terzis, L. Pedrosa, Y. Cheng, T. Karim, E. Katz-Bassett, and R. Govindan. An Internet-wide analysis of traffic policing. In *ACM SIGCOMM*, 2016.
7. H. Guo. rejwreply: a Linux kernel patch that adds echo-reply to feedback type of iptable reject rule. https://ant.isi.edu/software/rejwreply/index.html.
8. H. Guo and J. Heidemann. Datasets In This Paper. https://ant.isi.edu/datasets/icmp/.
9. H. Guo and J. Heidemann. Detecting ICMP rate limiting in the Internet. Technical Report ISI-TR-717, USC/Information Sciences Institute, May 2017.
10. J. Heidemann, Y. Pradkin, R. Govindan, C. Papadopoulos, G. Bartlett, and J. Bannister. Census and survey of the visible Internet. In *ACM Internet Measurement Conference*, 2008.
11. D. Leonard and D. Loguinov. Demystifying service discovery: Implementing an Internet-wide scanner. In *ACM Internet Measurement Conference*, Nov. 2010.
12. M. Luckie, A. Dhamdhere, B. Huffaker, D. C. rk, and kc claffy. bdrmap: Inference of borders between IP networks. In *ACM Internet Measurement Conference*, 2016.
13. H. V. Madhyastha, T. Isdal, M. P. ek, C. Dixon, T. Anderson, A. Krishnamurthy, and A. V. taramani. iPlane: An information plane for distributed services. In *7th USENIX Symposium on Operating Systems Design and Implementation*, 2006.
14. G. C. M. Moura, C. Gañán, Q. Lone, P. Poursaied, and H. Asghari. How dynamic is the ISPs address space? Towards Internet-wide DHCP churn estimation. In *IFIP Networking Conference*.
15. L. Quan, J. Heidemann, and Y. Pradkin. Trinocular: Understanding Internet reliability through adaptive probing. In *ACM SIGCOMM*, 2013.
16. L. Quan, J. Heidemann, and Y. Pradkin. When the Internet sleeps: Correlating diurnal networks with external factors. In *ACM Internet Measurement Conference*, 2014.
17. R. Ravaioli, G. Urvoy-Keller, and C. Barakat. Characterizing ICMP rate limitation on routers. In *IEEE International Conference on Communications*, 2015.
18. P. Richter, F. Wohlfart, N. Vallina-Rodriguez, M. Allman, R. Bush, A. Feldmann, C. Kreibich, N. Weaver, and V. Paxson. A multi-perspective analysis of carrier-grade NAT deployment. In *ACM Internet Measurement Conference*, 2016.
19. A. Schulman and N. Spring. Pingin' in the rain. In *ACM Internet Measurement Conference*.