Improving Coverage of Internet Outage Detection in Sparse Blocks

Guillermo Baltra[1,2] and John Heidemann[1,2]

[1] University of Southern California, Los Angeles CA 90089, USA
[2] Information Sciences Institute, Marina del Rey CA 90292, USA
{baltra,johnh}@isi.edu

Abstract. There is a growing interest in carefully observing the reliability of the Internet's edge. Outage information can inform our understanding of Internet reliability and planning, and it can help guide operations. Active outage detection methods provide results for more than 3M blocks, and passive methods more than 2M, but both are challenged by *sparse blocks* where few addresses respond or send traffic. We propose a new *Full Block Scanning* (FBS) algorithm to improve coverage for active scanning by providing reliable results for sparse blocks by gathering more information before making a decision. FBS identifies sparse blocks and takes additional time before making decisions about their outages, thereby addressing previous concerns about false outages while preserving strict limits on probe rates. We show that FBS can improve coverage by correcting 1.2M blocks that would otherwise be too sparse to correctly report, and potentially adding 1.7M additional blocks. FBS can be applied retroactively to existing datasets to improve prior coverage and accuracy.

1 Introduction

Internet reliability is of concern to all Internet users, and improving reliability is the goal of industry and governments. Yet government intervention, operational misconfiguration, natural disasters, and even regular weather all cause network outages that affect many. The challenge of measuring outages has prompted a number of approaches, including active measurements of weather-related behavior [15], passive observation of government interference [4], active measurement of most of the IPv4 Internet [12], passive observation from distributed probes [16], analysis of CDN traffic [14], and statistical modeling of background radiation [6].

Broad coverage is an important goal of outage detection systems. Since outages are rare, it is important to look everywhere. Active detection systems report coverage for more than 3M /24 blocks [12], and passive systems using CDN data report coverage for more than 2M blocks [14]. More specialized systems focus coverage on areas with bad weather (ThunderPing [15]), or provide broad, country-level or regional coverage, but perhaps without /24-level granularity inside the regions (CAIDA darknet outage analysis [4] and Chocolatine [6]). Although each of the systems provide broad coverage, each recognizes there are portions of the Internet that it cannot measure because the signal it measures is not strong enough. Systems typically detect and ignore areas where they have

Table 1. Coverage comparison in /24 blocks of different measuring approaches.

	Approach	Coverage
UCSD-NT	darknet	3.2M observed [3]
Akamai	passive/CDN	5.1M observed / 2.3M trackable [14]
ThunderPing	active/addrs	10.8M US IP addresses [11]
Disco	TCP disconnections	10.5k [16]
Trinocular	active/blocks	5.9M responsive / 3.4M trackable [12]

insufficient signal (in Trinocular, blocks with fewer than 15 addresses; in ThunderPing, events with fewer than 100 addresses in its region; the Akamai/MIT system, blocks fewer than 40 active addresses; in Chocolatine, blocks with fewer than 20 active IPs). Setting thresholds too high reduces coverage, yet setting them too low risks false outages from misinterpreting a weak signal.

The first contribution of our paper is two new algorithms: *Full Block Scanning* (FBS), to improve coverage in outage detection with active probing, while retaining accuracy and limits on probing rates (§3.1), and *Lone-Address-Block Recovery* (LABR), to increase coverage by providing partial results blocks with very few active addresses (§3.2). Our insight is to recognize that *sparse blocks* signal outages more weakly than other blocks, and so they require more information to make a decision. We chose to delay decisions until all block addresses (the full block) have been observed, thus gathering more information while maintaining limits on the probing rate. (An alternative we decline is to probe more aggressively.) We evaluate FBS as an extension to Trinocular §4.2, but the concept may apply to other outage detection systems.

Our second contribution is to show that FBS can *increase coverage* in two ways (§4.5). First, it correctly handles 1.2M blocks that would otherwise be too sparse to correctly report. Second, it allows addition of 1.7M sparse blocks that were previously excluded as unmeasurable. Together, coverage for 2017q4 can be 5.7M blocks. Moreover, FBS *improves accuracy* by reducing the number of false outage events seen in sparse blocks (§4.1). We confirm that it addresses most previously reported false outage events (§4.3).

The cost of FBS is reduced temporal precision, since it takes more time to gather more information (assuming we hold the probe rate fixed). We show that this cost is limited (§4.4): FBS is required for about one-fifth of blocks (only sparse blocks, about 22% of all blocks). Timing for non-sparse majority of blocks is unaffected, and 74% of recovered uptime for sparse blocks is within 22 minutes. About 40% of accepted outages in sparse blocks are reported within 33 minutes, and nearly all within 3.3 hours. (Reanalysis of old data shows the same results for non-sparse and recovered uptime, but requires twice the time for accepted outages.) Finally, we examine false uptime by testing against a series of known outages that affected Iraq in February 2017.

All of the datasets used in this paper that we created are available at no cost [17]. Our work was IRB reviewed and identified as non-human subjects research (USC IRB IIR00001648).

2 Challenges to Broad Coverage

Our goal is to detect Internet outages with broad coverage. Table 1 shows coverage of several methods that have been published, showing that active probing methods like Trinocular provide results for about 3.4M /24 blocks [12] and CDN-based passive methods provide good but somewhat less coverage (2.3M blocks for the Akamai/MIT system [14]). Passive methods with network telescopes provide very broad coverage (3.2M blocks [3]), but less spatial precision (for example, for entire countries, but not individual blocks in that country). Combinations of methods will provide better coverage: Trinocular and the Akamai/MIT system have a 1.6M blocks overlap, and unique contributions, each providing 1.9M unique 0.7M, from [14]. However, Akamai/MIT data is not publicly available.

Here we examine how to *improve coverage of active probing systems like Trinocular*. Trinocular gets results for 3.4M blocks, and another 2.5M blocks have some response but are not considered "trackable" since they have too few reliably responding addresses.

Our goal in this paper is to expand coverage by making these previously untrackable blocks trackable. We face two problems: sparse blocks and lone addresses, each described below. In the next section we describe two new algorithms to make these blocks trackable: *Full Block Scanning (FBS)*, which retains spatial precision and limited probing rates, but loses some temporal precision; and Lone Address Block Recovery (LABR), an approach that allows confirmation that lone-address blocks are up, although it cannot definitively identify when they are down.

Other active probing systems that follow the Trinocular algorithms (such as the active part of IODA [1]) might benefit from solutions to these problems. We seek algorithms that can reevaluate existing years of Trinocular data, so we follow Trinocular's use of IPv4 /24-prefix blocks and 11-minute rounds.

2.1 Problem: Sparse Blocks

Sparse blocks limit coverage: active scanning requires responses, so we decline to measure blocks with long-term sparsity, and we see a large number of *false outages* in blocks that are not sparse long-term, but often are temporarily sparse.

Sparse blocks challenge accuracy because of a tension between the amount of probing and likelihood of getting a response. To constrain traffic to each block, and to track millions of blocks, Trinocular limits each block to 15 probes per round. Limited probing can cause false outages in two ways: First, it may fail to reach a definitive belief and mark the block as *unknown*. Alternatively, if the block is usually responsive, a few non-responses may produce a down belief.

As an example, Fig. 1 shows four different levels of sparsity, (each starting 2017-10-06, 2017-10-27, 2017-11-14 and 2017-12-16) as (d) individual address responses to Trinocular probes, and (c) Trinocular state inferences. As the block gets denser, Trinocular improves its inference correctness.

Furthermore, every address in this block has responded in the past. But for the first three periods, only a few *are actually* used, making the block *temporarily sparse*. For precision, we use definitions from [12]: $E(b)$ are the addresses in block b that have *ever* responded, and $A(E(b))$ is the long-term probability that

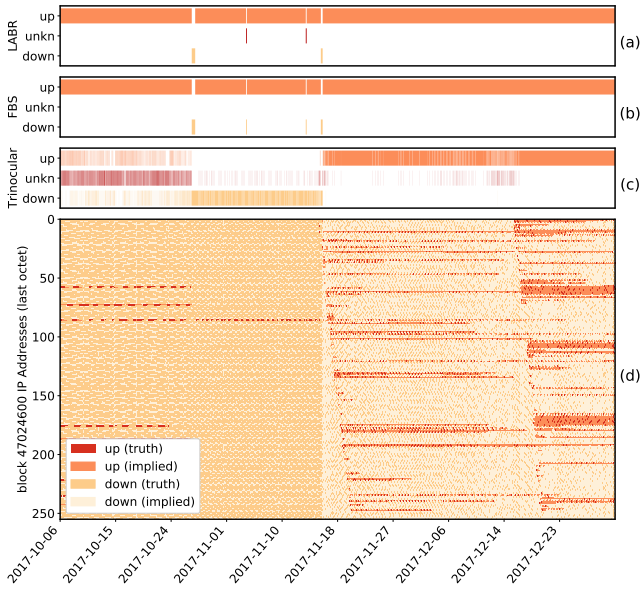

Fig. 1. A sample block over time (columns). The bottom (d) shows individual address as rows, with colored dots when the address responds to Trinocular. Bar (c) shows Trinocular status (up, unknown, and down), bar (b) is Full Block Scanning, and the top bar (a), Lone Address Block Recovery.

these addresses will respond. We also consider a short-term estimate, $\hat{A}(E(b))$. Thus problematic blocks have low $A(E(b))$ or $\hat{A}(E(b))$. We provide further block examples in Appendix A.

Prior systems sought to filter out these sparse blocks, both before and after measurement. Trinocular marks very sparse blocks as *untrackable* (when $A(E(b)) < 0.10$ or $|E(b)| < 15$). It also marked blocks as untrackable when observed A doesn't match predicted A [12], and later used an adaptive estimate for A [13]. Trinocular notes that its unmeasurability test is not strict enough: indeterminate belief can occur when the $A(E(b)) < 0.3$ and $|E(b)| \geq 15$. Accordingly, Richter's use of Trinocular data dropped all blocks with 5 or more outages in 3 months [14], based on our recommendation.

We consider blocks sparse when it is less than a threshold ($\hat{A}_s(E(b)) < T_{sparse}$), where $\hat{A}_s(E(b)$ is a short-term estimate of the *current* availability of the block, and T_{sparse} is a threshold, currently 0.2. Blocks have frequent outages (like Fig. 1) when they are sparse. We find that 80% of blocks with 10 or more down events are sparse, and yet sparse blocks represent only 22% of all blocks (see CDFs in Appendix B).

2.2 Problem: Lone Addresses

The second challenge to coverage are blocks where only one or two addresses are active—we call this problem *lone address blocks*. When a single address is active,

then lack of a response may be a network outage, but it may also be a reboot of a single specific computer or other causes—the implication of non-response from a single address is ambiguous. Trinocular has avoided blocks with few addresses as untrackable (when $|E(b)| < 15$). ThunderPing [15] tracks individual addresses, but recognizing the risk of decisions on single addresses, they typically probe multiple targets per weather event [11].

An example block with a lone-address is in Fig. 1. Of the four phases of use, the second phase, starting 2017-10-27, and for 18 days, only the .85 address replies. Our goal is to handle this block correctly in both of its active states, with many addresses and with a lone address.

3 Improving Outage Detection

3.1 Full Block Scanning for Sparse Blocks

The challenge of evaluating sparse blocks is that Trinocular makes decisions on too little information, forcing a decision after 15 probes, each *Trinocular Round* (*TR*, 11 minutes), even without reaching a definitive belief. We address this problem with *more* information: we consider a *Full Round* (*FR*), combining multiple TRs until all active addresses (all of $E(b)$) have been scanned. This *Full Block Scanning* algorithm makes decisions only on complete information, while retaining the promise of limiting scanning rate.

Formally, a Full Round ends at time t when the minimum N TRs before t that cover all $E(b)$ ever-active addresses of the block: $\sum_{i=t-N}^{t}(|\,TR_i|) \geq |E(b)|$.

Trinocular probes all addresses in $E(b)$ in a pseudo-random sequence that is fixed once per quarter, so we can guarantee each address is probed when we count enough addresses across sequential TRs. (Versions of Trinocular prior to 2020q1 reverse direction at end of sequence, reanalysis of data before this time must sense $2|E(b)|$ addresses to guarantee observing each. We call this retrospective version the 2FR version of FBS, and will use 1FR FBS for new data. They differ in temporal precision, see §4.4.)

Full Block Scanning (FBS) layers over Trinocular outage detection, re-evaluating outages it reports and reverting some decisions. If the block is currently sparse ($\hat{A}_s < T_{sparse}$) and the most recent Full Round included a positive response, then we override the outage. That is, if there are any positive responses in the last Full Round FR_t, we convert any outages to up if $\forall TR_i$ where $i \in [t-N, t]$.

The cost of FBS is that combining multiple TRs loses temporal precision, so we use FBS only when it is required: for blocks that are currently sparse. A block is currently sparse if the short-term running average of the response rate for the block $\hat{A}_s^{3FR}$, computed over the last three *FR*s, is below the sparse threshold ($\hat{A}_s^{3FR} < T_{sparse}$). (We choose three FRs to smooth $\hat{A}$ from multiple estimates.)

The reduction in temporal precision depends on how many addresses are scanned in each TR and the size of FR (that is, $E(b)$). When FBS verifies an outage, we know the block was up at the last positive response, and we know it is down after the full round of non-responses, so an outage could have begun any time in between. We therefore select a start time as the time of the last confirmed down event (the first known lit address, now down). That time has

uncertainty of the difference between the earliest possible start time and the confirmed start time. Theoretically, if all 256 addresses in a block are in use and 15 addresses are scanned each TR, a FR lasts 187 minutes. In practice, timing is often better; we show empirical results in §4.4

3.2 Lone-Address-Block Recovery

The FBS algorithm repairs any block with at least one responsive address in the last FR, allowing us to extend coverage to many sparse blocks. However, when a block has only a single active address, a non-reply may indicate an outage of the network *or* a problem with that single host.

To avoid false down events resulting from non-outage problems with a lone address, we define *Lone-Address-Block Recovery* (LABR). We accept up events, but because outages are rare (much rarer than packet loss), we convert down events to "unknown" for blocks with very few recently active addresses. We define "few" as one or two active addresses, and recently as the last three Full Rounds, so we use LABR when $|\hat{E}^{3FR}| < 3$. We require at least three addresses to avoid making decisions on one or two addresses where packet loss could change results.

This algorithm gives an asymmetric outcome: we can confirm blocks are up, but not that they are down. We believe that outcome is preferable to the alternatives: completely ignoring the block, or tolerating false outages. However, we identify LABR blocks to allow researchers wanting an estimator that can be both up and down to omit them.

4 Evaluation

4.1 Full Block Scanning Reduces Noise

Case Study of One Block Fig. 1 shows one block in CenturyLink (AS209, a U.S. ISP) with outage analysis as a case study.

This block has initially only 8 addresses responding. On 2017-10-27, there is a usage change that causes a down event with no address response for ∼13 hrs. This event is matched in other blocks for the same AS. Then, we see a lone address responding for 18 days. On 2017-11-14, the block starts receiving new users, and once again starting 2017-12-17. On 2017-11-16, it shows a partial outage that is observed only from our Los Angeles site, not from other Trinocular sites.

Trinocular results ((b), third bar) show frequent unknown states that result in false down events, particularly when block usage is sparse in October and early November.

By contrast, Full Block Scanning ((b), the second graph), resolves this uncertainty. FBS' more information confirms the block is usually up, while recognizing the usage change and the partial outage. However, in between, there are two down events inferred by a lone address which are changed to unknown by LABR ((a), the top graph).

False Outages: Does FBS Remove Noise? From this single block example, we next consider a country's Internet. Our goal is to see if FBS reduces noise by

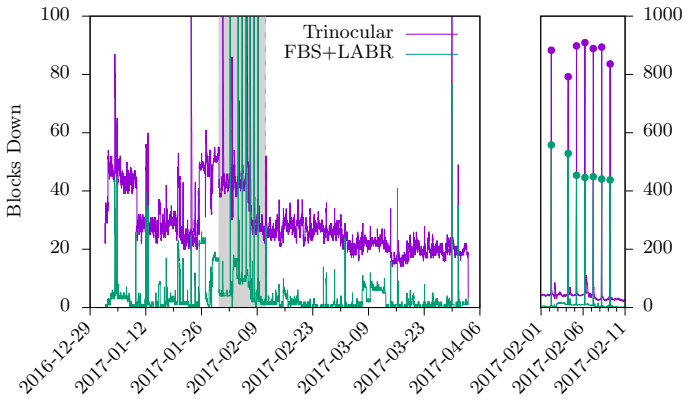

Fig. 2. Iraqi Government mandated outages Feb 2-9, 2017. Whole quarter (left), and exam week (right). Dataset A27. FBS processed using 2FR.

examining false down events (blocks correctly recovered by FBS because they were observation noise).

We study series of known outages that affected Iraq in February 2017. That country had seven government-mandated Internet outages (the local mornings on February 2, and also the 4th through 9th) with the goal of preventing cheating during academic placement exams [5]. This is a particularly challenging scenario to FBS, as closely spaced short outages test the algorithm's accuracy and precision. Furthermore, the fraction of sparse blocks is high in this country. We identified 1176 Iraqi blocks using Maxmind's city-level database [9]; 666 of these are sparse.

Fig. 2 shows Iraqi outages in 2017q1, grouped in 660 s timebins. We show outages without Full Block Scanning (the purple, top line) and with it (the green line). The Iraqi exam week is highlighted in gray on the left, and we plot that week with a larger scale on the right.

In each of the seven large peaks during exam week, most Iraqi blocks (nearly 900, or 76%) are out—our true outages. Outside the peaks, a few blocks (the 20 to 40 purple line, without FBS) are often down, likely false outages.

FBS suppresses most of the background outages (85% of outage area), from a median of 26 to a median of 1; these differences can be seen comparing the higher purple line to the lower green line. We confirm this reduction was due to noise by examining blocks that FBS recovers in 10 randomly-selected time periods with 34 down events. Nearly all down events (33 events, 97% of purple) were in sparse blocks that resemble Fig. 1; the other block was diurnal. This study confirms that FBS recovers false outages due to sparseness.

True Outages: Does FBS Remove Legitimate Outages? We next look at how Full Block Scanning interacts with known outage events. Its goal is to remove noise and false outages, but if FBS is too aggressive it may accidentally remove legitimate outages (a "true down event").

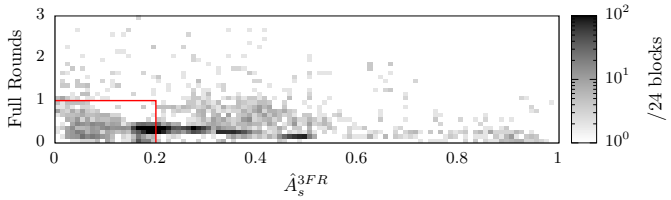

Fig. 3. Outage events during the 7 Iraqi outages, measured of their $\hat{A}_s^{3FR}$ and full round values. Single site W. Dataset: A27 (2017q1) subsetted to the 7 outage periods.

Table 2. Confusion matrix of 5200 Trinocular detected down events in 50 random blocks. Dataset A30, 2017q4.

		true condition (manually observed)	
		UP (Trinocular false down events)	DOWN (Trinocular true down events)
FBS	UP	4133 (79%, FBS fixes)	0
	DOWN	621 (12%, FBS misses)	446 (9%, FBS confirms)

We treat the seven nationwide outages corresponding with Iraqi exams as true down events and compare this ground truth, with and without FBS.

The seven peaks in Fig. 2 (right) show known Iraqi outages, with purple dots at "peak outage" without FBS, and lower, green dots with FBS. FBS removes somewhat less than half of the down events, with peaks around 440 to 560 instead of 790 to 910 blocks.

To understand this reduction we looked at the duration of the Iraqi events. FBS affects only the 35% of events in the red box in the lower left corner. (Examination of just sparse blocks confirms that they are the source of attenuation.)

It is important to note that these are *worst case* for FBS—many blocks are sparse, and the events are just shorter than one full round. If the event was longer or more blocks were not sparse, there would be no attenuation. A lower FBS threshold ($\hat{A}^{3FR}$) of 0.15 trims only 15% of events. However, we choose to leave FBS threshold at 0.2 to avoid overfitting our parameters to Iraq.

Random Sampling of Outage Events Finally, we confirm our results with a random sample of events. We select 50 random blocks that show some outage from the Trinocular 2017q4 dataset, then a best-estimate ground truth through manual examination. Table 2 shows the confusion matrix after applying FBS. Of the total 5200 down events detected by Trinocular, FBS fixes 4133 (79% are false outages), misses 621 down events (12% are not fixed, but should have been), and confirms 446 true down events (9% are not changed). The FBS Error Rate is 0.12 (621 false outages of 5200 events), so it is fairly successful at removing noise. Many of the false outages are due to moderately sparse blocks ($0.2 < A(E(b)) < 0.4$) where FBS does not trigger.

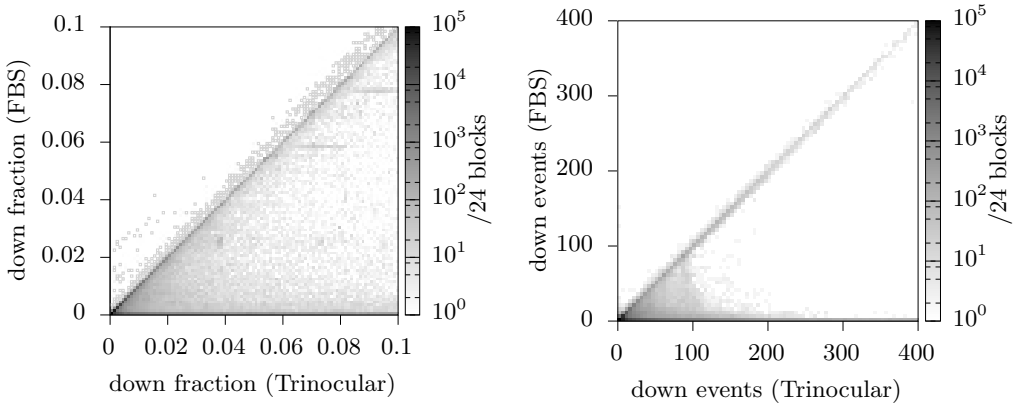

Fig. 4. Comparison of per-block down time (left) and number of down events (right) between 2FR-FBS and Trinocular during 2017q4 as seen from six sites. Dataset A30.

4.2 How Often Does FBS and LABR Change Outages?

We next evaluate how FBS and LABR change the overall down event duration and the number of down events. We expect FBS to repair false down events, so it should show less downtime and fewer down events.

We evaluate merged results from six Trinocular sites as measured during 2017q4 (dataset A30) and compute fraction of time and number of occurrences across the whole quarter each block was observed down. We repeat the procedure with data processed with FBS.

We compare outages for 2017q4 (dataset A30), processing and merging results from six sites with and without FBS. We found similar results when we repeated this study on a different quarter (2017q2, dataset A28).

FBS and Down Time: Fig. 4 (left) compares the fraction of total down time (0.0130) with FBS (0.0027). First, the vast majority of blocks (91%) have both values less than 0.02—they have little or no down time. (see Appendix C). Many of the remaining blocks are on the diagonal, with prior and new values within 0.005. We also see most of the changed blocks (9% of all blocks) appear below the diagonal, showing that FBS usually decreases downtime.

Surprisingly, 0.5% of blocks show *more* downtime after FBS. We examined a sample of these blocks and found that some sparse blocks did not transition from up-to-down in one round when 15 negative results did not fully change belief. FBS gathers more information and retrospectively marks the block down earlier. We believe this result better reflects truth.

FBS and Down Events: We can also evaluate how FBS affects the number of down *events* in addition to down *time* in Fig. 4(right). FBS reduces the number of down events by 6% of blocks, often considerably (see the large number of blocks near the x-axis). In these cases FBS is repairing false outages. Again, we

Table 3. Trinocular-detected disruptions in CDN logs. Dataset A28, 2017q2.

	Trinocular	filtered Trinocular	FBS
# disruptions	380k	132k	119k
confirmed	103k 27%	98k 74%	92k 77%
reduced activity	49k 13%	∼13k 10%	16k 14%
no change	228k 60%	∼21k 16%	11k 9%

see a small number of blocks (0.1%) where FBS shows more down events than without. Examination of these cases shows that FBS sometimes breaks longer down events into several shorter ones, interspersed with an up event. We believe these results better reflect the true state of the block.

LABR: In 2017q4, LABR affects only a few blocks (250k, 6% of trackable), where it resets 4M down events to unknown. LABR affects only a few blocks, but it allows them to be reported up much of the time, increasing coverage.

4.3 Comparing FBS Active and Passive Outages

Prior CDN-based results showed the large number of false outages that come from a few blocks [14]. To match their system, they compare the subset of 1.6M blocks from 2017q2 that are trackable in both Trinocular and their system and that are at least 1 hour or longer in Trinocular. We next review that result and show that FBS solves the problem they identified.

Table 3 shows this comparison of CDN events to Trinocular with both filtering (discarding blocks with more than 5 events, a short-term fix proposed for their paper at the time) and FBS. To recap prior results: The CDN-based results summarized in confirm that 27% of outage events found by Trinocular without FBS also appear in the CDN-based passive analysis. The remaining outages are either false outages in Trinocular (likely, since 60% show *no* change in the CDN) or false uptimes from the CDN. Given sparse blocks produce many events, discarding blocks with 5 or more events (the "filtered Trinocular" column) should avoid most false outages, although it may cause false uptime. As expected, most events (74%) that remain after this filter are confirmed by the CDN.

While CDN-data is proprietary and is not available, we thank Philipp Richter for redoing this comparison with a similar subset of our data updated, but now with FBS. The FBS column of Table 3 shows analysis of Trinocular with FBS compared to the same CDN results, now filtered only by the CDN requirements (1 hour events, and reported in the CDN system). FBS brings an even larger fraction of disruptions in-line with the CDN, with 77% of events being confirmed. Moreover, FBS is much more sensitive than the 5-event filter, applying only to the 22% of blocks that are sparse blocks. FBS therefore preserves Trinocular's 11-minute timing for the majority of blocks, reducing temporal precision only where necessary while providing generally good accuracy for outage detection across all blocks.

This result suggests that FBS addresses the majority of false outages, and confirms that most false outages are due to a small set of sparse blocks. (Addressing false outages due to ISP renumbering is ongoing work [2].)

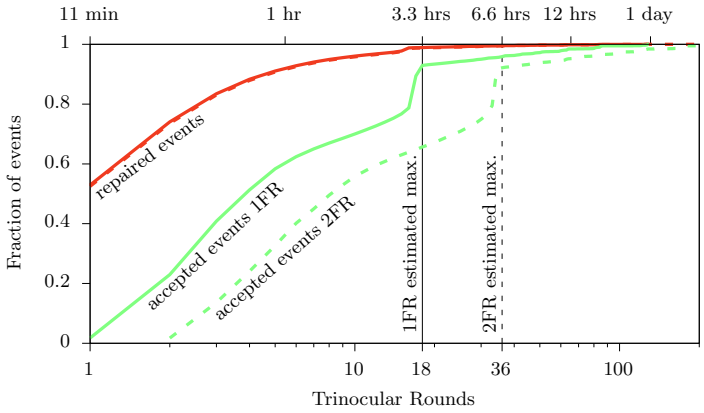

Fig. 5. FBS temporal precision analysis measured in repaired down events (false outages) and accepted events minimum (true outages) duration considering 1FR and 2FRs during 2017q4 as seen from Los Angeles. Dataset A30.

Finally, we note that FBS provides much larger coverage: 5.7M blocks compared to 2.3M trackable blocks in the CDN-system. We discuss coverage in detail in §4.5.

4.4 FBS Effects on Temporal Precision

We first examine how FBS affects temporal precision of outages. In sparse blocks, FBS will repair down events that are shorter than a Full Round. But the exact duration of a FR depends how many addresses are considered in the block ($E(b)$) and how active they are ($\hat{A}(E(b))$).

To analyze FBS changes to temporal precision, we consider *repaired events*, false down events corrected by FBS, and *accepted events*, true down events that pass through FBS unchanged. LABR does not affect temporal precision.

We study FBS effects by examining the 2017q4 outage dataset (A30) from one site (Los Angeles). Most blocks (2.8M blocks, 70%) are never affected by FBS because they are not sparse or do not have an outage. For the remaining 1.2M blocks that are at some point sparse ($\hat{A}^{3FR} < 0.2$) and for which Trinocular reports an outage, we examine each outage event.

We first examine the 308M events that FBS repairs (the top left, red line in Fig. 5, 1.2M blocks). We see that for about half the cases (53% of the events), FBS repairs a single-round of outage in 11 minutes. Almost all the remaining events are recovered in 15 or fewer rounds, as expected. Only a tiny fraction (0.5%) require longer than 18 rounds, for the few times when Trinocular is slow to detect large changes in $\hat{A}$ because it thinks the block may be down.

The light green solid line in the middle Fig. 5 shows how long full rounds last for outages that pass FBS. Of 5.1M events, we see that 60% are approved in less than one hour (five or fewer TRs). About 8% of events take longer than our expected maximum of 18 TRs. We examined these cases and determined these are cases where Trinocular has such confidence the block should be up it

Table 4. IPv4 address space coverage of Trinocular and FBS. (a), (b) and (c) different methods for filtering sparse blocks. (d) blocks fixed by FBS.

	Threshold	Blocks (in M) reject	accept	%resp	*%Tri*
IPv4 responsive	$\|E(b)\| \geq 1$	8.6	5.9	100	
Trinocular trackable	$\|E(b)\| \geq 15 \wedge A \geq 0.1$	1.9	4.0	67	*100*
a) mostly up blocks	up time > 0.8	0.2	3.8	64	*95*
b) infrequently down blocks	# down events < 5	0.3	3.7	63	*93*
c) non-sparse blocks	$A \geq 0.2$	0.9	3.1	53	*78*
d) FBS considered	$\hat{A}^{3FR} < 0.2$	2.8	1.2	-	*30*
overlap with (c)		0.6	0.8	-	-
FBS trackable	$\|E(b)\| \geq 3$	0.2	5.7	96	*142*

does not probe all 15 tries. We confirm this result examining 50 random blocks within the tail.

Use of FBS on old Trinocular data requires the 2FR variant of FBS, with more TRs per FR (see §3.1). Dashed lines in Fig. 5 show 2FR analysis. We see almost no change in temporal precision of repaired events (nearly all the range of the solid and dashed red lines overlap). Accepted outages take roughly twice as long as with 1FR FBS, and the number drops to roughly one half (3.1M accepted down events); fortunately only 0.13% of all 4M blocks require 2FR and have actual outages in 2017q4.

We currently use FBS in batch processing, and we plan to implement it in our near-realtime (NRT) outage pipeline soon. For NRT processing one can either delay results while FBS is considered, or report preliminary results and then change them if FBS corrects.

4.5 Increasing Coverage

Sparse blocks limit coverage. If historical information suggests they are sparse, they may be discarded from probing as untrackable. Blocks that become sparse during measurement can create false outages and are discarded during post-processing. We next show how FBS and LABR allow us to increase coverage by correctly handling sparse blocks.

Correctly tracking sparse blocks: We first look at how the accuracy improvements with our algorithms increase coverage. Three thresholds have been used to identify (and discard) sparse blocks: a low response probability ($A < 0.2$, quarter average, from [12]), low up time (up time < 0.8, from [13]), and high number of down events (5 or more down events, from [14]).

We use these three thresholds over one quarter of Trinocular data (2017q4-A30W), reporting on coverage with each filter in Table 4. With 5.9M responsible blocks, but only 4M of those (67%) are considered trackable by Trinocular. Filtering removes another 0.2M to 0.9M blocks, leaving an average of 53 to 64%.

Trinocular with FBS gets larger coverage than other methods of filtering or detection. FBS repairs 1.2M blocks, most sparse: of 0.9M sparse blocks, we find that FBS fixes 0.8M. The remaining 100k correspond to either good blocks that went dark due to usage change and therefore pushing the quarterly average of A down, or sparse blocks with few active addresses (for example, $|E(b)| < 100$) where Trinocular can make a better job inferring the correct state.

Can FBS+LABR expand baseline coverage? Finally, we examine the number of blocks discarded as untrackable from historical data, and are not tracked for outages. For instance, Trinocular looks at the last 16 surveys [7], and filter all blocks with $|E(b)| < 15$ and $A < 0.1$, left with its baseline of 4M blocks.

In a similar approach, we use the 2017-04-27 survey as our upper bound of the responsive Internet [8]. As Table 4 shows, we find 5.9M responsive blocks, of which 5.7M had at least three active addresses during the measured period. That is 1.7M (43%) more blocks than the baseline become trackable. When adding 1.7M with the number of FBS-repaired blocks (1.2M), our effective coverage increment adds to 2.9M blocks.

5 Related Work

Several groups have methods to detect outages at the Internet's edge: ThunderPing first used active measurements to track weather-related outages on the Internet [15,11]. Dainotti et al. use passive observations at network telescope to detect disasters and government censorship [4], providing the first view into firewalled networks. Chocolatine provides the first published algorithm using passive network telescope data [6], with a 5 minute detection delay, but it requires AS or country level granularity, much more data than /24s. Trinocular uses active probes to study about 4M, /24-block level outages [12] every 11 minutes, the largest active coverage. Disco observes connectivity from devices at home [16], providing strong ground truth, but limited coverage. Richter et al. detect outages that last at least one hour with CDN-traffic, confirming with software at the edge [14]. They define disruptions, showing renumbering and frequent disagreements in a few blocks are false down events in prior work. Finally, recent work has looked at dynamic addressing, one source of sparsity [10]. Our work builds on prior active probing systems and the Trinocular data and algorithms, and addresses problems identified by Richter, ultimately due to sparsity and dynamics.

6 Conclusions

This paper defines two algorithms: Full Block Scanning (FBS), to address false outages seen in active measurements of sparse blocks, and Lone Address Block Recovery (LABR), to handle blocks with one or two responsive addresses. We show that these algorithms increase coverage, from a nominal 67% (and as low as 53% after filtering) of responsive blocks before to 5.7M blocks, 96% of responsive blocks. We showed these algorithms work well using multiple datasets and natural experiments; they can improve existing and future outage datasets.

Acknowledgments

We thank Yuri Pradkin for his input on the algorithms and paper.

We thank Philipp Richter and Arthur Berger for discussions about their work, and Philipp for re-running his comparison with CDN data.

The work is supported in part by the National Science Foundation, CISE Directorate, award CNS-1806785; by the Department of Homeland Security (DHS) Science and Technology Directorate, Cyber Security Division (DHS S&T/CSD)

via contract number 70RSAT18CB0000014; and by by Air Force Research Laboratory under agreement number FA8750-18-2-0280. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon.

References

1. IODA: Internet outage detection & analysis, https://ioda.caida.org
2. Baltra, G., Heidemann, J.: Improving the optics of active outage detection (extended). Tech. Rep. ISI-TR-733 (May 2019), https://www.isi.edu/%7ejohnh/PAPERS/Baltra19a.html
3. Dainotti, A., Benson, K., King, A., Huffaker, B., Glatz, E., Dimitropoulos, X., Richter, P., Finamore, A., Snoeren, A.: Lost in Space: Improving Inference of IPv4 Address Space Utilization. IEEE Journal on Selected Areas in Communications (JSAC) **34**(6), 1862–1876 (Jun 2016)
4. Dainotti, A., Squarcella, C., Aben, E., Chiesa, M., Claffy, K.C., Russo, M., Pescapé, A.: Analysis of country-wide Internet outages caused by censorship. In: Proceedings of the ACM Internet Measurement Conference. pp. 1–18. ACM, Berlin, Germany (Nov 2011). https://doi.org/http://dx.doi.org/10.1145/2068816.2068818
5. Doug Madory: Iraq downs internet to combat cheating...again! https://dyn.com/blog/iraq-downs-internet-to-combat-cheating-again/ (2017), accessed: 2019-01-08
6. Guillot, A., Fontugne, R., Winter, P., Merindol, P., King, A., Dainotti, A., Pelsser, C.: Chocolatine: Outage detection for internet background radiation. In: Proceedings of the IFIP International Workshop on Traffic Monitoring and Analysis. IFIP, Paris, France (Jun 2019), https://clarinet.u-strasbg.fr/~pelsser/publications/Guillot-chocolatine-TMA2019.pdf
7. Heidemann, J., Pradkin, Y., Govindan, R., Papadopoulos, C., Bartlett, G., Bannister, J.: Census and survey of the visible Internet. In: Proceedings of the ACM Internet Measurement Conference. pp. 169–182. ACM, Vouliagmeni, Greece (Oct 2008). https://doi.org/http://dx.doi.org/10.1145/1452520.1452542
8. Internet Addresses Survey dataset, PREDICT ID: USC-LANDER/internet-address- survey-reprobing-it75w-20170427/:
9. MaxMind: GeoIP Geolocation Products. http://www.maxmind.com/en/city (2017)
10. Padmanabhan, R., Dhamdhere, A., Aben, E., kc claffy, Spring, N.: Reasons dynamic addresses change. In: Proceedings of the ACM Internet Measurement Conference. ACM, Santa Monica, CA, USA (Nov 2016). https://doi.org/https://doi.org/10.1145/2987443.2987461
11. Padmanabhan, R., Schulman, A., Levin, D., Spring, N.: Residential links under the weather. In: Proceedings of the ACM Special Interest Group on Data Communication. pp. 145–158. ACM (2019)
12. Quan, L., Heidemann, J., Pradkin, Y.: Trinocular: Understanding Internet reliability through adaptive probing. In: Proceedings of the ACM SIGCOMM Conference. pp. 255–266. ACM, Hong Kong, China (Aug 2013). https://doi.org/http://doi.acm.org/10.1145/2486001.2486017
13. Quan, L., Heidemann, J., Pradkin, Y.: When the Internet sleeps: Correlating diurnal networks with external factors. In: Proceedings of the ACM Internet Measurement Conference. pp. 87–100. ACM, Vancouver, BC, Canada (Nov 2014). https://doi.org/http://dx.doi.org/10.1145/2663716.2663721

14. Richter, P., Padmanabhan, R., Spring, N., Berger, A., Clark, D.: Advancing the art of Internet edge outage detection. In: Proceedings of the ACM Internet Measurement Conference. ACM, Boston, Massachusetts, USA (oct 2018). https://doi.org/https://doi.org/10.1145/3278532.3278563
15. Schulman, A., Spring, N.: Pingin' in the rain. In: Proceedings of the ACM Internet Measurement Conference. pp. 19–25. ACM, Berlin, Germany (Nov 2011). https://doi.org/https://doi.org/10.1145/2068816.2068819
16. Shah, A., Fontugne, R., Aben, E., Pelsser, C., Bush, R.: Disco: Fast, good, and cheap outage detection. In: Proceedings of the IEEE International Conference on Traffic Monitoring and Analysis. pp. 1–9. Springer, Dublin, Ireland (Jun 2017). https://doi.org/https://doi.org/10.23919/TMA.2017.8002902
17. USC/ISI ANT project: `https://ant.isi.edu/datasets/outage/index.html`

A Other Block Examples

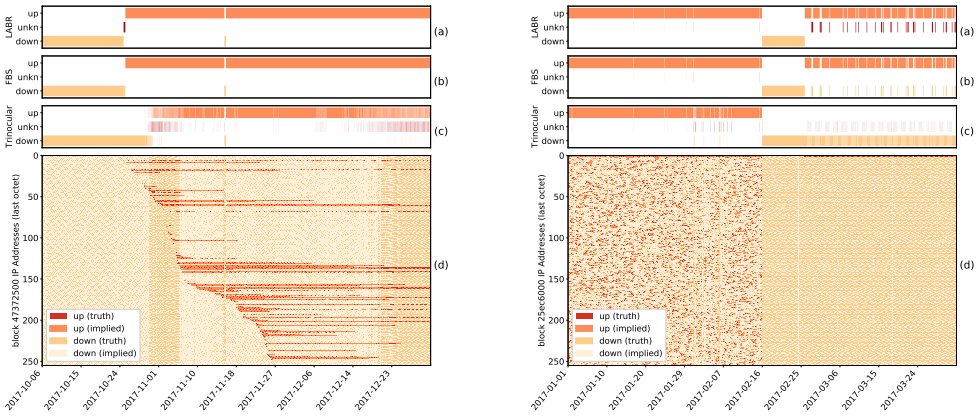

Fig. 6. Sample blocks over time (columns). The bottom (d) shows individual address as rows, with colored dots when the address responds to Trinocular. Bar (c) shows Trinocular status (up, unknown, and down), bar (b) is Full Block Scanning, and the top bar (a), Lone Address Block Recovery.

§2.1 described the problem of sparse blocks and why FBS is needed. Here we provide examples of other blocks where sparsity changes to illustrate when FBS is required.

The block in the left part of Fig. 6 has no activity for three weeks, then sparse use for a week, then moderate use, and back to sparse use for the last two weeks. Reverse DNS suggests this block uses DHCP, and gradual changes in use suggest the ISP is migrating users. The block was provably reachable after the first three weeks. Before then it may have been reachable but unused, a false outage because the block is inactive.

The third bar from the top (c) of the left of Fig. 6 we show that Trinocular often marks the block unknown (in red) for the week starting 2017-10-30, and again for weeks after 2017-12-12. Every address in this block has responded in

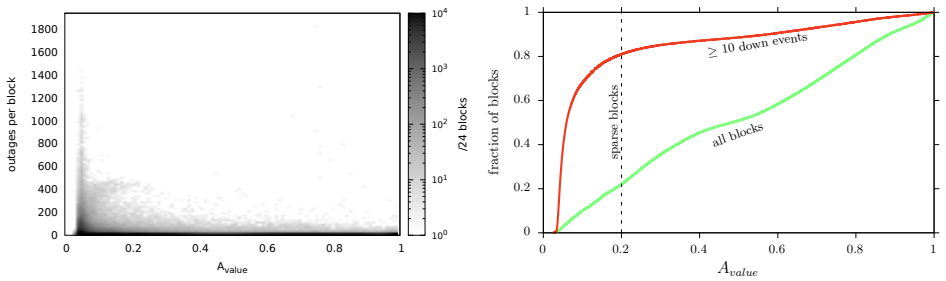

Fig. 7. Blocks distributed according to the number of outages versus their $A(E(b))$ (left), and cumulative distribution function of the A value per block (right) as collected during 2017q4 for the whole responsive IPv4 address scope. Dataset A30.

the past. But for these two periods, only a few *are actually* used, making the block *temporarily sparse*. Fig. 6 (left, bar b) shows how FBS is able to accurately fix Trinocular's pitfalls in such a DHCP scenario.

Fig. 6 (right) shows a block example with a lone address. This block has three phases of use: before 2017-02-16, many addresses are in use; then for about 9 days, *nothing* replies; then, starting on 2017-02-25 only the .1 address replies. During the last phase, Trinocular (Fig. 6 (right, bar c)) completely ignores that there is one address responding, while FBS (Fig. 6 (right, bar b)) sets block status depending on responses of this lone-address. However, LABR (Fig. 6 (right, bar a)) changes all the FBS detected down events to unknown, as there is not information to claim a down event, in contrast to what the end of phase one shows.

B Block Usage Change

As mentioned in §2.1, when blocks become temporarily sparse (showing a small $A(E(b))$), the number of false outages increases. On the other hand, denser blocks offer higher inference correctness.

Our prior work dynamically estimated A [13], but Richter et al. showed that block usage changes dramatically, so blocks can *become* overly sparse even with tracking [14].

We first show that sparse blocks cause the majority of outage events. In Fig. 7 (left) we compare the number of outages in all 4M responsive blocks with their measured $A(E(b))$ value during 2017q4. Blocks with a higher number of outages tend to have a lower $A(E(b))$ value. In particular those closer to the lower bound. Trinocular does not track blocks with long term $A(E(B)) < 0.1$, however as blocks sparseness changes, this value does change during measure time.

The correlation between sparse blocks and frequent outage events is clearer when we look at a cumulative distribution. Fig. 7 (right) shows the cumulative distribution of A for all 4M responsive blocks (light green, the lower line), and for blocks with 10 or more down events (the red, upper line) as measured during 2017q4. These lines are after merging observations obtained from six Trinocular vantage points. We find that 80% of blocks with 10 or more down events have an

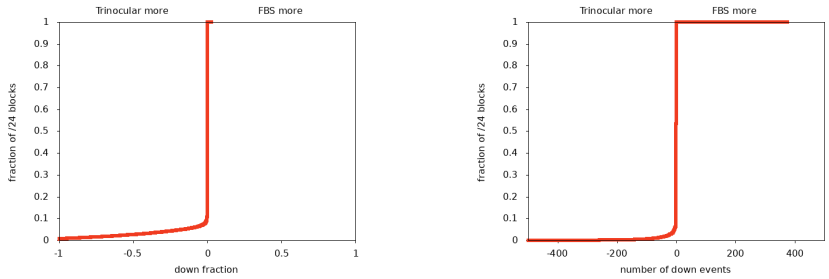

Fig. 8. Cumulative distribution of down fraction difference (left) and number of down events difference (right) between Trinocular and FBS for 2017q4. Dataset A30.

$A < 0.2$, at around the knee of the curve, and yet these sparse blocks represent only 22% of all blocks. The figure suggests a correlation between high number of down events and low $A(E(b))$ per block due to the faster convergence of the line representing blocks with multiple down events. (It confirms the heuristic of "more than 5 events" that was used to filter sparse Trinocular blocks in the 2017 CDN comparison [14].)

Although we observe from multiple locations, merging results from different vantage points is not sufficient to deal with sparse blocks, because these multiple sites all face the same problem of sparseness leading to inconsistent results. Addressing this problem is a goal of FBS, and it also allows us to grow coverage.

C Comparing Trinocular and FBS

In §4.2 we discuss how often FBS changes outages when compared to Trinocular. We examine two different metrics: total block down time and number of down events. Here we provide further information distribution about the distribution of these metrics.

In Fig. 8 (left) we show block distribution of Trinocular and FBS down time fraction difference. The majority of blocks (91%) have little or no change. Blocks on the left side of the figure representing 9% of the total, have a higher down time fraction when processed only with Trinocular than when processed with FBS. For example, a -1 shows a block that was down for Trinocular during the whole quarter, while FBS was able to completely recover it. This outcome occurs when a historically high $|E(b)|$ block has temporarily dropped to just a few stable addresses.

We also see a small percentage (0.5%) where FBS has a higher down fraction than Trinocular. This increase in outages fraction happens when Trinocular erroneously marks a block as UP. With more information, FBS is able to correctly change block state and more accurately reflect truth.

In Fig. 8 (right) we look to the distribution of blocks when compared by the number of down events observed in FBS and Trinocular. Similarly, the number of down events remains mostly unchanged for the majority of blocks (94%). Trinocular has more down events for 6% of blocks, and FBS shows more events for 0.1%. FBS can increase the absolute number of events in a block when breaking long events into shorter pieces.