Crowd-based Detection of Routing Anomalies on the Internet

Rahul Hiran
Linköping University, Sweden
rahul.hiran@liu.se

Niklas Carlsson
Linköping University, Sweden
niklas.carlsson@liu.se

Nahid Shahmehri
Linköping University, Sweden
nahid.shahmehri@liu.se

***Abstract*—The Internet is highly susceptible to routing attacks and there is no universally deployed solution that ensures that traffic is not hijacked by third parties. Individuals or organizations wanting to protect themselves from sustained attacks must therefore typically rely on measurements and traffic monitoring to detect attacks. Motivated by the high overhead costs of continuous active measurements, we argue that passive monitoring combined with collaborative information sharing and statistics can be used to provide alerts about traffic anomalies that may require further investigation. In this paper we present and evaluate a user-centric crowd-based approach in which users passively monitor their network traffic, share information about potential anomalies, and apply combined collaborative statistics to identify potential routing anomalies. The approach uses only passively collected round-trip time (RTT) measurements, is shown to have low overhead, regardless if a central or distributed architecture is used, and provides an attractive tradeoff between attack detection rates (when there is an attack) and false alert rates (needing further investigation) under normal conditions. Our data-driven analysis using longitudinal and distributed RTT measurements also provides insights into detector selection and the relative weight that should be given to candidate detectors at different distances from the potential victim node.**

***Keywords*—Crowd-based detection, Collaboration, Routing anomalies, Interception attacks, Imposture attacks**

I. Introduction

Despite being one of the most important infrastructures in today's society, the Internet is highly susceptible to routing attacks that allow an attacker to attract traffic that was not intended to reach the attacker [1], [2]. Recently there have been increasing occurrences of routing attacks. Some of these attacks have been performed to gain access to sensitive or valuable information, or to circumvent constitutional and statutory safeguards against surveillance of a country's citizen, while others have been accidental or tried to deny users access to certain services (for censorship purposes) [2]–[5]. With many of these attacks affecting regular end users, routing path integrity is becoming increasingly important not only for operators, but also for end users.

While *black-holing attacks* (e.g., as used in some censorship attacks) are relatively easy to detect, as the traffic is terminated at the attacker and the traffic source may not obtain expected end-to-end responses, attacks where the attacker also impersonates the destination (*imposture attacks*) or re-routes the traffic to the destination (*interception attacks*) are much more difficult to detect.

Although this is a globally important problem and several crypto-based efforts to secure internet routing have been proposed [6]–[8], deployment have been hampered by high deployment costs and failure to incentivize network operators to deploy these solutions [1], [2]. Instead, operators and other organizations running their own Autonomous Systems (AS) are typically limited to monitoring path announcements made by other ASes and/or the data paths taken by the data packets themselves [9], [10]. However, these detection-based approaches are typically limited by the number of BGP vantage points contributing BGP updates, the limited view these provide, or the high overhead associated with continuous monitoring using active measurement techniques such as traceroutes.

As an effort to help detect routing anomalies and to push operators to implement secure mechanisms, we foresee a user-driven approach in which concerned citizens collaboratively detect and report potential traffic hijacks to operators and other organizations that can help enforce route security. To reduce the number of active measurements needed we argue that clients could use passive monitoring of their day-to-day network traffic to identify prefixes (range of IP addresses) that may be under attack.

In this paper we present and analyze a user-centric crowd-based approach in which clients (i) passively monitor the round-trip times (RTT) associated with their users' day-to-day Internet usage, (ii) share information about potential anomalies in RTT measurements, and (iii) collaboratively help identify routing anomalies that may require further investigation. The use of passive measurements ensures zero measurement bandwidth overhead, helps distribute the monitoring responsibility of prefixes based on the users' interaction with services provided within each prefix, and ensures timely initial detection of potential routing anomalies as seen by the users themselves. The use of crowd-based sharing of potential anomalies allows the number of alerts that need further investigation to be reduced during normal circumstances and the attack detection rates to be increased during attacks.

This paper makes three main contributions. First, we use longitudinal RTT measurements from a wide range of locations to evaluate the anomaly detection tradeoffs associated with crowd-based detection approaches that use RTT information (Section III). Considering two different types of stealthy

and hard-to-detect attacks (interception attacks and imposture attacks), we provide insights into the tradeoff between attack detection rates (when there is an attack) and false alert rates needing further investigation under normal circumstances (when there is no attack), and how this tradeoff depends on different system parameters such as system scale, participation, and the relative distances between detectors, attackers, and victims. Second, using a simple system model and trace-driven simulations, we evaluate the set of detector nodes that provides the best detection rates for a candidate victim (Section IV). Of particular interest is the relative weight that should be given to candidate detectors at different distance from the victim node. Finally, we provide a discussion and analysis of the overhead associated with different candidate architectures, including both central directory approaches and fully distributed solutions (Section V). Regardless of the implementation approach, the system is shown to operate with relatively low overhead and simple heuristics can be used to identify sets of nodes that are well-suited for collaborative detection of individual prefixes.

The paper is organized as follows. Section II provides background on routing attacks and the techniques typically used to detect and secure against these attacks. Sections III-V present each of our three main contributions (outlined above). Finally, Section VI presents related work, before Section VII concludes the paper.

II. Background

A. Routing Attacks

Internet packets are highly vulnerable to routing attacks. This is in part due to the complex nature of Internet routing and in part due to the lack of globally deployed security mechanisms. A typical Internet packet traverses many routers operated by different operators and Autonomous Systems (AS), each with a separate administrative domain and its own policies, and the packet's wide-area (interdomain) route is determined by the Border Gateway Protocol (BGP).

While many routing incidents go undetected, there have recently been some bigger incidents that have drawn global attention, including a small Indonesian ISP taking Google offline in parts of Asia, Pakistan Telecom taking YouTube offline for most of the Internet, China Telecom attracting and re-routing a large fraction of the world's Internet traffic, and highly targeted traffic interceptions by networks in Iceland and Belarus [2], [3], [11]. Although not all of these incidents were intentional, or can be proven intentional, it is important to be able to effectively detect them when they happen.

One of the biggest vulnerabilities with BGP is its inability to confirm the allocation of prefixes to corresponding ASes. This allows malicious entities to perform a *prefix hijack attack* in which the attacker announces one or more prefixes allocated to other networks. The effectiveness of this type of attack is typically determined by the route announcements made by the different ASes announcing the same prefix and the routing policies of the ASes that must select which of the alternative paths to use. Since forwarding routers always select more specific subprefix, a more effective attack is a *subprefix hijack attack* in which the attacker announces one or more subprefixes of the prefix allocated to the victim network.

These attacks can be further classified by the actions taken by the attacker. In *black-holing attacks* the traffic is terminated at the attacker and the originator of the packet will not see a proper response. Although some investigation may be needed to determine the cause, in general, these attacks are relatively easy to detect. In contrast, it is more difficult to detect attacks in which the attacker relays the attracted traffic to the intended destination (allowing proper end-to-end communication), or where the attacker impersonates the intended destination. These two types of attacks are typically referred to as *interception attacks* and *imposture attacks*, respectively, and will be the focus of this paper.

B. BGP Security and Monitoring

Many techniques have been proposed to protect against routing incidents and attacks such as those described above. These techniques typically either use prefix filtering or cryptography-based techniques. With prefix filtering, a responsible organization (AS) can use whitelists to protect the rest of the Internet from potential attacks performed by its customer networks (i.e., ASes that pay the provider AS to send and receive data through its network). Resource Public Key Infrastructure (RPKI) [6] and other cryptographic origin validation techniques typically build a trusted and formally verifiable database of prefix-to-AS pairings between the IP prefixes and the ASes that are allowed to originate them. Using protocols such as BGPsec [12], for example, RPKI can be used to protect the AS path attribute of BGP update messages. Recently, DNSSEC-based [13] approaches such as ROVER [14] have been proposed to cryptographically securing and authorizing BGP route origins.

Unfortunately, deployment of these solutions has been slow and the solutions do not work well unless a large number of networks deploy them [1], [2]. The main reasons for slow deployment have been limited incentive for independent organizations, and because there is no single centralized authority that can mandate the deployment of a (common) security solution. Deployment may also have been hampered by political and business implications from hierarchal RPKI management giving some entities (e.g., RIRs) significant control over global Internet routing [15].

Without large-scale deployment of crypto-based solutions [1], organizations often instead rely on centralized and decentralized monitoring solutions for anomaly detection in BGP. For example, BGPMon[1] and Team Cymru[2] centrally collect routing information from distributed monitors, and create alerts/summary reports about routing anomalies to which organizations can subscribe. PrefiSec [16] and NetReview [17] provide distributed and/or collaborative alternatives.

Data-plane based [9], [18], control-plane based [10], [19], and hybrid techniques [16], [20], [21] have been proposed

[1]BGPMon, http://www.bgpmon.net/, May 2014.

[2]Team Cymru, http://www.team-cymru.org, May 2014.

to detect routing anomalies. Typically data-plane based techniques use active traceroute measurements by different organizations and control-plane techniques rely on the AS-PATH in different BGP route announcements. Rather than relying on active measurements or sharing of BGP announcements between organizations, this paper considers an alternative approach in which we leverage passive measurements at the clients to detect suspicious RTT deviations caused by underlying attacks. This allows any concerned citizen to help detect anomalous routes, which then can be analyzed more closely using active measurements and complimentary control-plane information.

III. System Model and Detection Tradeoffs

We consider a general system framework, which we call CrowdSec, in which stationary end-user clients passively monitor the RTTs to the prefixes (range of IP addresses) with which the client applications (e.g. a web browser) are interacting. Leveraging such measurements has many advantages. First, passive measurements does not add any measurement traffic to the network and hence does not affect the bandwidth share given to the user applications. Second, the high skew in service accesses [22] ensures repeated measurements (over time) to many prefixes.

Each CrowdSec client passively monitors the RTTs associated with their users' day-to-day Internet usage. Using Grubb's test [23], each client individually identifies anomalies in the RTTs and shares information about these anomalies through the CrowdSec system, including statistics about how unlikely such an RTT deviation (or more extreme) are, given past observations. Using binomial testing (or alternative tests) these shared statistics are then combined into refined statistical measures that capture how unlikely the combination of observations would be under normal conditions, allowing for collaborative detection of routing anomalies.

As RTT-based measures include less information than full traceroutes and other active measurement techniques might provide, CrowdSec and other purely RTT-based approaches should not be used as primary evidence for routing attacks. Instead, further investigation is typically needed to distinguish temporary increases in RTTs due to bufferbloat and other temporal congestion events, for example, from routing anomalies. Yet, as active traceroute-driven approaches that capture the data path come at much higher overhead and can be forged by the attacker, the use of passively collected longitudinal RTT measurements can be used to identify opportune times to perform such active measurements and to sanity check the claimed data paths. In the following we describe and evaluate the CrowdSec approach and how well such systems are able to detect attacks (when attacks occur), while maintaining low alert rates under normal conditions (when there is no attack).

A. System model and evaluation framework

We consider a simple model with a set of detectors ($\mathcal{D}$), attackers ($\mathcal{A}$), and victims ($\mathcal{V}$). In the case of a successful *interception attack*, we assume that an attacker $a \in \mathcal{A}$ successfully re-routes the traffic on its way from a detector $d \in \mathcal{D}$ to a victim $v \in \mathcal{V}$ (that would normally take the route $d \to v \to d$) such that it instead takes the route $d \to a \to v \to d$. The hijacked path in this example is illustrated in Figure 1(a). Similarly, in the case of a successful *imposture attack* (Figure 1(b)), we assume that the attacker a intercepts the traffic between the detector d and victim v, and responds directly to the detector d, such that the end-to-end traffic (including the replies) takes the route $d \to a \to d$ instead of the intended route $d \to v \to d$.

TABLE I. Summary of datasets analyzed in paper.

			Simulated attack scenarios	
Year	Nodes (ave)	Traceroutes	Interception	Imposture
2014	113	278,690	15,279	62,576
2015	169	368,887	18,233	–

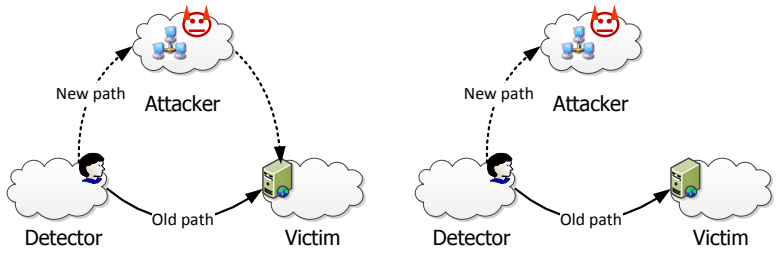
Under the above assumptions, the RTT effects of any successful interception and imposture attack can be estimated based on observed individual RTTs. For example, in the case of a successful interception attack by a on d and v, we compare the new RTT (estimated as $\frac{1}{2}(RTT_{d,a} + RTT_{a,v} + RTT_{d,v})$ during the attack) with the previously measured $RTT_{d,v}$ of the original path (over some prior time period). Here, we use $RTT_{x,y}$ to denote the RTT between a source-destination pair (x, y) in our measurements. Similarly, for the imposture attack, we compare the new RTT (estimated as $RTT_{d,a}$ during the attack) with the previously measured $RTT_{d,v}$ of the original path (over prior time period). Finally, we also evaluate the system during an example day without attack, when comparing with the previously measured $RTT_{d,v}$ of the original path (over prior time period).

For our evaluation, we leverage traceroute measurements recorded by PlanetLab nodes as part of the iPlane [24] project. In particular, we use daily RTT measurements from more than 100 globally distributed PlanetLab nodes to other planet lab nodes. We use a month's worth of training data (e.g., 278,690 successful traceroutes during July 2014) and evaluate the performance of different detection techniques for the following week, during which we simulate different attack combinations. Table I summarizes the datasets and simulated attacks analyzed in this paper. The lower number of simulated interception attack scenarios (compared to imposture scenarios) are due to both $RTT_{d,a}$ and $RTT_{a,v}$ needing to be present for the day of the attack, in addition to sufficient history of $RTT_{d,v}$ values during the earlier (training) period.

B. Single node detection

We envision that clients monitor their RTTs with different candidate victim IP addresses (or prefixes) and raise alarms when the measured RTTs deviate significantly from previously observed RTTs. For this analysis we apply Grubbs' test for anomaly detection [23]. We use an initial warmup period of at least N measurements, during which we assume that there are no anomalies, and then apply the Grubbs' test one measurement at a time. To minimize the effect of RTT variations, each "daily" RTT measurement considered here in fact is generated by taking the minimum RTT over three tracerouts performed each day.

We have applied both one-sided and two-sided hypothesis tests in each step. One-sided tests are natural for interception attacks, as the re-routing typically will not reduce the RTT.



(a) Interception attack (b) Imposture attack

Fig. 1. Evaluation scenarios for example interception and imposture attacks.

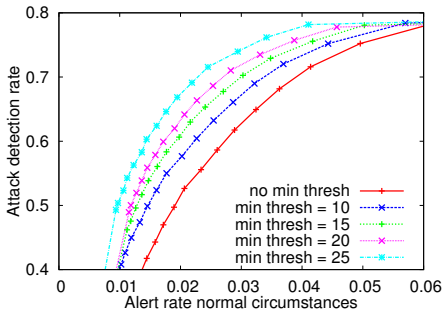


Fig. 2. Tradeoff between the alert rates of individual clients during attack and normal circumstances.

Although imposture attacks can result in both increases and decreases of the RTTs, it is important to note that the impersonator can easily hide such RTT reduction by adding additional delays. Unless otherwise stated, in the following we will show results for one-sided tests.

For our evaluation we use a 31 day warmup period (July 2014 or Jan. 2015) and evaluate the alert rates when there is an attack and when there is no attack, respectively, during the following week (first week of Aug. 2014 and Feb. 2015). Figure 2 shows the tradeoff between alerts raised by a single client during an attack and the (false) alerts raised by the client during normal circumstances for different thresholds N of the minimum measurements needed before raising alerts. For each threshold N, we varied the decision variable p^*_{ind} and counted the fraction of simulated attack (and non-attack) cases in which the observed RTT (with probability p) was considered an outlier (i.e., cases for which $p \leq p^*_{ind}$) and plotted the measured fractions during attack and non-attack conditions on the y-axis and x-axis respectively, while keeping p^*_{ind} as a hidden variable.

We note that the detection accuracy significantly depends on the number of measurements prior to detection, and that there are substantial improvements with increasing N. This is seen by the tradeoff curves for larger N being shifted more and more towards the top-left corner. Furthermore, we see that with a single node, the detection rate is consistently at least an order of magnitude higher than the alert rate during normal circumstances and can be more than 50 times higher for some thresholds. For example, with just $N = 25$ measurements a single node can achieve an alert rate of approximately 50% during attacks at the cost of less than 1% alerts (that may need further investigation) under normal conditions.

Given the high number of packets and high skew in the services that modern clients communicate with [22], most clients will quickly build up a significant sample history for many prefixes. For these services, N may therefore be much greater than 25, allowing for an even more advantageous tradeoff than illustrated here. Next, we look more closely at how these alert tradeoffs can be improved through collaborative information sharing, regardless of the individual clients' alert rate tradeoff.

C. Collaborative detection

As routing anomalies typically affect many users, CrowdSec can leverage the observations from different clients to refine the alert rate tradeoffs, and identify the extent of the anomaly (e.g., by measuring how many clients are affected). With CrowdSec, the users that detect an anomaly share the information with each other, allowing alerts about a particular prefix to be based on the combined anomaly information from multiple observers. In the following, we first describe a general collaboration analysis framework, before briefly discussing different statistical techniques.

We assume that clients always report their significant p-values (smaller than p^*_{ind}) and only report non-significant p-values with a small tunable probability. Here, a p-value represents an estimate of the probability that a client sees the observed RTT value given a history of RTT values, as estimated using Grubbs' test, for example. Given these two sets of p-values and knowledge of the tunable reporting probability, we can estimate both the number of non-reporting clients and the p-value distribution of the clients, allowing us to place the significant p-values in context, while keeping the communication overhead at a (tunable) minimum.

We use hypothesis testing to determine which prefixes and events to flag as potential anomalies. As for the single node case, our null hypothesis is that there is no significant RTT deviations from normal conditions. However, in these collaborative tests we can use all reported p-values. For our evaluation, we have tried three alternatives: the Binomial test, Fisher's test, and Stouffer's test. With the binomial test, a combined probability is calculated based on the binomial distributed probability of observing k or more "significant" p-values out of a total of n, given a probability threshold p^*_{bin}. With Fisher's and Stouffer's test, respectively, corresponding Chi-square and Z-score based metrics are calculated.

While Fisher's and Stouffer's tests theoretically should be able to make better use of the information in the individual p-values, both these tests runs into numerical problems for the type of values reported in our context (typically resulting in combined p-scores of either 0 or 1). Of these three methods, only the Binomial test therefore seems applicable to our context. In the following we therefore apply the Binomial test, with a normal distribution approximation when applicable.

D. Evaluation results

We next take a closer look at the attack detection rate during example attacks and (false) alert rate during normal circumstances when applying the Binomial test. Figure 3

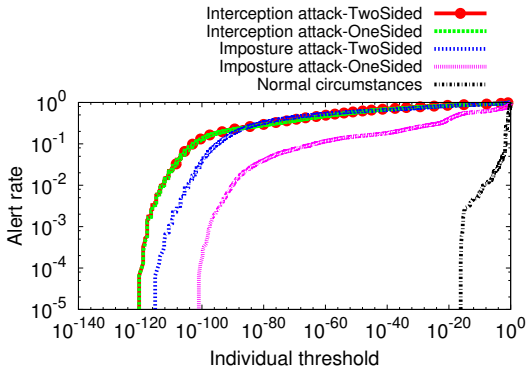


Fig. 3. Alert rate during normal circumstances, during interception attacks, and during imposture attacks.

shows the alert rates as a function of the probability threshold p^*_{bin} used in the binomial test for the cases when (i) there is no attack (normal circumstances), (ii) interception attacks occur, and (iii) imposture attacks occur. For the attacks, we include results for both single-sided and two-sided tests. As expected, for the interception attacks there is almost no difference between the one-sided and two-sided tests and the imposture attacks generally have lower detection rates (especially in the case where only the one-sided test can be applied).

While the alert rates are closely coupled with the selected probability threshold, it is important to note that there are very large differences in the alert rates during an attack (regardless of the type of attack) and during normal circumstances. This is very encouraging, as it shows that a high alert detection rate (during attacks) can be achieved, while maintaining a small (false) alert rate during normal circumstances. For example, in this case, a threshold $p^*_{bin} = 10^{-20}$ can detect the majority of the interception attacks, without resulting in a single false alert during normal circumstances.

As expected, the absolute alert tradeoffs are highly dependent on the Binomial threshold p^*_{bin} values. For example, both the attack detection (Figures 4(a)) and alerts during normal circumstances (Figures 4(b)) decrease substantially with decreasing p^*_{bin} values. For these and the remaining experiments we have extended our evaluation model to take into account that not all nodes will be affected by the routing anomaly. Here, we have used individual threshold $p^*_{ind} = 10^{-6}$ and assumed that 50% of the nodes are effected. This choice is based on the work of Ballani et al. [20], which suggests that the fraction of ASes whose traffic can be hijacked by any AS varies between 38% and 63% for imposture attacks, between 29% and 48% for interception attacks, and the fraction increases to 52-79% for tier-1 ASes. Figure 4(c) shows the impact of the percent of affected nodes (with 40 detector nodes). Note that the alert rates for normal conditions here are the same as for the case when none of the detectors are affected; i.e., the right-most curve in Figure 4(c).

In all of the above cases, there is a substantial region of p^*_{bin} values for which we observe significant differences between the attack detection rates and the alert rates under normal conditions (for that same p^*_{bin} value). Figure 5 better illustrates these differences for the case of interception attacks. (The results for imposture attacks are similar, and have been omitted due to lack of space.) Here, we plot the fraction of successfully detected attacks during interception attacks as a function of the alert rate during normal circumstances, with the threshold value p^*_{bin} as a hidden variable. For example, Figure 5(a) simply combines the information in Figures 4(a) and Figures 4(b) into a more effective representation of the tradeoffs when there are different numbers of detectors.

We note that there are substantial improvements when using additional detectors (Figure 5(a)). For example, with 60 detectors (50% of which are affected) we can achieve a detection rate of 50% while maintaining an alert rate (under normal conditions) below 10^{-4}. This is more than two orders of magnitude better than with a single node (Figure 2), illustrating the power of the information sharing and distributed detection achieved with the CrowdSec approach.

Figure 5(b) shows the tradeoffs for different fractions of affected nodes. Here, 0% affected nodes corresponds to the case when no detector nodes are affected and the 100% case is when all nodes are affected. As expected, the best tradeoffs are achieved when many nodes are affected. However, with as little as 30% affected nodes, substantial improvements in the detection tradeoff curve are possible (difference between 0% and 30% curves) and an attack detection rate of 50% can be achieved with an alert rate of less than 10^{-2}.

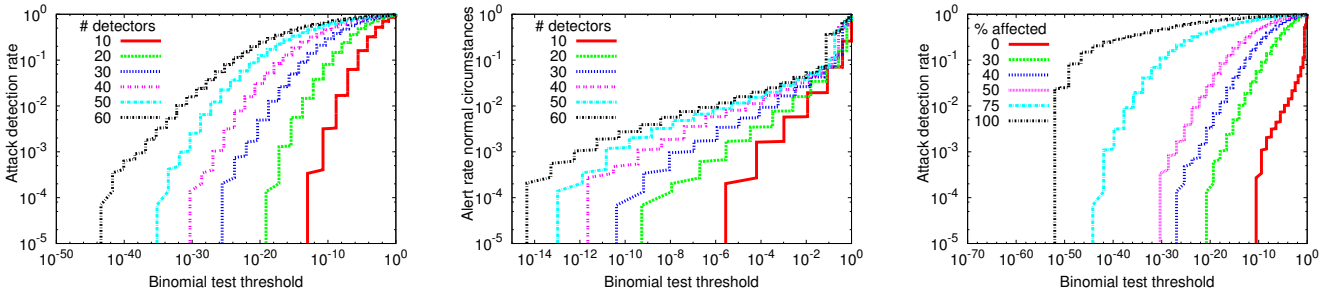
Overall, the low false positive rates (note x-axis on log scale) combined with high detection rates (y-axis on linear scale) suggests that the CrowdSec approach provides an attractive tradeoff as long as there is a reasonable number of affected detectors. Of course, careful data path measurements would always be needed to validate and diagnose the underlying causes behind any detected route anomaly.

Finally, Figure 5(c) shows the impact of the individual threshold value p^*_{ind} used to decide when a candidate anomaly should be reported (and considered in the binomial test). While there are some (smaller) benefits from using larger individual threshold p^*_{ind} values when trying to keep the communication overhead at a minimum (left-most region), we note that there are no major differences in the tradeoffs achieved with different threshold values. Section V takes a closer look at the communication overhead.

IV. Detector Selection

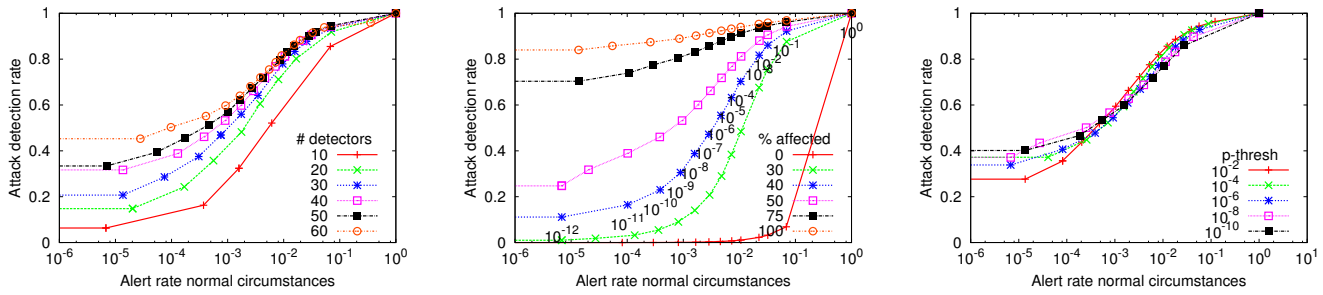
This section looks closer at which nodes are the best detectors and which detectors' alerts are best combined.

Figure 6(a) and 6(b) show the attack detection rate (during attacks) and alert rate during normal circumstances, as reported by nodes at different distances, respectively, when running all possible interception victim-attacker-detector attack combinations within our dataset. Here, we groups triples based on their detector-victim distance $RTT_{d,v}$ and detector-attacker distance $RTT_{d,a}$. Although these distances are most applicable to imposture attacks (where the RTT on the y-axis replaces the RTT on the x-axis), we note that the detection rates during an interception attack (Figure 6(a)) also are high for all values above the diagonal, while none of the buckets result in substantial alert rates under normal conditions (Figure 6(b)). This is encouraging, as it suggests



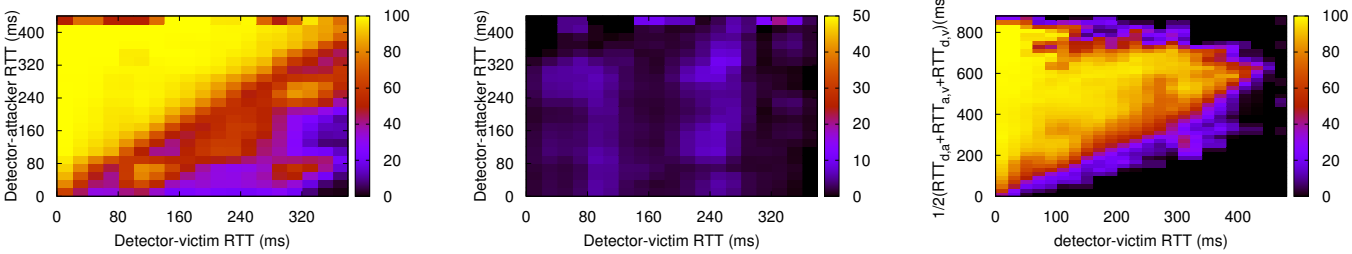
(a) Detectors during attack (b) Detectors normal circumstances (c) Fraction affected during attack

Fig. 4. Alert rates as a function of the binomial threshold p^*_{bin} under different circumstances.



(a) Number detectors (b) Percent affected detectors (c) Individual threshold p^*_{ind}

Fig. 5. Tradeoff between the fraction of detected attacks during interception attacks and the false alert rate during normal circumstances.



(a) Single-hop distance (attack) (b) Single-hop distance (normal) (c) Multi-hop distance (attack)

Fig. 6. Alert rates as function of the relative RTT distances between detector, attacker, and victim.

that it may be possible to use the same set of nodes as detectors for both imposture attacks and interception attacks.

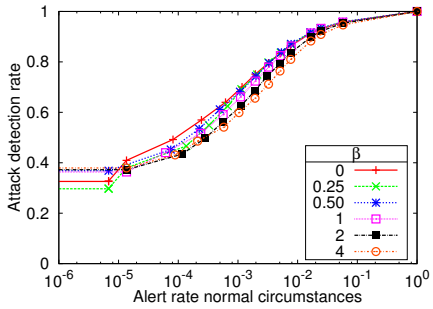
While the distance $\frac{1}{2}(RTT_{d,a} + RTT_{a,v})$ in most cases (except when the triangle inequality is violated) is no less than the original distance $RTT_{d,v}$, we also include a heatmap for the detection rate for these candidate distances in Figure 6(c). As expected only points close to the diagonal (with similar RTTs for the two cases) have intermediate detection rates.

Thus far, we have ignored the relative distances of the affected nodes. We next take a closer look at how the set of affected nodes impacts the best set of detector nodes to combine. For this analysis, we extend our system model to include a parameter β that determines the set of detector nodes most likely to be affected by an attack. In particular, we assume that a detector node $d \in \mathcal{D}$ is affected by an attacker a of victim v with a probability proportional to $\frac{1}{RTT^{\beta}_{d,a}}$. When $\beta = 0$ there are no biases and when $\beta = 1$ the probability is inversely proportional to the RTT distance to the attacker. This bias model is motivated by ASes close to the attacker being most likely to be affected by the attack [20].

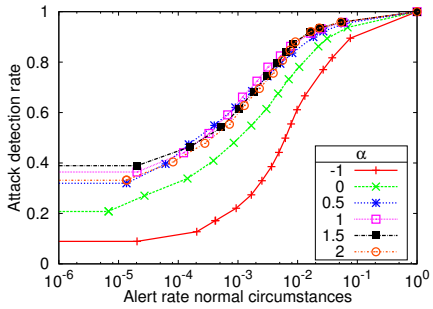
To assess the impact of node selection to protect candidate victims v, we similarly consider candidate policies in which sets of candidate detectors are selected based on their distance to the victim v. We have considered both policies that skew the probability according to a Zipf distribution (with probability proportional to $\frac{1}{r^{\alpha}}$, where r is the rank of the node, after sorting them based on distance to the victim, with rank 1 being closest to the victim) and based on the relative distance to the victim (with probabilities being assigned proportional to $\frac{1}{RTT^{\alpha}_{d,v}}$). As the conclusions are the same for both policy types, we only show results for the rank-based Zipf approach.

Figures 7(a) and 7(b) show the tradeoff in detection rates as a function of the alert rates for different skew (β) in the affected detectors and different Zipf parameters (α) for the selection probabilities, respectively. In these figures we have used $\beta = 1$ and $\alpha = 1$ as our default values. Although the tradeoff curves are better (with higher detection rates, given the same alert rates) when the affected nodes are less biased (small β), the bias (Figure 7(a)) has a relatively modest impact on the tradeoff. This result suggests that our results thus far are robust to which nodes are affected by the attacks.

However, referring to the impact of detector selection (Figure 7(b)), it is interesting to see that there are some additional tradeoffs to take into consideration when selecting detectors for each candidate victim. For example, when low false positives are desired, there are benefits to biasing the



(a) Bias in affected nodes (β)



(b) Bias in detector selection (α)

Fig. 7. Detection tradeoff for different skew in the rates at which nodes are affected and detector nodes selected, respectively. (Default parameters: $\alpha = 1$, $\beta = 1$, 50% affected nodes, and 40 detector nodes.

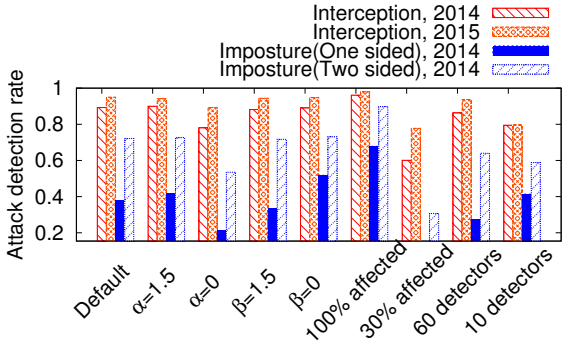


Fig. 8. Attack detection rates under different attacks, while keeping a fixed alert rate of 0.01. Default parameters: $\alpha = 1$, $\beta = 1$, 50% affected nodes, and 40 detector nodes.

selection to intermediate skews. However, the results are relatively insensitive to smaller variations in the chosen bias. In fact, a fairly simple policy that picks detectors uniformly at random ($\alpha = 0$) achieves a relatively nice tradeoff even when there is an underlying bias of which nodes are affected.

Whereas most of the results presented here have been for interception attacks using the Jul/Aug. 2014 iPlane data, we note that the results for imposture attacks are similar and that the results are not limited only to that time period. Figure 8 summarizes our results for both imposture and interception attacks, and for the Jan/Feb. 2015 time period. Here, we show the attack detection rates for example scenarios where we vary one factor at a time (both to a smaller and to a larger value) relative to a default scenario when we fix the alert rates during normal circumstances at 0.01. While the imposture attacks are somewhat more difficult to detect, the general tradeoffs otherwise remain the same for the different parameters.

V. Candidate Architectures and Overhead

This paper describes a general crowd-based approach for passive RTT monitoring and collaborative anomaly detection to help identify routing anomalies. Although we envision that the system would primarily be implemented at stationary end hosts run by concerned citizens, the system could also easily be operated at any node that can perform passive end-to-end RTT measurements, including at operator operated proxies and middle boxes. In this section we discuss the different ways a general CrowdSec system could be implemented.

Monitoring: Host-based packet capture (pcap) tools for network traffic monitoring and analysis is available for Unix-based, Linux-based, and Windows systems. Many of these tools use the libpcap/WinPcap libraries and allow RTT measurements to be passively recorded. For example, generic tools can be created based on the basic per-packet timing information available in standard tools such as Wireshark[3] or `tcpdump`[4]. Intrusion detection systems such as Bro[5], also provide built-in functionality for a wide range of per-packet, per-object, per-connection, or per-host analysis.

Alternatively, browser-based extensions such as Firebug[6] and Fathom [25] could also be used to collect RTT measurements and other connection information. While socket-based methods typically incur lower delay overhead than HTTP-based methods [26], it has been shown that Fathom is able to eliminate most upper-layer RTT measurement overhead [25]. Such browser-based measurement techniques may therefore be an attractive monitoring option, as they allow greater operating system flexibility and simplify ubiquitous deployment.

Architecture: The general CrowdSec approach is not bound to a particular architecture and could be implemented using both centralized and decentralized architectures. At one end, a centralized directory service would allow clients to report their outliers (and corresponding p-values) to a central server that would then perform the "collaborative" outlier detection and apply the Binomial test. This solution is both simple and effective, but places a lot of responsibility at one or more central nodes. At the other end of the spectrum, Distributed Hash Tables (DHTs) such as Chord [27] and Pastry [28] can be used to distribute the workload across participating nodes. In this case, the reports are sent to a "holder" node of a particular prefix, where holders can easily be determined using a prefix-aware DHT [16].

For the collaborative detection calculations, the holder (or server) responsible for each prefix must keep track of outlier reports and maintain an estimate of the total number of current detectors for the prefix. Such estimates can easily be done through periodic or probabilistic detector reporting, for example. Although it is possible to further offload the holder nodes (or centralized directory servers) by moving the collaborative detection calculations to a subset of the detector nodes, these calculations can easily be done at very low overhead on the holder nodes.

[3]Wireshark, https://www.wireshark.org//, May. 2015.
[4]tcpdump, http://www.tcpdump.org/, May. 2015.
[5]Bro, https://www.bro.org/, May. 2015.
[6]Firebug, http://getfirebug.com/, May. 2015.

Detector selection: In the case it is not feasible to maintain reports from all detectors, a subset of detectors may be selected. As we have seen here, there are some benefit tradeoffs based on which detectors are used. Similar to in our simulations, such detector selection could be performed by assigning detector weights based on their relative distance to the monitored prefix (candidate victim). Assuming that detectors report their RTTs (as measured to the monitored prefix), such weighting and/or selection could easily be done at the holders or central servers.

Privacy: Only a subset of RTT measurements will need to be shared by participants. Detectors would not be expected to reveal how often services/prefixes are accessed and should have the option to opt in/out of monitoring selected services/prefixes. This can easily be achieved by ensuring that only the minimum necessary information is shared in the outlier reports. For example, in particularly privacy concerned systems the reports could report only whether a measurement is an outlier or not (not the p-values themselves). Note that as long as all detectors use the same individual threshold p^*_{ind} the Binomial tests applied here would still be directly applicable.

Overhead: Without loss of generality, assume that each detector node $d \in \mathcal{D}$ keeps track of a set $\mathcal{V}_d$ of prefixes (each associated with a candidate victim). Furthermore, let $D = |\mathcal{D}|$ denote the total number of detectors, $\overline{V_d} = \frac{1}{D}\sum_{d\in\mathcal{D}} |\mathcal{V}_d|$ the average number of prefixes monitored per detector node, and $V = |\cup_{d\in\mathcal{D}} \mathcal{V}_d|$ the total prefixes monitored.

Using this notation, a centralized directory would keep track of V prefixes and in a decentralized setting each holder node (assuming all detectors also act as holders) would be responsible for on average $\frac{V}{D}$ prefixes. It is also easy to show that the total number of individual outlier reports inserted into the systems (either at the centralized directory or a holder node, for example) is proportional to $\sum_{d\in\mathcal{D}} |\mathcal{V}_d| = D\overline{V_d}$. For example, in the case that the D detector nodes also act as holder nodes, each holder node would see a reporting rate equal to the average outlier rate observed by a single detector node (such as itself), equal to $\overline{V_d}$ times the individual nodes' per-prefix outlier alert rate.

Furthermore, the total number of active measurements (e.g., traceroutes) needed to further investigate if the data path had in fact been compromised is equal to the product of the the number of detectors per prefix ($\frac{DV_d}{V}$) times the total collaborative alert rate (equal to V times the collaborative per-prefix alert rate). After cancelation of the V terms, we note that the number of active measurements triggered by the system would be proportional to DV_d, but this time with a smaller proportionality constant (equal to the collaborative per-prefix alert rate). Spread over D detector nodes, such measurement effort is very low, especially given the relatively low (compared to a non-collaborative system, for example) collaborative per-prefix alert rates observed with the help of the collaborative techniques discussed in this paper.

VI. Related Work

This section complements the related works discussed in Section II with a discussion of the work in a broader context.

Passive and active end-to-end measurements: Network level end-to-end measurements have been used for a wide range of purposes, including identifying and troubleshooting disruptions on the Internet. For example, NetDiagnoser [29] leverage the end-to-end measurements between sensors and knowledge of the sensor topography to provide a troubleshooting tool that allows ISPs to detect the location of network failures. Others have used active probing techniques to detect Internet outages [30] and other network reachability issues [31]. Passive network measurement data have been used to characterize censorship by nation-states or the effects of natural disasters such as earth-quakes on communication [32]. In contrast, we focus on detecting interception and imposture attacks, which generally are difficult to detect since these attacks do not lead to service outages or interruptions.

Crowd-based techniques: Client-side measurement tools and browser extensions have been designed to measure a wide range performance and security issues, including website performance, port filtering, IPv6 support, and DNS manipulations [25], [33]. Passive monitoring from a large number of users has also been used to characterize geographic differences in the web infrastructure [34], to build location-based services [35], to build AS-level graph [36], and to detect outages and other network events [37]. None of these works consider crowd-based detection of routing anomalies caused by stealthy routing attacks such as interception and imposture attacks.

BGP attack detection: Section II described several data-plane based, control-plane based, and hybrid techniques. Perhaps most closely to our work is the works by Zheng el al. [18] and Shi et al. [21]. Similar to us, Zheng et al. [18] use measurements to detect imposture and interception attacks. However, in contrast to our passive RTT measurements, their framework requires combined control-plane data (route announcement information) and significant active (traceroute) measurements. Shi et al. [21] shows the value of using distributed diagnosis nodes when detecting black-holing attacks. In contrast to these works, we show that seemingly simple measurement data such as RTT collected passively can be effectively used to raise alerts for possible interception and imposture attacks.

VII. Conclusions

In this paper we have presented and evaluated a user-centric crowd-based approach in which users passively monitors their RTTs, share information about potential anomalies, and apply combined collaborative statistics such as the Binomial test to identify potential routing anomalies. We have shown that the approach has low overhead, and provides an attractive tradeoff between attack detection rates (when there is an attack) and alert rates (needing further investigation) under normal conditions. Detection-alert tradeoffs and weighted detector selection, based on candidate detectors' distance from the potential victim nodes, have been evaluated using longitudinal RTT measurements from a wide range of locations. We have shown that there are significant advantages to collaboration (e.g., as shown by the high attack detection rates that can be achieved without adding many additional alerts

during normal circumstances), that there are benefits from introducing a slight bias towards selecting nearby detectors, and that these systems can be effectively implemented using relatively simple monitoring (e.g., on client machines) and data sharing over both central and distributed architectures at a low overhead.

We believe that this type of system can allow concerned citizens to take a greater role in protecting the integrity of their own and others' data paths. Given the slow deployment of operator-driven solutions (typically cryptographic and requiring significant adaptations to work properly), this type of passive crowd-based monitoring provides an effective way to bring traffic anomalies (due to attacks, router misconfigurations, or other valid reasons) to the forefront. As such, this type of crowd-based system can push more operators to take actions to provide better protection for the data traffic paths of their customers and others. Future work includes analyzing the interaction between active and passive measurements to diagnose traffic anomalies. Interesting research questions include determining the best way to combine longitudinal passively collected information (e.g., as collected and analyzed here) with targeted traceroutes and other complementary information (such as topology information) to determine the chain of events that allowed the anomaly to take place.

REFERENCES

[1] K. Butler, T. Farley, P. McDaniel, and J. Rexford, "A survey of BGP security issues and solutions," *Proc. of IEEE*, vol. 98, no. 1, pp. 100–121, Jan. 2010.

[2] S. Goldberg, "Why Is It Taking So Long to Secure Internet Routing?" *ACM Queue*, vol. 12, no. 8, pp. 327–338, Oct. 2014.

[3] R. Hiran, N. Carlsson, and P. Gill, "Characterizing large-scale routing anomalies: A case study of the china telecom incident," in *Proc. PAM*, Mar. 2013.

[4] Dyn Research. (2008) Pakistan hijacks youtube. [Online]. Available: http://research.dyn.com/2008/02/pakistan-hijacks-youtube-1/

[5] A. Arnbak and S. Goldberg, "Loopholes for circumventing the constitution: Unrestrained bulk surveillance on americans by collecting network traffic abroad," in *Proc. HOTPETS*, Jul. 2014.

[6] M. Lepinski and S. Kent, "An Infrastructure to Support Secure Internet Routing," RFC 6480 (Informational), IETF, Feb. 2012.

[7] S. Kent, C. Lynn, and K. Seo, "Secure border gateway protocol (s-bgp)," *IEEE JSAC*, vol. 18, no. 4, pp. 582–592, Apr. 2000.

[8] R. White, "Securing bgp through secure origin bgp," *The Internet Protocol Journal*, vol. 6, no. 3, pp. 15–22, Sep. 2003.

[9] Z. Zhang, Y. Zhang, Y. C. Hu, Z. M. Mao, and R. Bush, "iSPY: Detecting IP Prefix Hijacking on My Own," *ACM CCR*, vol. 38, no. 4, pp. 327–338, Aug. 2008.

[10] J. Karlin, S. Forrest, and J. Rexford, "Pretty good bgp: Improving bgp by cautiously adopting routes," in *Proc. IEEE ICNP*, Nov. 2006.

[11] Dyn Research. (2013, Nov.) The new threat: Targeted internet traffic misdirection. [Online]. Available: http://research.dyn.com/2013/11/mitm-internet-hijacking/

[12] M. Lepinski, "Bgpsec protocol specification," Informational, IETF, Nov. 2013.

[13] R. Arends, R. Austein, M. Larson, D. Massey, and S. Rose, "DNS Security Introduction and Requirements," RFC 4033 (Proposed Standard), IETF, Mar. 2005.

[14] J. Gersch and D. Massey, "Rover: Route origin verification using dns," in *Proc. IEEE ICCCN*, Jul/Aug. 2013.

[15] D. Cooper, E. Heilman, K. Brogle, L. Reyzin, and S. Goldberg, "On the risk of misbehaving rpki authorities," in *Proc. ACM HotNets*, Nov. 2013.

[16] R. Hiran, N. Carlsson, and N. Shahmehri, "Prefisec: A distributed alliance framework for collaborative bgp monitoring and prefix-based security," in *Proc. ACM CCS WISCS*, Nov. 2014.

[17] A. Haeberlen, I. Avramopoulos, J. Rexford, and P. Druschel, "NetReview: detecting when interdomain routing goes wrong," in *Proc. NSDI*, Apr. 2009.

[18] C. Zheng, L. Ji, D. Pei, J. Wang, and P. Francis, "A light-weight distributed scheme for detecting ip prefix hijacks in real-time," in *Proc. ACM SIGCOMM*, Aug. 2007.

[19] M. Lad, D. Massey, D. Pei, Y. Wu, B. Zhang, and L. Zhang, "Phas: a prefix hijack alert system," in *Proc. USENIX Security Symp.*, Jul. 2006.

[20] H. Ballani, P. Francis, and X. Zhang, "A study of prefix hijacking and interception in the internet," *ACM CCR*, vol. 37, no. 4, pp. 265–276, Aug. 2007.

[21] X. Shi, Y. Xiang, Z. Wang, X. Yin, and J. Wu, "Detecting prefix hijackings in the internet with argus," in *Proc. ACM IMC*, Nov. 2012.

[22] P. Gill, M. Arlitt, N. Carlsson, A. Mahanti, and C. Williamson, "Characterizing organizational use of web-based services: Methodology, challenges, observations, and insights," *ACM TWEB*, vol. 5, no. 4, pp. 19:1–19:23, Oct. 2011.

[23] F. E. Grubbs, "Sample criteria for testing outlying observations," *The Annals of Mathematical Statistics*, vol. 21, no. 1, pp. 27–58, Mar. 1950.

[24] H. Madhyastha, T. Isdal, M. Piatek, C. Dixon, T. Anderson, A. Krishnamurthy, and A. Venkataramani, "iplane: An information plane for distributed services," in *Proc. OSDI*, Nov. 2006.

[25] M. Dhawan, J. Samuel, R. Teixeira, C. Kreibich, M. Allman, N. Weaver, and V. Paxson, "Fathom: A browser-based network measurement platform," in *Proc. ACM IMC*, Nov. 2012.

[26] W. Li, R. K. Mok, R. K. Chang, and W. W. Fok, "Appraising the delay accuracy in browser-based network measurement," in *Proc. ACM IMC*, Oct. 2013.

[27] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan, "Chord: A scalable peer-to-peer lookup service for internet applications," in *Proc. ACM SIGCOMM*, Aug. 2001.

[28] A. I. T. Rowstron and P. Druschel, "Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems," in *Proc. IFIP/ACM Middleware*, Nov. 2001.

[29] A. Dhamdhere, R. Teixeira, C. Dovrolis, and C. Diot, "Netdiagnoser: Troubleshooting network unreachabilities using end-to-end probes and routing data," in *Proc. ACM CoNEXT*, Dec. 2007.

[30] L. Quan, J. Heidemann, and Y. Pradkin, "Detecting internet outages with precise active probing," USC/Information Sciences Institute, Tech. Rep., May 2012.

[31] E. Katz-Bassett, H. V. Madhyastha, J. P. John, A. Krishnamurthy, D. Wetherall, and T. Anderson, "Studying black holes in the internet with hubble," in *Proc. NSDI*, 2008.

[32] A. Dainotti, R. Amman, E. Aben, and K. Claffy, "Extracting benefit from harm: using malware pollution to analyze the impact of political and geophysical events on the Internet," *ACM SIGCOMM CCR*, vol. 42, no. 1, pp. 31–39, Jan. 2012.

[33] C. Kreibich, N. Weaver, B. Nechaev, and V. Paxson, "Netalyzr: Illuminating the edge network," in *Proc. ACM IMC*, Nov. 2010.

[34] M. Arlitt, N. Carlsson, C. Williamson, and J. Rolia, "Passive crowd-based monitoring of world wide web infrastructure and its performance," in *Proc. IEEE ICC*, Jun. 2012.

[35] P. Shankar, Y.-W. Huang, P. Castro, B. Nath, and L. Iftode, "Crowds replace experts: Building better location-based services using mobile social network interactions," in *Proc. IEEE PerCom*, Mar. 2012.

[36] A. Faggiani, E. Gregori, L. Lenzini, V. Luconi, and A. Vecchio, "Smartphone-based crowdsourcing for network monitoring: Opportunities, challenges, and a case study," *IEEE Communications*, vol. 52, no. 1, pp. 106–113, Jan. 2014.

[37] D. R. Choffnes, F. E. Bustamante, and Z. Ge, "Crowdsourcing service-level network event monitoring," *ACM SIGCOMM CCR*, vol. 40, no. 4, pp. 387–398, Oct. 2010.