Public Review for

Scanning the Internet for Liveness

S. Bano, P. Richter, M. Javed, S. Sundaresan, Z. Durumeric, S. Murdoch, R. Mortier, V. Paxson

Internet-wide scanning relies heavily on the notion of the "liveness" of IP addresses, i.e., whether an IP address can respond to a probe packet or not. Despite its importance, the definition of liveness is often not well understood. This paper presents a taxonomy of liveness as observed at different layers of the network stack (network, transport, and application layers) using suitable probing mechanisms (e.g., ICMP at the network layer vs. TCP probes at the transport layer). With the help of the proposed taxonomy, we can identify whether the network layer indicates that a host is alive, and the transport and application layers indicate whether they are active and operational. Based on their taxonomy, the authors develop a measurement method, and incorporate it in the ZMap tool, to conduct a scan of the IPv4 address space. Their results include a number of surprising findings, such as HTTPS being more responsive than HTTP, and a large size of CWMP alive population. The authors have made their measurement results as well as the modified ZMap tool available to the research community. Overall, the reviewers felt that this paper can be highly useful for those who want to conduct liveness experiments over the Internet, and can lead to interesting future research efforts related to cross-layer liveness, and in gaining deeper understanding of the operational aliveness of different applications.

Public review written by
Fahad Dogar
Tufts University

Scanning the Internet for Liveness

Shehar Bano
University College London

Philipp Richter
MIT

Mobin Javed
LUMS Pakistan, ICSI Berkeley

Srikanth Sundaresan
Princeton University

Zakir Durumeric
Stanford University

Steven J. Murdoch
University College London

Richard Mortier
University of Cambridge

Vern Paxson
UC Berkeley, ICSI Berkeley

ABSTRACT

Internet-wide scanning depends on a notion of *liveness*: does a target IP address respond to a probe packet? However, the interpretation of such responses, or lack of them, is nuanced and depends on multiple factors, including: how we probed, how different protocols in the network stack interact, the presence of filtering policies near the target, and temporal churn in IP responsiveness. Although often neglected, these factors can significantly affect the results of active measurement studies. We develop a taxonomy of liveness which we employ to develop a method to perform concurrent IPv4 scans using ICMP, five TCP-based, and two UDP-based protocols, comprehensively capturing all responses to our probes, including negative and cross-layer responses. Leveraging our methodology, we present a systematic analysis of liveness and how it manifests in active scanning campaigns, yielding practical insights and methodological improvements for the design and the execution of active Internet measurement studies.

CCS Concepts

•**Networks** → **Signaling protocols; Transport protocols; Application layer protocols; Network dynamics;** *Cross-layer protocols;*

Keywords

Active Measurement, Scanning, Cross-protocol, Census

1. INTRODUCTION

Internet-wide scanning has emerged as a key measurement technique to study a diverse set of the Internet's properties, including address space utilization [5, 16], host reachability [4], topology [6, 36, 18], service availability [20, 21], vulnerabilities [11, 17, 23], and service discrimination [19]. In simplest terms, active scanning campaigns involve the sending of one or more *probe packets* to a target IP address and observing a response (or absence thereof) from the targeted host. If a host replies to a probe packet, we refer to it as *alive*. Individual measurement campaigns (see above) are typically crafted to elucidate individual properties of the Internet and its host population. Yet despite the widespread use of active scanning and its critical importance for Internet measurement, we still lack a systematic framework that allows us to understand IP *liveness* and, more importantly, how it manifests in the form of host replies to active probing. What type of probe packets should we send if we, for example, want to maximize the responding host population? What type of responses can we expect and which factors determine such responses? What degree of consistency can we expect when probing the same host with different probe packets?

Fundamentally, *liveness* is not a straightforward binary matter, but varies depending on: (*i*) probe type and the target or target network's policies related to firewalling and filtering, (*ii*) temporal churn due to targets going up and down, and (*iii*) protocol interdependencies that result in probes to one target eliciting responses from another (e.g., ICMP Error responses to TCP probes). While seemingly nuanced, such characteristics have the potential to significantly affect the result of active measurement campaigns. We argue that developing a systematic understanding of these issues yields methodological improvements and practical implications for active measurements at large. Towards achieving this goal, we make the following contributions in this paper.

First, we propose a taxonomy of *liveness*, examining what it means to say that a target IP address is alive and how liveness can be inferred considering responses to active probing (§2). This taxonomy develops our understanding of liveness at different layers, and covers responses across protocols and from non-targets (e.g., middleboxes). Informed by our taxonomy, we introduce a methodology for performing Internet-wide scans *concurrently* across a set of different protocols at various layers, including ICMP, popular TCP services and popular UDP services (§3). Our diverse set of probe packets allows us to study the responsiveness or non-responsiveness of individual host populations to specific protocols. Our scans are *comprehensive* in that we capture all replies to our probe packets, including negative (e.g., TCP Rst packets) as well as cross-layer replies (e.g., ICMP error replies). This enables us to uncover otherwise invisible host populations and to study cross-layer protocol interactions. Based on our gathered data, we present an in-depth view of *liveness* (§4), slicing our analysis along two dimensions: (*i*) *probe type* (i.e., what type of packet we send), and (*ii*) *reply* (i.e., what kind of response we receive).

Our analysis yields important insights for both the design of active scanning campaigns as well as the interpretation of scanning results. Our key findings include: (*i*) TCP and UDP probes increase the population responsive over ICMP by 18%, (*ii*) comprehensively capturing reply traffic (i.e., taking into account negative reply packets) increases the responsive population by more than 13%, (*iii*) TCP stacks do not consistently respond with a TCP Rst for non-available services—in our measurements only 24% of hosts with an active TCP stack respond to all the probes, (*iv*) our concurrent scans allow us to identify nearly 2M tarpits that would bias measurements that do not take them into account, and (*v*) we report on the correlation of responsiveness across protocols uncovering potential filtering practices.

We believe that our measurements paint the most comprehensive and least noisy picture of the state of Internet liveness available to date. Our taxonomy of IP liveness can serve as a basis for designing and executing future measurement studies, particularly when it

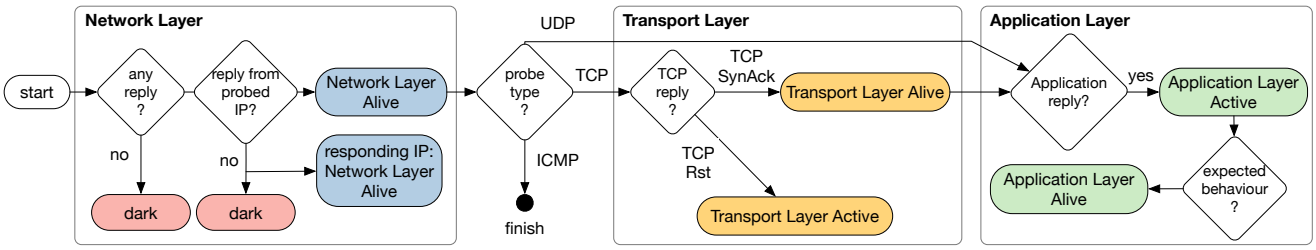


Figure 1: Flow chart showing liveness inference. We consider liveness at different layers based on responses to our probes.

comes to decisions such as what type of probe packets should be employed and what type of responses should be captured, how to interpret responses, as well as whether it is appropriate to use the output of one scan as input for subsequent measurements. We release the code and data of this work as open source to allow for reproducibility of the results, and to enable further research.

2. TAXONOMIZING IP LIVENESS

To systematize our understanding of IP liveness and its inference using active probing, we introduce the following terminology.

Network-layer liveness. An IP address is network-layer *alive* if it responds to a probe with an IP packet. This is the most basic liveness criterion.

Transport-layer liveness. An IP address is considered transport-layer *active* if it is capable of sending TCP packets (whether SynAck or Rst) from at least one port. That is, the transport protocol stack of the IP address is operational, sending packets with transport layer semantics. An IP address is transport-layer *alive* if it accepts TCP connections on a specific port number, indicated by a SynAck response to a probe.

Application-layer liveness. Application-layer *active* means that the IP address sends a payload, valid or invalid, for at least one application protocol. An IP address is application-layer *alive* if it speaks the probed application protocol.

Figure 1 shows a flowchart of our methodology for inferring liveness, which depends both on the *probe type* and the *reply*, per our taxonomy above. Observe that probes of different types have varying degrees of specificity (e.g., a TCP Syn probe targets a specific port number, whereas an ICMP Echo probe targets a vanilla IP address), as well as different inference power: ICMP Echo requests can only infer network layer liveness, while TCP probes can infer transport layer liveness in addition. The dependency on the *probe type* is important to realize, as different probes can reveal different views of liveness at a given layer. For example, IP addresses might not directly reveal network layer liveness (e.g., we might not receive an ICMP EchoReply in response to an ICMP Echo probe), while we can observe them to be transport layer alive using TCP probes, indirectly inferring their network liveness. Since UDP is connectionless and relies on ICMP for negative replies, we do not have an explicit notion of transport layer liveness for UDP. Our UDP probes contain application-specific requests, but the corresponding responses can indicate network- or application-layer liveness (i.e., ICMP Error replies indicate network layer liveness while UDP replies can indicate application layer liveness and network layer liveness).

In the remainder of this work, we employ this taxonomy to analyze liveness through active probing scans. We focus our analysis on network and transport layers, but include application layer liveness in the taxonomy for completeness.

3. SCANNING IP LIVENESS

In this section, we discuss our experimental setup, and scanning methodology and considerations. We focus on intra-scan liveness, that is how the visible IP population varies when considering different probe types and captured responses across the same scan campaign. Characterization of long-term temporal churn (i.e., the birth and death of responsive IP addresses) and diurnal behaviour, as well as spatial churn (i.e., difference in responses due to diverse scan locations) is outside the scope of this work.

Overview. We collect data from simultaneous scans covering 8 protocols performed on September 5, 2017. For validation, we performed the same scans also on 4th and 7th September, finding consistent results. We target the entire IPv4 address space, less a blacklist covering 14.7% of the total IPv4 address space. The blacklist includes private and reserved space (covering 14% of IPv4) and users opting out of measurements (0.7% of IPv4 at the time of measurement). The entire scan takes less than 24 hours to complete and generates 2.3 TB of data. We select ports that run well-documented applications, and cover both the server as well as (at least partially) the client space. We restrict the analysis to eight concurrent scans for feasibility. We perform one network layer ICMP Echo scan, five transport layer TCP Syn scans covering popular ports 22 (SSH), 23 (Telnet), 80 (HTTP), 443 (HTTPS), and 7547 (CPE WAN Management Protocol, CWMP [3]) and two popular UDP-based applications, DNS and NTP.

Tools. We use ZMap [40] for the network, transport, and UDP-based application layer scans. ZMap uses raw sockets, crafting ICMP Echo, TCP Syn and UDP packets embedded directly into Ethernet frames. In contrast to earlier studies using ZMap, we tightly synchronize simultaneous scans of several protocols and probe each IP address for all protocols within a short time window, minimizing the effect of temporal churn. Moreover, we customize ZMap so as to capture *any reply* to our probes including negative responses such as ICMP errors and TCP Rsts. We use SiLK [38] for scan data analysis. We convert all IP sets of interest to SiLK IPset data structure that uses a compressed binary tree structure to store IPs. We mainly use relevant SiLK tools (such as `rwsettool`) to perform fast set operations on this data.

Cross-protocol scans. Internet liveness legitimately changes over time due to temporal churn, for example caused by hosts going up and down and dynamic IP address assignment [32]. To minimize the effect of temporal churn, we probe each IP address for all of our protocols within a short time window. To do so, we partition the IPv4 space into /3 blocks, and configure parallel scans to probe IPs in the same order by using the same seed value for the ZMap address generator within each block. This block-scanning strategy provides an opportunity to synchronize at the start of each new block. We quantify the resulting lag by recording the timestamp of every millionth packet sent by ZMap to measure the maximum time between different protocol probes sent to the same target. While it

can be up to 25 minutes, over 80% of probes are sent to the target within a 10 minute window.

Reply capture completeness. To provide a comprehensive picture of liveness across different layers, we must capture all replies to our probe packets, not just those expected for a successful response. However, by default ZMap does not record ICMP error messages for TCP scans, and only records ICMP PortUnreachable responses for UDP. We modified ZMap to capture all ICMP error messages in response to our probes. We link these error responses back to the probe that generated them by checking the header of the original packet, included in the payload of the error-response packets. Since stock ZMap only records TCP Rst packets with the Ack bit set (as others fail ZMap's validation checks for a deterministic Ack number), we also modified it to record all TCP Rst packets.

Packet loss mitigation. Executing multiple concurrent scans on shared hardware increases the risk of packet loss, at the host and due to transient network problems. To minimize in-network loss, we probe redundantly. Since network losses often occur in bursts, we modify ZMap to perform *delayed retransmissions*. We make ZMap store each IP address it scans for the first time in a queue of size N. When the queue is full, ZMap begins de-queuing and retransmitting, interleaving with new probes. The delay between the original and retransmitted probes depends on the size of the queue and ZMap's packet-sending rate. We use $N = 1\text{M}$ and a scan rate of 100kpps, giving a delay of 20 seconds. Our probe redundancy increases the population of active IP addresses (i.e., responsive to ICMP Echo probes) by 2.2%. We determine this percentage by running two ZMap instances (with no retransmission) in parallel, with a delay of 1 minute between them. To calculate the retransmission hit-rate, we aggregate responsive IPs across both scans, while we only consider the first scan to determine the hit-rate of single-packet transmission. Additionally, we employ additional `pcap` filters in ZMap and adjust host buffer sizes to avoid local packet loss when running multiple ZMap instances on the same host. We inspected scan monitoring logs to confirm that no scans experienced drops at the NIC or in `pcap`.

4. CHARACTERIZING IP LIVENESS

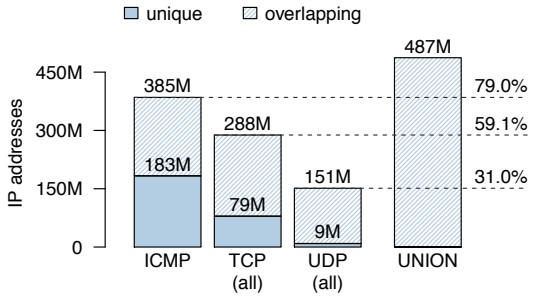
We next analyze our scan data to answer a number of practical questions on how the probe type and the corresponding responses affect the view of liveness at the network and transport layers.

4.1 Network Layer Liveness

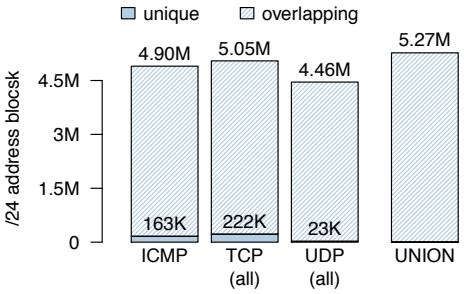
Overall, our scans recorded 487M network alive IPs (IP_{all}) out of 3.6B probed.

What is the coverage of different probe types?

Reachability, performance, and topology studies often employ ICMP Echo and traceroute to scan the Internet for network aliveness. Here, we investigate the effect of the probe type on the measured network alive population: Figure 2a shows the coverage of IP_{all} over different scans by protocol. *Overlapping* IPs potentially respond over multiple probe types while *unique* IPs respond to only one probe type. As others have found [16, 21, 22, 4], we see that ICMP Echo probes are most effective in discovering network active IPs, revealing 79% of IP_{all}, followed by TCP probes. UDP probes, however, illuminate a very restrictive view. Further we find that 16% of IP_{all} can only exclusively be discovered via TCP, and a small but significant $\approx 2\%$ can only be discovered via UDP probes. The high percentage of exclusive coverage from TCP is somewhat surprising, suggesting widespread filtering/firewalling of ICMP traffic within networks and at target hosts. A number of studies measure network aliveness at the granularity of /24 address



(a) Network layer alive IP addresses.



(b) Network layer alive /24 blocks.

Figure 2: Network layer aliveness inferred by scan types.

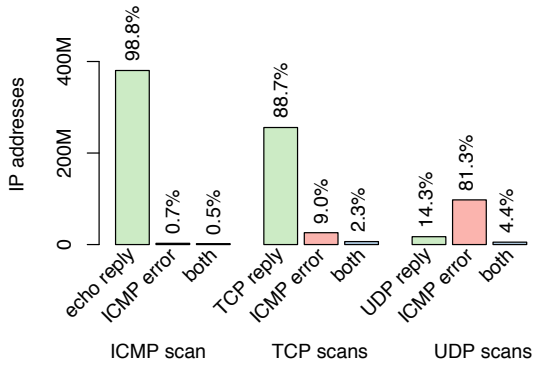
blocks [13, 6, 36, 18]. Figure 2b shows the aliveness breakdown for /24 blocks, where we find that the effect of the probe type is much less pronounced. The set of /24 blocks discovered by individual probes is far more uniform in its coverage of */24*$_{all}$ (the set of all discovered /24s). Surprisingly, our TCP scans show the highest coverage, discovering some 5M active /24 blocks, slightly more ($\approx 3\%$) than ICMP Echo.

What is the coverage of different probe responses?

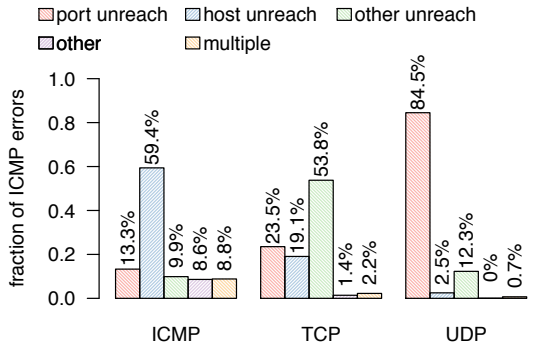
A probe can trigger multiple types of responses, for example TCP Syn can trigger TCP SynAck, TCP Rst, or ICMP Error responses. Interpretation of network aliveness depends on what responses are captured. In the previous analysis, we aggregated all replies per scan, for example an ICMP response to a TCP Syn probe is treated equivalently to a SynAck response. Figure 3a decomposes scan replies to characterize their contribution to the overall scan coverage of IP_{all}. Replies to ICMP Echo probes are dominated by ICMP EchoReply as we would expect. However, ICMP Error responses comprise a sizable portion of TCP, and are the dominant means of inferring network aliveness via UDP. We find that 2.3% of IP_{all} are discoverable only through ICMP Error responses, with ICMP probes lighting up 20% of such IPs, TCP probes some 76%, and UDP probes 35%. This might be due to routers and middleboxes that are configured to ignore direct probes but indirectly reveal activity via ICMP Error packets [15], as well as due to filtering and firewalling in networks and end hosts.

What do ICMP error responses reveal?

ICMP Error messages, even though often neglected, not only increase the visible population of network alive IP addresses, but can also reveal characteristics of the target host and network. In Figure 3b, we break down ICMP Error messages into four categories: (*i*) ICMP PortUnreachable is a type-3 (Destination unreachable) message typically generated by end hosts when a port is not active, (*ii*) ICMP HostUnreachable is a type-3 message sent by gateway devices (e.g., routers) when the host is unreachable, (*iii*) ICMP OtherUnreachable represents all other type-3 messages sent by

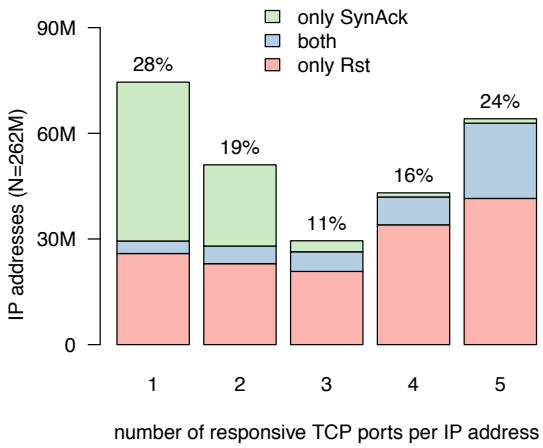


(a) Breakdown of responses to scan types.

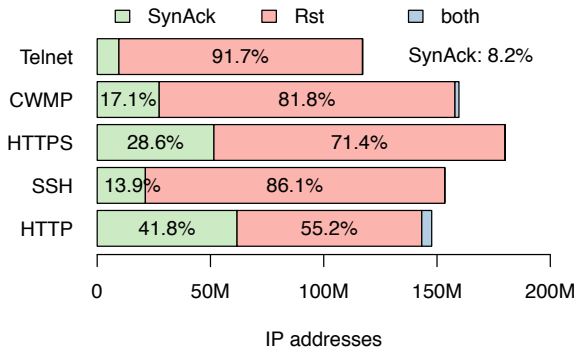


(b) Breakdown of ICMP Error responses to scan types.

Figure 3: Breakdown of responses to scan types.



(a) TCP stack completeness/consistency.



(b) Breakdown of transport layer responses.

Figure 4: Transport layer liveness.

gateway devices when the destination is unreachable (e.g., protocol unreachable [27]), and (*iv*) ICMP Other represents the remaining three ICMP Error packets (ICMP TimeExceeded, ICMP Redirect, and ICMP SourceQuench) that we observed in our data. We consider the source IP of any of the above packets as network alive even if it is generated by an IP other than the one we probed.

Our TCP and UDP scans generate the majority of ICMP Port-Unreachable messages, typically generated directly by the end host. Indeed, it is expected behavior (RFC 1122) that hosts generate such messages if no service is a available on a given port number. However, our scans also resulted in a large number of ICMP Error messages that were generated by intermediate devices on the path towards the target. The majority of such messages are indicative of network misconfigurations, or firewalling. The latter is very prominent: among the ICMP OtherUnreachable messages for TCP and UDP, we find that code 13 "Communication administratively prohibited" dominates (representing about 80% of such messages), hinting towards either routers or gateways (e.g., Carrier-Grade NAT deployments [33, 37]) on the path towards the target. Another 15% of ICMP OtherUnreachable messages correspond to code 0 "Destination network unreachable" and code 10 "Host administratively prohibited". In future work, we plan to inspect ICMP messages more closely to reason about firewalling.

4.2 Transport Layer Liveness

Recall that we measure transport layer liveness by conducting TCP SYN scans on five different ports. In total, we find 262M transport active IPs (i.e., those responding to TCP Syn with a TCP Rst or TCP SynAck) representing 53.8% of IP_{all}.

How does the probed port affect the responsive population?

If hosts responded consistently across TCP ports, we would expect to see the same number of transport active IPs across all five scans since an IP would respond with either TCP SynAck or TCP Rst for each probe, in accordance with RFC standards. We find, however, that the number of active hosts varies vastly when probed on different port numbers. Figure 4a breaks down transport active IPs into 5 classes: IPs that responded only on one probed port, IPs that responded on exactly 2 ports, and so forth. We also show whether the responses were TCP SynAck, TCP Rst, or both, per class. Only 24% of active hosts respond to probe packets on all five ports. This, in turn, shows that the vast majority of hosts selectively suppress responses for particular application protocols, due to firewalling and/or filtering. Their visibility or non-visibility in active scanning campaigns heavily depends on the choice of the probed port numbers. Following up on this observation, we next look at the coverage of the 262M TCP active IP address population by protocol and response type (Figure 4b). HTTPS is the most active port number, with 180M IPs, surprisingly followed closely by CWMP with 159M IPs. We find that the HTTP port is surprisingly less active than the HTTPS port, and the Telnet port shows the least activity of all the probed protocols. Each of the probed ports contributes a unique set of otherwise unresponsive hosts: some 11.5% of all TCP activity can exclusively be found by probing the CWMP

port. SSH, HTTP, and HTTPS provide unique coverage of 3–6% of active IPs, while the exclusive coverage of Telnet is low (0.8%).

What is the coverage by probe response type?

As introduced in our taxonomy (§2), we make a distinction between transport layer activity and aliveness per RFC 793: TCP stacks should respond to TCP Syn probes with a TCP SynAck if a service is listening on the probed port, or TCP Rst otherwise [28]. We term the subset of the transport active population that responds with a TCP SynAck as 'TCP alive', indicating a service is running on that port. Figure 4b shows that except for HTTP, for a given protocol, the TCP alive population is vastly smaller than the TCP active population on that port. Hence, negative replies (TCP Rst) are crucial for capturing the population of TCP active hosts comprehensively. We also find surprising results regarding the TCP alive (i.e., replying with a TCP SynAck) population: the size of the of CWMP alive population is surprisingly large, and as a point of comparison, it is greater than the SSH alive population. CWMP provides means for remote management of end-user devices such as modems, routers, gateways, set-top boxes, and VoIP-phones [7]. A possible explanation could be related to widespread distribution of CWMP-speaking CPE devices by ISPs.

How do fabricated responses affect the measured population?

One consideration in enumerating the TCP alive population of any given protocol are network tarpits: IPs masquerading as fake hosts, responding positively with a TCP SynAck to all TCP Syn probes [1]. We discovered 1.9M transport alive IPs that appear in all TCP scans. To confirm that these are tarpits, we scanned these IPs on a random high port six days after the original scan—89% responded positively, strengthening our belief that these are tarpits. (Further analysis of the identified potential tarpits would require studying the application-layer behaviour—or absence thereof—of the concerned hosts, which we will undertake in future work.) If, as is common practice, transport alive IPs are taken as a proxy for the service population (e.g., IPs that respond to TCP Syn probe on port 80 with SynAck represent Web servers), then the 1.9M tarpit IPs inflate HTTP, HTTPS, and CWMP footprints by 3–4% of their original size, SSH by 10% and Telnet as high as 23%. To mitigate bias due to tarpits in studies conducting transport-layer measurements from which to make application-layer inferences, simultaneously probing a high random port number for liveness can aid in identifying such instances.

4.3 Cross-protocol Liveness

In this section, we investigate what fraction of the host population that responds to a certain probe/protocol can also be captured when probed for different protocols. Understanding these interdependencies is vital for designing multi-stage scanning campaigns, as well as for understanding consistency in filtering behavior across protocols. Figure 5 shows the conditional probabilities for activity (which includes both positive and negative responses) of our probed TCP- and UDP-based protocols. For ICMP, we consider network-layer aliveness (i.e., IPs that respond with ICMP EchoReply). We make several observations: the bottom-most row shows that a significant fraction of transport active hosts (26% on average for TCP services and 12% for UDP) cannot be discovered via ICMP. This is an important consideration, given that it is common practice to use the subset of ICMP-alive IP addresses for further scans, e.g., to measure service availability. Correlation across TCP and UDP protocols is generally lower when contrasted to protocols *within* each family. Secondly, the TCP and UDP blocks indicate varying degrees of correlation in filtering behavior across services, when seen pair-wise. On one hand, we find consistent filtering practices: for example, a Telnet-active host is very likely to elicit a response

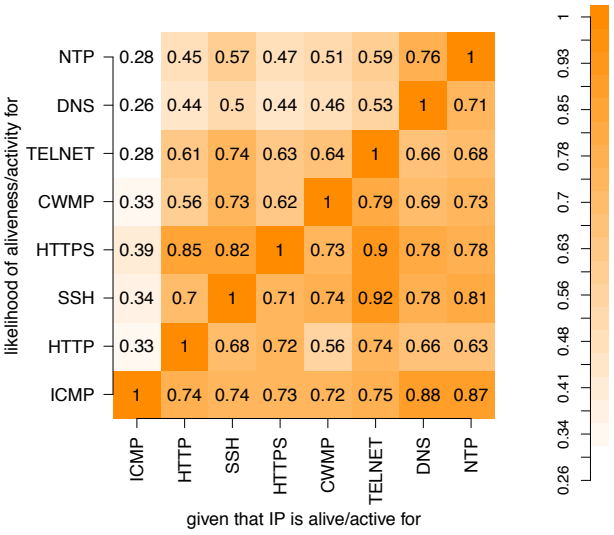


Figure 5: Conditional activity per probe type. (For ICMP, we consider network-layer aliveness.)

from both SSH and HTTPS. Put another way, if a given host is active for Telnet, then with high probability (>=0.9), it is active per SSH and HTTPS. On the other hand, for CWMP only 56% of active hosts respond to HTTP probes, indicating an underlying filtering pattern of the CWMP-active population. We plan to investigate cross-protocol filtering practices in more depth in future work.

5. RELATED WORK

Measurement of Internet liveness has received considerable attention in contexts including network topology, performance and reachability [16, 30, 36, 35, 6], outages [31, 30], service characterization [25, 20, 29], security vulnerability tracking [11, 17, 23], and service discrimination [19]. Measurement studies relied on passive vantage points [8, 9, 32], active probing [21, 12, 16, 24], and both in combination [2]. We focus here on inference of liveness via active probing. Internet-wide scanning has historically taken significant time and resources. Early work limited its scope to BGP prefixes [39, 21, 36], though subsequent work demonstrated the inadequacy of control-plane data to measure data-plane conditions [4, 34]. Another way to limit the scope of active probing is to use "hitlists" to target active hosts. Early hitlists comprised IP addresses selected from various passive sources [26, 18]. Later work shifted towards more informed random selection of target IP addresses offering greater coverage and higher likelihood of liveness [6, 14], and using Internet censuses to derive responsive, complete, and stable hitlists [13]. Heidemann et al. conducted the first full IPv4 scan over the course of 2–3 months in 2007 [16]. IRLScanner scans the advertised IPv4 address space, obtained from a RouteViews BGP dump and the local border router, in approximately 24 hours [21]. Recently, ZMap [12] and its application layer counterpart ZGrab [10] dramatically reduced the time to complete a full IPv4 scan to a few hours. These tools operate on commodity hardware, and data from regular Internet scans using these tools is made publicly available at `scans.io`. These data enabled a large number of follow-on Internet-wide security-modeling and performance studies. All these studies discuss practical considerations in

active probing, such as temporal churn, the types of probes, firewalls, and the scanning tool itself triggering blocking.

Our work adds to this rich body of literature by systematically examining how liveness manifests over different protocols and across layers with active probing, the factors affecting these views, and how they are correlated.

6. CONCLUSION

Liveness—whether or not a target IP address responds to a probe packet—is a nuanced concept without a simple yes/no answer. Responsiveness directly depends on the probe type, the configuration of the targeted host, as well as on firewalling and filtering behaviors at the edge or within networks. The interpretation of responses (positive, negative, absent) in turn allows for drawing conclusions about liveness on different layers. Towards the goal of systematically understanding these issues, we presented a taxonomy of liveness that encapsulates the inherent dependencies between different protocols and layers. We developed and evaluated a methodology for performing concurrent Internet-wide scans across multiple protocols, comprehensively capturing positive, negative, and cross-layer responses to our probes. We find that responsive host populations are highly sensitive to the choice of probe: while ICMP discovers the highest number of raw IPs, our TCP and UDP measurements exclusively contribute a fifth to the total population of responsive hosts. Collecting ICMP Error messages for TCP and UDP scans significantly improves coverage and provides new opportunities to interpret scan results. At the transport layer, our concurrent measurements reveal that the majority of hosts exhibit inconsistent behavior when probed on different ports and that capturing negative responses significantly improves scanning completeness. Our study of cross-protocol liveness shows that, while responsiveness for protocols is correlated, using the result of one scan to bootstrap another should be taken with care, since every probe type introduces an individual bias.

In the future, we plan to deepen our understanding of active scanning in multiple dimensions, looking at: (*i*) liveness at the application layer, (*ii*) how liveness varies over time and IP space, and (*iii*) the multivariate probability distributions of transport layer liveness, and exploring using existing results and their correlations to reduce scan traffic.

Acknowledgments

Shehar Bano was supported by The EU H2020 DECODE project under grant agreement number 732546 and in part by EPSRC Grant EP/N028104/1 'Glass Houses'. Steven J. Murdoch and Shehar Bano (for part of this work) were supported by The Royal Society [grant number UF110392]; Engineering and Physical Sciences Research Council [grant number EP/L003406/1]. Philipp Richter was supported by the MIT Internet Policy Research Initiative, William and Flora Hewlett Foundation grant 2014-1601. Richard Mortier was supported by grant EPSRC EP/K031724/2. Vern Paxson was supported by NSF grants CNS-1237265 and CNS-1518921. Thanks to Jonathan Spring for his valuable feedback. Thanks to the SysAdmins at University College London, especially John Andrews, for their support.

Source code and data release

The source code of our modifications to ZMap, scripts to run block-wise scans, and analysis can be found at https://github.com/sheharbano/scan_liveness (and also https://doi.org/10.5281/zenodo.1209947). Data created during this research is available at https://doi.org/10.5281/zenodo.1068899.

7. REFERENCES

[1] Lance Alt, Robert Beverly, and Alberto Dainotti. Uncovering Network Tarpits with Degreaser. In *Proceedings of the 30th Annual Computer Security Applications Conference*, ACSAC '14, New Orleans, Louisiana, USA, 2014.

[2] Genevieve Bartlett, John Heidemann, and Christos Papadopoulos. Understanding Passive and Active Service Discovery. In *Proceedings of ACM IMC 2007*, San Diego, California, USA, 2007.

[3] John Blackford and Mike Digdon. CPE WAN Management Protocol. Technical Report TR-069, Broadband Forum, November 2013. Issue 1 Amendment 5. CWMP v1.4.

[4] Randy Bush, Olaf Maennel, Matthew Roughan, and Steve Uhlig. Internet Optometry: Assessing the Broken Glasses in Internet Reachability. In *Proceedings of ACM IMC 2009*, Chicago, Illinois, USA, 2009.

[5] Xue Cai and John Heidemann. Understanding Block-level Address Usage in the Visible Internet. In *Proceedings of ACM SIGCOMM 2010*, New Delhi, India, 2010.

[6] k. claffy, Y. Hyun, K. Keys, M. Fomenkov, and D. Krioukov. Internet Mapping: from Art to Science. In *IEEE DHS Cybersecurity Applications and Technologies Conference for Homeland Security (CATCH)*, pages 205–211, Waltham, MA, Mar 2009.

[7] TR-069 CPE WAN Management Protocol. https://www.broadband-forum.org/technical/download/TR-069_Amendment-5.pdf.

[8] A. Dainotti, K. Benson, A. King, k. claffy, M. Kallitsis, E. Glatz, and X. Dimitropoulos. Estimating Internet address space usage through passive measurements. *ACM CCR*, 44(1):42–49, Jan 2014.

[9] A. Dainotti, K. Benson, A. King, B. Huffaker, E. Glatz, X. Dimitropoulos, P. Richter, A. Finamore, and A. Snoeren. Lost in Space: Improving Inference of IPv4 Address Space Utilization. *IEEE Journal on Selected Areas in Communications (JSAC)*, 34(6):1862–1876, Jun 2016.

[10] Zakir Durumeric, David Adrian, Ariana Mirian, Michael Bailey, and J. Alex Halderman. A Search Engine Backed by Internet-Wide Scanning. In *Proceedings of the 22nd ACM Conference on Computer and Communications Security*, October 2015.

[11] Zakir Durumeric, James Kasten, Michael Bailey, and J. Alex Halderman. Analysis of the HTTPS Certificate Ecosystem. In *Proceedings of ACM IMC 2013*, Barcelona, Spain, 2013. ACM.

[12] Zakir Durumeric, Eric Wustrow, and J. Alex Halderman. ZMap: Fast Internet-wide Scanning and Its Security Applications. In *Proceedings of the 22Nd USENIX Conference on Security*, SEC'13, pages 605–620, Berkeley, CA, USA, 2013. USENIX Association.

[13] Xun Fan and John Heidemann. Selecting Representative IP Addresses for Internet Topology Studies. In *Proceedings of ACM IMC 2010*, Melbourne, Australia, 2010.

[14] Ramesh Govindan and Hongsuda Tangmunarunkit. Heuristics for Internet map discovery. In *Proceedings of INFOCOM 2000*, Tel Aviv, Israel, 2000.

[15] M. H. Gunes and K. Saracc. Analyzing router responsiveness to active measurement probes. In *Proceedings of PAM 2009*, 2009.

[16] John Heidemann, Yuri Pradkin, Ramesh Govindan, Christos Papadopoulos, and Joseph Bannister. Exploring Visible

Internet Hosts through Census and Survey. Technical Report ISI-TR-2007-640, USC/Information Sciences Institute, May 2007.

[17] Nadia Heninger, Zakir Durumeric, Eric Wustrow, and J. Alex Halderman. Mining Your Ps and Qs: Detection of Widespread Weak Keys in Network Devices. In *Proceedings of the 21st USENIX Conference on Security Symposium*, Security'12, Berkeley, CA, USA, 2012.

[18] B. Huffaker, M. Fomenkov, D. Moore, and k. claffy. Macroscopic analyses of the infrastructure: measurement and visualization of Internet connectivity and performance. In *PAM 2001*, Amsterdam, Netherlands, 2001.

[19] Sheharbano Khattak, David Fifield, Sadia Afroz, Mobin Javed, Srikanth Sundaresan, Vern Paxson, Steven J. Murdoch, and Damon McCoy. Do You See What I See?: Differential Treatment of Anonymous Users. In *Proceedings of NDSS 2016*, San Diego, CA, United States, 2016.

[20] Marc Kührer, Thomas Hupperich, Jonas Bushart, Christian Rossow, and Thorsten Holz. Going Wild: Large-Scale Classification of Open DNS Resolvers. In *Proceedings of ACM IMC 2015*, Tokyo, Japan, 2015.

[21] Derek Leonard and Dmitri Loguinov. Demystifying Service Discovery: Implementing an Internet-wide Scanner. In *Proceedings of ACM IMC 2010*, Melbourne, Australia, 2010.

[22] M. Luckie, Y. Hyun, and B. Huffaker. Traceroute Probe Method and Forward IP Path Inference. In *Proceedings of ACM IMC 2008*, Vouliagmeni, Greece, 2008.

[23] Antonio Nappa, Zhaoyan Xu, Juan Caballero, and Guofei Gu. CyberProbe: Towards Internet-Scale Active Detection of Malicious Servers. In *Proceedings of NDSS 2014*, San Diego, CA, USA, 2014.

[24] Ramakrishna Padmanabhan, Amogh Dhamdhere, Emile Aben, kc claffy, and Neil Spring. Reasons Dynamic Addresses Change. In *Proceedings of ACM IMC 2016*, Santa Monica, California, USA, 2016.

[25] Jeffrey Pang, James Hendricks, Aditya Akella, Roberto De Prisco, Bruce Maggs, and Srinivasan Seshan. Availability, Usage, and Deployment Characteristics of the Domain Name System. In *Proceedings of ACM IMC 2004*, Taormina, Sicily, Italy, 2004.

[26] Jean-Jacques Pansiot and Dominique Grad. On Routes and Multicast Trees in the Internet. *ACM CCR*, 28(1):41–50, January 1998.

[27] J. Postel. Internet Control Message Protocol. RFC 792, September 1981. https://tools.ietf.org/html/rfc792.

[28] J. Postel. Transmission Control Protocol. RFC 793, September 1981. https://tools.ietf.org/html/rfc793.

[29] N. Provos and P. Honeyman. ScanSSH - Scanning the Internet for SSH Servers. In *16th USENIX Systems Administration Conference (LISA)*, New York, NY, USA, 2001.

[30] Lin Quan and John Heidemann. Detecting Internet Outages with Active Probing (extended). Technical Report ISI-TR-2011-672, USC/Information Sciences Institute, May 2010.

[31] Lin Quan, John Heidemann, and Yuri Pradkin. When the Internet Sleeps: Correlating Diurnal Networks With External Factors (extended). Technical Report ISI-TR-2014-691b, USC/Information Sciences Institute, May 2014. (updated August 2014).

[32] Philipp Richter, Georgios Smaragdakis, David Plonka, and Arthur Berger. Beyond Counting: New Perspectives on the Active IPv4 Address Space. In *Proceedings of ACM IMC 2016*, Santa Monica, California, USA, 2016.

[33] Philipp Richter, Florian Wohlfart, Narseo Vallina-Rodriguez, Mark Allman, Randy Bush, Anja Feldmann, Christian Kreibich, Nicholas Weaver, and Vern Paxson. A Multi-perspective Analysis of Carrier-Grade NAT Deployment. In *Proceedings of ACM IMC 2016*, Santa Monica, California, USA, 2016.

[34] Matthew Roughan, Walter Willinger, Olaf Maennel, Debbie Perouli, and Randy Bush. 10 Lessons from 10 Years of Measuring and Modeling the Internet's Autonomous Systems. *IEEE Journal on Selected Areas in Communications*, 29(9):1810–1821, 2011.

[35] Yuval Shavitt and Eran Shir. DIMES: Let the Internet Measure Itself. *ACM CCR*, 35(5):71–74, October 2005.

[36] Neil Spring, Ratul Mahajan, and David Wetherall. Measuring ISP Topologies with Rocketfuel. In *Proceedings of ACM SIGCOMM 2002*, New York, NY, USA, 2002.

[37] P. Srisuresh, B. Ford, S. Sivakumar, and S. Guha. NAT Behavioral Requirements for ICMP. RFC 5508 (Best Current Practice), April 2009. Updated by RFC 7857.

[38] Mark Thomas, Leigh Metcalf, Jonathan M. Spring, Paul Krystosek, and Katherine Prevost. SiLK: A tool suite for unsampled network flow analysis at scale. In *IEEE BigData Congress*, pages 184–191, Anchorage, Jul 2014.

[39] Feng Wang, Zhuoqing Morley Mao, Jia Wang, Lixin Gao, and Randy Bush. A Measurement Study on the Impact of Routing Events on End-to-end Internet Path Performance. In *Proceedings of ACM SIGCOMM 2006*, Pisa, Italy, 2006.

[40] ZMap. https://github.com/zmap/zmap/.