Towards Practical Infrastructure for Decoy Routing (Positional Paper)

Sambuddho Chakravarty, Vinayak Naik, Hrishikesh B. Acharya and Chaitanya Singh Tanwar
Indraprastha Institute of Information Technology Delhi (IIITD), New Delhi, India
{sambuddho,naik,acharya,chaitanya1393}@iiitd.ac.in

***Abstract*—Network censorship and surveillance generally involves ISPs working under the orders of repressive regimes, monitoring (and sometimes filtering) users' traffic, often using powerful networking devices, *e.g.* routers capable of performing *Deep Packet Inspection* (DPI). Such routers enables their operators to observe contents of network flows (traversing their routers) having specific byte sequences. *Tor*, a low-latency anonymity network has also been widely used to circumvent censorship and surveillance. However, recent efforts have shown that all anti-censorship measures employable using Tor, *e.g. Bridges* (unadvertised relays) or camouflaging Tor traffic as unfiltered protocol messages (*e.g. SkypeMorph*), are detectable. To bypass this arms race, several recent efforts propose network based anti-censorship systems, collectively and colloquially referred to as *Decoy Routers*.**

Decoy Routing systems, relying on "friendly" network routers, aid users behind censorious ISPs to covertly access filtered networks. These Decoy Routers, otherwise operating as "normal" network routers, can on-demand double as Decoy Routers, forwarding network traffic of censored users to covert destinations. Such architectures however assume complex functionalities and programmable capabilities in commodity network routers, that currently seem infeasible. However *Software Defined Networking* (SDN), the emergent network design and management paradigm, involving centralized control over a network of switches, seems well suited for such requirements. In this position paper, we present the overview of a network based anti-censorship system consisting of several centrally co-ordinated switches, operating as Decoy Routers. Deploying centrally controlled switches, that double as Decoy Routers, could potentially have several advantages over existing proposal, that have until now only been prototyped through commodity desktops – efficiency to switch traffic at line speeds, detecting misbehaving switches, cascading multiple Decoy Routers to assume a hybrid posture for both anonymity and censorship resistance, load-balancing, and automatic failover.

I. Introduction

Online privacy and anonymity has become an oft-discussed topic in several circles. Interestingly, there has been a lot of research efforts towards designing robust anonymous communication systems [1, 2, 3]. Most of these systems trace their origins to David Chaum's seminal paper on sending untraceable emails [4]. Such systems are designed to hide the network identities (IP addresses) of either or all of the communicating peers. Such systems, derived from Chaum's paper, involve communication initiators, e.g. clients, transmitting their traffic to their communication peers, via a cascade of globally distributed, volunteer operated proxies. The traffic is also encrypted in such a way that only the communication initiator knows all the relays in the path leading to the destination, thereby ensuring anonymity against eavesdropping adversaries that might try to determine the actual source and destination of messages.

Tor [1] is a low-latency anonymous communication system designed for providing anonymity for semi-interactive services, e.g. WWW. Serving over 2 million users [5], it has acquired the de-juro status of being the most popular anonymous communication network. While originally designed only for anonymity, circumstances [6, 7, 8] forced users to exploit its distributed architecture, anonymity (of the communication peers and the proxies chosen by the users) and partial confidentiality guarantees, to bypass censorship by network operators, possibly working under the orders of repressive regimes.

Traffic to and from Tor relays can be easily filtered, since their IP addresses are publicly advertized[1]. The maintainers of the Tor project have thus suggested the use of *Bridges*, which though functionally same as regular relays, do not advertise their presence and reachability information through directory services. The information about bridges is generally communicated to users either through out-of-band methods or through *Bridge Authorities* [9]. However, encrypted bridge traffic can also be identified [10]. Thus, systems such as *Skypemorph* [11] and *Stegtorus* [12], suggest ways to camouflaging Tor messages through various cover protocol, e.g. VoIP. Sadly, such camouflaging can also be detected [13].

End-to-Middle (E2M) censorship resistance or *network based anti-censorship systems*, refers to a new paradigm of anti-censorship system design [14, 15, 16, 17], that rely on "friendly" network routers which, though most of the times operate as ordinary network routers, can stealthily aid censored users to access filtered network destinations. Which we shall conveniently refer to such systems as *Decoy Routers*[2], The connection initiators use covert signalling mechanism, often steganographic, to inform the Decoy Router to hijack their network traffic and divert it towards the censored destinations. The Decoy Router, having identified the covert signal, which also often encodes the master secret of the SSL/TLS connection, hijacks the connection and diverts it to the covert destination.

SENT '15, 8 February 2015, San Diego, CA, USA

http://dx.doi.org/10.14722/sent.2015.23011

[1] Through Tor *directory services*

[2] A term that was used to describe the first of such network based anti-censorship systems [14]

To improve the scalability of such systems, Shmatikov *et. al.* [18] suggested that such routers need to be placed in several networks in such a way that they potentially intercept large fraction of network traffic.

All such proposals have only been prototyped using commodity hardware, running desktop operating systems, forwarding a very small volume of traffic. The belief that their performance would scale proportionately in case of the wide-scale deployments remains unfounded. Moreover, all such proposals assume trusting one or possibly more network routers. Such trust models do not drastically differ from that assumed by Tor that volunteered relays. Such volunteered Decoy Routers might maliciously deviate from their expected behavior – e.g. eavesdrop on users' traffic and launch MITM attacks, misdirect traffic to malicious destinations for aiding phishing attacks, and inject large volumes of traffic to launch DDoS traffic[3]. Moreover, a user might have no information about network path characteristics and thus might not be able to select Decoy Routers to optimize performance. One possible solution to all such problems could be to assume a centralized way to control the various Decoy Routers, which could otherwise physically reside in different Autonomous Systems (ASes) and geographic regions.

Software Defined Networks (SDN) [19], the emerging network design and management paradigm, involving separation and consolidation of *control* and *data plane* functionalities seen in traditional networking devices, seems to be naturally suitable for designing, implementing, deploying and managing network based anti-censorship systems. Traditional network routers combine two separate computers into a single physical chassis. One, implementing control plane functionalities, assumes decision making capabilities such as route computations, reachability testing, access controls and traffic engineering. The other, implementing data plane functionalities, uses the inputs from control plane and primarily forwards data packets at line speeds. The controller consolidates the control plane by adding programmable functionalities, that can be used to implement complex operations like network-wide access control and policy enforcement [20, 21]. The policies are sent to the array of computationally constrained switches that actually implement the required functionalities.

In this position paper, we propose the overview of a network based anti-censorship system that relies on SDN infrastructure that may solve some of the aforementioned issues like: misbehavior detection, detection of covert signalling, cheap traffic filtering and redirections, efficient and scalable performance relying on series of L3 tunnels [22] or NAT traversals (instead of application layer proxying), automatic failover, *etc.* The system would essentially enable scaling the number of Decoy Router. Thus these Decoy Routers, though physically distributed, would remain under the control of a centralized co-ordinator. The primary tasks of the centralized controller would involve responding to covert signalling, hijacking connections to seemingly unfiltered destinations and diverting them to covert destinations, load balancing, and detecting misbehavior by maliciously acting Decoy Routers. The switches, acting as Decoy Routers would primarily operate under the control of a centralized *master* controller. The switches would communicate the covert signals to the controller, send back covert acknowledgements to the clients and aid in hijacking and redirecting the client's traffic to the covert destinations. Lastly, we also hypothesize that switches acting as Decoy Routers would forward users' traffic at line speeds, compared to commodity desktops, hitherto used to prototype such system.

As a part of some initial feasibility explorations, we tried to see it was possible to perform traffic redirection using switches. These explorations were carried out on a simulated network having a controller with few switches and hosts. In these simulations, the controller was able to successfully redirect traffic from a hypothetical client to a hypothetically unfiltered destination to an filtered one. The details of this simulation are described in Section IV. Albeit rudimentary at this stage, we plan to evaluate our ideas through complex set-ups.

II. Background Information and Related Research

The seminal work of David Chaum [4], describing a cryptographic technique which allows e-mail senders to hide their true identities from their recipients, forms the basis of Onion Routing [23] and its most popular forerunner, namely Tor. Designed for latency sensitive applications like WWW, Tor relies on a volunteer operated overlay network, comprising of over 6000 relays. Users communicate to their peers via a cascade of proxies chosen amongst these. Further, the layered encryption ensures anonymity against eavesdropping adversaries. No one, other than the connection initiator, knows the actual source *and* destination of the traffic.

Though designed for providing anonymity, people have started to use Tor for evading censorious ISPs, working under the orders of repressive regimes. However, the reliance on publicly known relays, makes Tor an easy target for traffic censorship and surveillance. Tor *Bridges* were introduced by Tor maintainers to bypass such censorship. Bridges are relays that are not publicly known, but secretly revealed to users. However, it is not impossible to detect Bridge traffic [10]. To prevent such detection, protocol obfuscation for Tor bridges has been suggested [11, 12], that involves steganographically obfuscating the user's messages to the bridges through widely used protocols such as VoIP. Sadly, all such protocol obfuscations can still be detected [13].

As mentioned earlier, E2M censorship resistance, collectively called Decoy Routers, could potentially check this arms race. Such systems propose using hard to bypass network routers that aid censorship resistance [14, 15, 16, 17]. Users covertly signal such networking elements to redirect their traffic, seemingly destined to unfiltered destinations (regarded by some researchers as *overt destination*), towards the actual covert destinations, that are filtered by the users hosting ISPs.

Decoy Routers are hard to detect because of the lack of centralized directories like Tor (users get to know about these Decoy Routers out-of-band) and covert interactions with the users. Moreover, Houmansadr *et. al.* [18] analysed how Decoy Routers could be appropriately placed such that bypassing them would incur suboptimal routing costs.

Software defined networking(SDN) [19], the emergent network design and management paradigm, involves separation

[3]In the worst scenario, several volunteer operated Decoy Routers might by compromised by adversaries that could turn it into a Botnet.

and consolidation of control and data plane functionalities. The control plane functionalities, e.g. route computation, policy enforcements, traffic engineering, monitoring *etc.*, are all consolidated into a centralized controller. Relatively computationally less intensive tasks, like traffic forwarding, is delegated to a distributed array of switches, that are centrally controlled. Figure 1 schematically describes the architecture of the SDNs, a centralized controller controlling several switches.

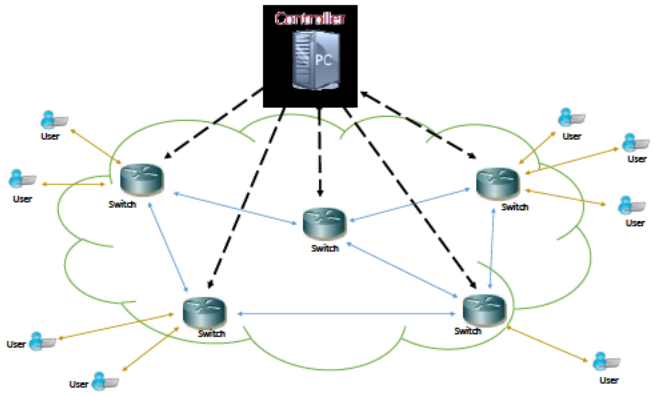

Fig. 1. A typical SDN setup within an enterprise, wherein a centralized SDN controller communicates switching rules to all the underlying switches.

In the past there have been several efforts to design network security postures, involving centrally controlled, programmable network architectures. Systems like SANE [20] have demonstrated how centrally controlled programmable network architectures could be used for network-wide security policy enforcement. In 2008, Joseph *et. al.* [24] propose a policy-aware switching layer, called *PLayer*, consisting of inter-connected policy-aware switches, called *pswitches*. These switches can be controlled centrally to explicitly forward traffic through different types of middleboxes (e.g. firewalls and SSL offloaders). Naous *et. al.* [25] proposed `ident++` a protocol that allows enforcing centrally computed security policies to end-hosts based on the latters' specific configurations, services and capabilities, that the network administrator might not have. Mehdi *et. al.* [26], demonstrated how SDN based infrastructures could be used to diagnose security breaches in SOHO set-ups. Specifically, they tried to implement various network based anomaly detection strategies.

III. Threat Model

In this paper, we assume an adversary similar in capabilities to those described by Karlin *et. al.* [14]. The client resides within networks controlled by the adversary; the adversary can, thus, eavesdrop or intercept all client traffic. However, its purpose is not to block the client, only to censor communication. For this purpose, it filters the client traffic as it passes over the network. Usually, the adversary (ISP etc.) has Deep Packet Inspection (DPI) capable routers, that look for specific byte sequences or keywords within network flows.

Further, we assume that the adversary is aware that the client is trying to circumvent censorship with Decoy Routers. It is possible that it might look for "secret handshakes" used to signal decoy routers, and run malicious Decoy Routers to deliberately redirect this traffic to some unwanted destination(or more simply, drop flows that request decoy routing services). The adversary has good monitoring capability - in fact, it is usually a nation-state - and may be able to observe network flows in various networks and perform traffic analysis to identify stealthy activity.

Against such adversaries, we suggest the use of a multi-hop Decoy Routing infrastructure. Messages could be encrypted in layers (similar to onion routing) and forwarded along a chain of Decoy Routers to reach the covert destination. Thus even traffic-analysis capable adversaries might not be able to infer the source and destination of the messages from observing traffic over only a subset of Decoy Routers along the path. Next, we discuss our proposed architecture in detail.

IV. Proposed System Architecture

This section introduces the architecture of our proposed network-based anti-censorship system. Our primary idea is that the network-based anti-censorship systems can be strengthened, with respect to usability as well as security, by using multiple Decoy Routers rather than a single one. Such a distributed architecture consisting of independently operating Decoy Routers may be both more robust and more pervasive, but immediately introduces a new problem: the multiple routers each carrying out their own changes and decisions, could potentially make mistakes or worse maliciously deviate from expected behaviour. So the natural question is, *how do we prevent errors of consensus etc. from breaking the system?*

Happily, the answer to this question has been answered by the networks research community, in the form of logically-centralized routing, i.e. Software-Defined Networks (SDN). We propose a system with several switches, potentially in different ASes and/or geographic locations, which work as Decoy Routers, under the control of a common controller. These Decoy Routers intercept the clients' traffic towards the overt destination, carrying the covert signalling message; this information is then transmitted to the controller, which responds by pushing rules to the switch that cause the client's traffic to be diverted to the covert destination.

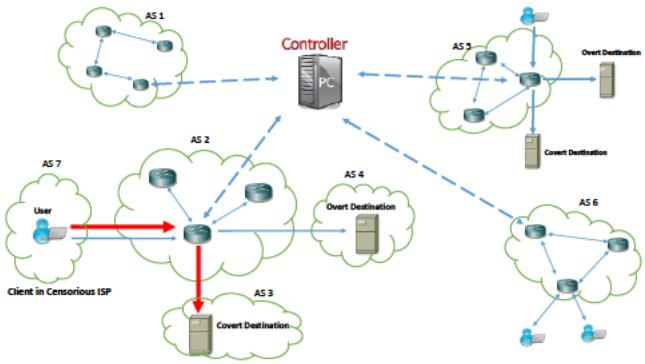

Fig. 2. Proposed network-based censorship circumvention system using SDN.

Figure 2 is a schema for our proposed architecture. The client is served by an adversary IP, which censors its communication with the covert destination. Our aim is to enable this communication to take place without the knowledge of the client's IP (or other censors, perhaps operating close to the client).

The client sends packets overtly addressed to the decoy destination. These packets bear a covert message, intended for Decoy Routers along the way, indicating that the connection should be hijacked and diverted to a covert destination. Covert messages are thus "camouflaged" in innocuous-looking overt messages, as seen in Telex [16] and Cirripede[15].

In our proposed system, the users signal the Decoy Routers for connection hijacking using chosen-ciphertext steganography, a technique demonstrated by Wustrow *et. al.* [17]. During the bootstrap process, the client communicates with the (unfiltered) decoy destination through a SSL connection that traverses a Decoy Router and covertly reveals the SSL master secret to it (which can thereafter decrypt the messages).

In Tapdance and Telex, the Decoy Router then establishes TCP/IP connections to the covert destinations on behalf of the client, and transmits the aforementioned decrypted messages. In our system, the SDN switches (Decoy Routers) need not establish connections on behalf of the client; they could simply replace the source IP address with their own, performing Network Address Translation (NAT) while sending the traffic to its appropriate covert destination (and the reverse, for reply packets from the destination). The Decoy Router could maintain the source and destination mappings to forward the acknowledgement packets back to the client.

The natural question to ask, after all this, is: *is SDN really necessary? If we wish to scale up the Decoy Router network, to use multiple routers, why not use multiple volunteer routers, just as multiple volunteer nodes are used in Tor, and use some communication between routers to keep them "in sync"?*

The answer to this question has two parts. The first one is related to security concerns of the system. Simply building a volunteer network, without oversight, could leave the system vulnerable to misbehaving Decoy Routers. (For example, such routers may eavesdrop on users' traffic, launch SSL MITM attacks, or simply divert the traffic to an incorrect destination.) A distributed, yet centrally controlled architecture could potentially solve such problems. Thus, as a part of our research we propose to design such capabilities into our SDN controller. For example, it could generate decoy traffic [27] through the switches, or use certificate and public key pinning [28] to detect possible MITM attacks.

The second reason is to do with performance and features. Any distributed architecture will provide redundancy, but in case of an organized system (SDN), it can also achieve automatic failover and load balancing. The SDN controller can periodically probe the Decoy Router, similar to Tor Bandwidth Authorities, to determine the availability, reachability, and capacity, and could covertly inform a requesting client to choose a Decoy Router that may potentially provide high bandwidth to the client.

Further, the use of SDN switches ensures that they have a common platform, with common powers. In Tapdance, a copy of the client's messages continue to reach the overt destination. These messages are however discarded by their recipients due to their unexpected TCP sequence numbers. The authors posit that inline blocking of traffic to the original overt destination (as assumed by older Decoy Routing strategies) was not feasible with existing hardware. It is our observation that SDN infrastructure is highly programmable, and it should be possible to use SDN features to implement inline blocking.

Finally, we also hypothesize that using programmable switches, capable of performing complex operations, like decrypting traffic and redirecting them to covert destinations, would offer scalable performance, compared to the current prototypes that have been merely been tested on commodity desktops offering no real-time performance upper-bounds.

To summarize, the main design goals of our system are as follows.

- **Centralized Decoy Router Management** The centralized controller adds several features to the distributed Decoy Routing architecture: covert signal detection, available bandwidth monitoring, and automatic failover detection. For example, it maybe possible to covertly signal the client to choose a different decoy destination, if no high-bandwidth Decoy Routers are available along a certain route.
- **Misuse/Misbehavior Detection** As the system essentially relies on SDN, it can support known techniques by the controller to detect misbehaving switches. This essentially empowers the controller to also act as a monitor, and prevent switches from (for example) launching MITM attacks on users traffic.
- **Scalability** The system should potentially support several decoy routers. This not only reduces the vulnerability (to discovery) of a single router, but also provides the user with options: it is possible to connect to a wide variety of overt destinations, knowing that on each of these paths there is at least one decoy router. Also, the total amount of traffic that can be served in such an anti-censor router network is greater than with a single anti-censorship router.
- **Efficient Traffic Diversion** The centralized controller could effectively divert client to overt destination client to the intended covert destination, rather than simply allowing the packets also reach the overt destination, as in TapDance. Besides the fact that SDN switches can have rich functionality (inline blocking), they may also be able to divert the connection to covert destinations by replacing the source IP address with one of their choosing (NAT).
- **Key Negotiation for Multi-hop Decoy Routing** An adversary, observing traffic flowing through a decoy router, could launch traffic analysis attack to identify covert connections. To hide against such adversaries, the controller could aid the client to *cascade* Decoy Routers, transforming the message sent over each hop between decoys (similar to Onion routing).
  To accomplish this, we extend the use of steganographic signaling to perform covert communication, not only between the client and the routers, but between the client and the controller. The client's hidden messages are picked up by the SDN switch and passed on to the controller; now a key exchange can be performed between the client and the controller, etc. Along with deciding the path through the Decoy Routers, the controller pushes *match-action* rules to

the effect of "decrypt message with this key". The client generates messages encrypted in layers, to be decrypted hop-by-hop by two or more Decoy Routers; the final Decoy Router, which decrypts the last layer of encryption, transmits the decrypted traffic to the covert. This process would essentially mimic Onion Routing, and may serve to prevent traffic analysis attacks by eavesdropping adversaries.

A. *Initial Feasibility Study of Traffic Redirection*

In order to study the feasibility of realizing traffic redirection in the proposed architecture, we simulated a network having a controller, three switches, and two hosts. The simulation was run on Mininet, using `OpenDaylight` [29] as the controller and `OVS-switches` [30]. Figure 3 shows the network structure; it is noteworthy that each switch, as well as the controller, could be in different ASes.

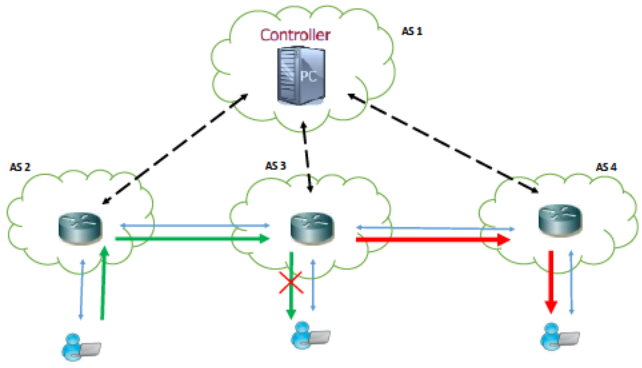

Fig. 3. Network topology that was simulated using Mininet: Traffic from AS2, originally destined to AS3 was redirected to AS4.

As is standard in SDN, the controller populates the forwarding table of each of the switches with match-action rules. In this simulation, we pushed a rule which diverted all TCP/IP requests from AS2 to AS3 to go to AS4 instead. (In the simulation, each host is in its own individual AS.)

The objective of the simulation was to demonstrate that the network agnostic presence of our SDN could be used to implement and run Decoy Routers. The simulation was successful; we were able to redirect traffic on the basis of the sender's IP address. Though only an initial feasibility study at this point, in future, a full-fledged research paper would involve evaluating our ideas through complex topologies.

V. Discussions and Limitations

Our research goal is to practically demonstrate how SDN based infrastructure could be used to design efficient network based anti-censorship systems that have hitherto been merely prototyped using commodity desktop hardware, running non-dedicated operating systems. SDN based infrastructures, consisting of centrally controlled networking elements with programmable capabilities are potentially useful substrates to build privacy preserving and network-based censorship resistance systems. Having presented an overview of the proposed system, we now describe some of the security and privacy concerns potentially pertinent to such systems.

We plan to design our system similar to Tapdance [17], a recently proposed network-based anti-censorship system. Though our immediate future goals involve designing a distributed network-based anti-censorship system with several Decoy Routers, in the long run, SDN infrastructures could also be used to design network based anonymity preserving system, similar to Tor. Such systems could rely on a cascade of Decoy Routers to forward the clients traffic to the covert destination. During the bootstrap process, the client could covertly signal one or more Decoy Routers to request Decoy Routing services. Additionally, the covert signal could also encode information to aid key exchange with multiple Decoy Routers. A covert response, aided by the centralized controller could acknowledge the request along with information to complete the key exchange. The key exchange protocol could specifically encode information that is only relevant for use with specific Decoy Routers. The client could thereafter encrypt the messages, intended for the covert destination, using some kind of layered encryption (like Onion Routing) with the keys established from the previous step. The encrypted message could be tunnelled via the SSL connection to the overt destination. The appropriate Decoy Router en-route intercept the packets to the overt destination, decrypts it and realizes the next Decoy Router to forward the message. The second Decoy Router decrypts yet another layer of encryption and might forward it to a third one, depending upon the layers of encryption used, or sends them to the intended covert destination. Such measures could confound an adversary that can launch traffic analysis attack for identifying the source and destination of the messages.

The controller could have operations similar to a hidden *Bandwidth Authority* [31], that, apart from periodically scanning the available capacity of the various Decoy Routers, could monitor them to identify sudden surges of traffic and detect malicious activities like MITM attacks. As mentioned previously, the controller could inject decoy SSL traffic to detect such activities [27].

Unlike Onion Routing based systems, each Decoy Router on the path could forward the encrypted message to the next one along the path or to the covert destination, by merely changing the source IP address of the message, without transporting (proxying) it through another TCP/IP connection. Onion Routing systems, like Tor, often incur heavy performance penalties because their own traffic scheduling policies interfere with underlying TCP congestion control mechanism [32]. Such slow-down could potentially be prevented by wholly avoiding application layer proxying. As mentioned previously, the switches (Decoy Routers) could achieve this by replacing the source IP address of the messages with their own (NAT). This may essentially have the same affect of hiding the true source IP address, as is achieved through proxies.

A single point of control could aid centralized failure detection and recovery. However, it could also be make it easier for an adversary to compromise and assume control of all the Decoy Routers. Moreover, SDN architectures involving a centralized controller that monitors several switches, could potentially be abused by attackers to design Botnets, wherein the controller (acting as the Botmaster) could, for example, force switches to generate traffic to launch link-flooding DDoS attacks. One way to avoid such attacks may involve using the

Decoy Routing switches itself for routing the traffic between the controller and the switches.

VI. Conclusions

In this position paper, we propose to explore, the efficacies of using centralized network control architectures (*e.g.* SDN) to design network-based anti-censorship systems. Intuitively, such centralized, programmable network infrastructures seem ideal for designing network-supported censorship resistance systems, collective and colloquially referred to as Decoy Routing. Such systems are designed to aid users residing in censorious ISPs, that censor (or merely surveil) users' traffic. Their designs involve "ordinary" network routers that double as device capable of covertly diverting certain users' traffic to filtered network destination. Hence, they are generally considered more resilient to network censorship, compared to older architectures involving proxies and overlays (*e.g.* Tor). Designed primarily to achieve network anonymity, Tor traffic is easy to monitor or censor, since the anonymization relays are publicly announced.

Our proposed distributed architecture using SDN, would involve multiple switches, in different ASes, that would function as Decoy Routers. These Decoy Routers would be controlled by a centralized controller. The controller's main tasks would involve: observing the network traffic to identify covert signaling by clients who seek Decoy Routing services, decryption and appropriate traffic redirection, identifying maliciously behaving switches, load balancing, automatic failover *etc.*. Though we plan to model our system like Tapdance, a recently proposed Decoy Routing system, we believe that centralized, programmable architectures like SDNs are ideal substrates to build, design, deploy and manage various kinds of complex network-based censorship-circumvention architectures, that might potentially feature a cascade of Decoy Routers (instead of the usual single one) en-route to the covert destination.

Decoy Routing proposals have till now been prototyped on commodity hardware running non-dedicated operating systems. Such systems often rely on application layer forwarding (proxying) to redirect the traffic to the covert destinations. Thus often the congestion control mechanisms of underlying connection, that is being proxied, interferes with the connection scheduling at the proxy itself. This tends to degrading end-to-end performance for the users. It maybe difficult to expect line speed performance guarantees when such implementations are deployed to serve large volume of traffic. Therefore, we also believe that using high speed switches, acting as Decoy Routers redirecting traffic to covert destination using NAT (or lower layer tunnelling), could potentially provide usable quality of service guarantees.

References

[1] R. Dingledine, N. Mathewson, and P. Syverson, "Tor: The Second-Generation Onion Router," in *Proceedings of the 13th USENIX Security Symposium)*, August 2004, pp. 303–319.

[2] T. Isdals, M. Piatek, A. Krishnamurthy, and T. Anderson, "Privacy-preserving P2P data sharing with oneswarm," in *Proceedings of the Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications (SIGCOMM)*, 2010, pp. 111–122.

[3] H.-C. Hsiao, T. H.-J. Kim, A. Perrig, A. Yamada, S. Nelson, M. Gruteser, and W. Ming, "LAP: Lightweight anonymity and privacy," in *Proceedings of the 2012 IEEE Symposium on Security and Privacy*, May 2012.

[4] D. L. Chaum, "Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms," *Communications of the ACM*, vol. 24, no. 2, pp. 84–90, February 1981.

[5] "Tor Metrics Portal," http://metrics.torproject.org/.

[6] "Internet censorship in syria," http://www.thenational.ae/news/world/middle-east/syria-tightens-control-over-internet.

[7] "Internet censorship in the people's republic of china," https://en.wikipedia.org/wiki/Internet_censorship_in_the_People%27s_Republic_of_China.

[8] "Internet censorship in iran," https://http://iranmediaresearch.org/en/research/pdffile/1296.

[9] "Tor bridges."

[10] P. Winter and S. Lindskog, "How the Great Firewall of China is blocking Tor," in *Proceedings of the USENIX Workshop on Free and Open Communications on the Internet (FOCI 2012)*, August 2012.

[11] H. M. Moghaddam, B. Li, M. Derakhshani, and I. Goldberg, "Skypemorph: Protocol obfuscation for Tor bridges," in *Proceedings of the 19th ACM conference on Computer and Communications Security (CCS 2012)*, October 2012.

[12] Z. Weinberg, J. Wang, V. Yegneswaran, L. Briesemeister, S. Cheung, F. Wang, and D. Boneh, "StegoTorus: A camouflage proxy for the Tor anonymity system," in *Proceedings of the 19th ACM conference on Computer and Communications Security (CCS 2012)*, October 2012.

[13] A. Houmansadr, C. Brubaker, and V. Shmatikov, "The parrot is dead: Observing unobservable network communications," in *Proceedings of the 2013 IEEE Symposium on Security and Privacy*, May 2013.

[14] J. Karlin, D. Ellard, A. W. Jackson, C. E. Jones, G. Lauer, D. P. Mankins, and W. T. Strayer, "Decoy routing: Toward unblockable internet communication," in *FOCI'11 - Proceedings of the USENIX Workshop on Free and Open Communications on the Internet*, San Francisco, CA, USA, August 2011.

[15] A. Houmansadr, G. T. K. Nguyen, M. Caesar, and N. Borisov, "Cirripede: Circumvention infrastructure using router redirection with plausible deniability," in *Proceedings of the 18th ACM conference on Computer and Communications Security (CCS 2011)*, October 2011.

[16] E. Wustrow, S. Wolchok, I. Goldberg, and J. A. Halderman, "Telex: Anticensorship in the network infrastructure," in *Proceedings of the 20th USENIX Security Symposium*, August 2011.

[17] E. Wustrow, C. M. Swanson, and J. A. Halderman, "Tapdance: End-to-middle anticensorship without flow blocking," in *Proceedings of 23rd USENIX Security Symposium (USENIX Security 14)*. San Diego, CA: USENIX Association, August 2014.

[18] A. Houmansadr, E. L. Wong, and V. Shmatikov, "No direction home: The true cost of routing around decoys," in *Proceedings of the Network and Distributed Security Symposium - NDSS '14*. Internet Society, February 2014.

[19] "Software defined networking," http://en.wikipedia.org/wiki/Software-defined_networking.

[20] M. Casado, T. Garfinkel, A. Akella, M. J. Freedman, D. Boneh, N. McKeown, and S. Shenker, "Sane: A protection architecture for enterprise networks," in *Proceedings of the 15th Conference on USENIX Security Symposium - Volume 15*, ser. USENIX-SS'06. Berkeley, CA, USA: USENIX Association, 2006. [Online]. Available: http://dl.acm.org/citation.cfm?id=1267336.1267346

[21] M. Casado, M. J. Freedman, J. Pettit, J. Luo, N. McKeown, and S. Shenker, "Ethane: Taking control of the enterprise," *SIGCOMM Comput. Commun. Rev.*, vol. 37, no. 4, pp. 1–12, Aug. 2007. [Online]. Available: http://doi.acm.org/10.1145/1282427.1282382

[22] G. Hampel, M. Steiner, and T. Bu, "Applying software-defined networking to the telecom domain," in *Computer Communications Workshops (INFOCOM WKSHPS), 2013 IEEE Conference on*, April 2013, pp. 133–138.

[23] P. F. Syverson, D. M. Goldschlag, and M. G. Reed, "Anonymous Connections and Onion Routing," in *IEEE Symposium on Security and Privacy*, May 1997, pp. 44–54.

[24] D. A. Joseph, A. Tavakoli, and I. Stoica, "A policy-aware switching layer for data centers," in *Proceedings of the ACM SIGCOMM 2008 Conference on Data Communication*, ser. SIGCOMM '08. New York, NY, USA: ACM, 2008, pp. 51–62. [Online]. Available: http://doi.acm.org/10.1145/1402958.1402966

[25] J. Naous, R. Stutsman, D. Mazieres, N. McKeown, and N. Zeldovich, "Delegating network security with more information," in *Proceedings of the 1st ACM Workshop on Research on Enterprise Networking*, ser. WREN '09. New York, NY, USA: ACM, 2009, pp. 19–26. [Online]. Available: http://doi.acm.org/10.1145/1592681.1592685

[26] S. A. Mehdi, J. Khalid, and S. A. Khayam, "Revisiting traffic anomaly detection using software defined networking," in *Recent Advances in Intrusion Detection*, ser. Lecture Notes in Computer Science, R. Sommer, D. Balzarotti, and G. Maier, Eds., vol. 6961. Springer Berlin Heidelberg, 2011, pp. 161–180.

[27] P. Winter and S. Lindskog, "Spoiled Onions: Exposing Malicious Tor Exit Relays," Karlstad University, Tech. Rep., 2014. [Online]. Available: http://veri.nymity.ch/spoiled_onions/techreport.pdf

[28] "Certificate and public key pinning," https://www.owasp.org/index.php/Certificate_and_Public_Key_Pinning.

[29] "Open daylight," http://www.opendaylight.org/.

[30] "Open vswitch," http://openvswitch.org/.

[31] K. Loesing, M. Perry, and A. Gibson, "Bandwidth scanner specification," https://gitweb.torproject.org/torflow.git/blob/HEAD:/NetworkScanners/BwAuthority/README.spec.txt.

[32] J. Reardon and I. Goldberg, "Improving tor using a tcp-over-dtls tunnel," in *Proceedings of* 18^{th} *USENIX Security Symposium 2009*, August 2009.