Poster: Large-Scale Tor Experimentation

Sukhbir Singh, Ian Goldberg, David Taylor
Cheriton School of Computer Science
{s3singh,iang,dtaylor}@uwaterloo.ca
University of Waterloo,
Waterloo, Ontario N2L 3G1, Canada

***Abstract*—Tor is the most popular low-latency anonymous communication system for the Internet, helping people to protect their privacy online and circumvent Internet censorship. Its distributed design and low-latency based anonymity has attracted the attention of network and security researchers alike who work on evaluating the performance and security of the network. Evaluating changes to the design of the protocol or studying attacks against it in the live network is undesirable as it can invade the privacy of users and even put them in harm's way. Traditional Tor research has been limited to testing a few hundred nodes with the Modelnet network emulator, which may not accurately represent the real-world Tor network. We present SNEAC (Scalable Network Emulator for Anonymous Communication), a large-scale network emulator that allows us to emulate a network with thousands of nodes. Our hope is that with such large-scale experimentation, we can more closely emulate the live Tor network with half a million users.**

I. Introduction

The Internet provides a platform for various forms of communication and activism that helps the free flow of information and ideas spanning geographic boundaries. To control the freedom of their citizens in the virtual world, many regimes started censoring the Internet by restricting access to or publication of certain content that they deemed inappropriate, while others started spying on the online activities of their citizens. Fortunately for the people, attempts to restrict their freedom on the Internet were not fruitful — many systems were developed which help people maintain their anonymity on the Internet and circumvent censorship.

The most effective and widely used such system is called Tor, which is based on a technique called onion routing [1] designed at the U.S. Naval Research Laboratory in 1998. In onion routing, messages are repeatedly encrypted and sent across the network through several nodes called *onion routers*. Analogous to removing the layers of an onion, each router strips off a layer of encryption and passes the message to the next router; this process is repeated until the message reaches its final destination. Tor [2] is the most prominent implementation of onion routing and has been in development since 2002 with about half a million people using it every day to protect their privacy on the Internet and to resist state-sponsored Internet censorship.

II. Traditional Tor Research

As Tor is under active development, the research community frequently proposes design changes to the protocol, which may focus on improving the performance of the network or implementing defenses for possible attacks against it. Testing these proposals in the live network itself is undesirable as doing so can invade the privacy of users, and in some cases, even put them in harm's way. As a workaround, a local offline Tor network can be set up and tested using the ExperimenTor [3] network emulator based on Modelnet [4], or using Shadow [5], a discrete event simulator.

Modelnet is a network emulator that allows distributed networks to be evaluated locally. A typical Modelnet setup consists of the emulator machine (the core) which sets up the topology and emulates network characteristics, such as latency, bandwidth, jitter and packet loss. There are multiple machines connected to the emulator called *edge nodes* that run unmodified TCP-based applications (which in our case is Tor). When a process on an edge node wants to communicate with another process on the same or a different edge node, it goes through the emulator and experiences the characteristics described above, thus emulating a packet as it travels across a real world network.

Traditional Tor experimentation with ExperimenTor has been limited to a few hundred nodes due to the limitations of the underlying Modelnet emulator that it runs on. Modelnet runs (only) on a patched FreeBSD 6.3 kernel [6], which has limitations on the amount of resources it can access, such as the physical memory it can address; the number of CPU cores it supports; and the maximum bandwidth it can push through the NIC. Modelnet is no longer maintained (the last release was in 2005) and because of this, it also has some other issues as well, such as inducing kernel panics. Because of all these issues, it is not possible to run an experiment with thousands of nodes on Modelnet.

To alleviate these problems, we present SNEAC, a network emulator that is easy to set up and scales to thousands of nodes. We show that SNEAC is an effective replacement for Modelnet by comparing the two systems' performance. We also discusss SNEAC's scalability.

III. Our Contribution

In SNEAC, we set up a network using Mininet [7], an emulator that allows rapid prototyping of software-defined networks. Mininet creates a virtual network that runs hosts (Linux containers) and switches (Open vSwitch) on a single machine. Open vSwitch [8], an implementation of the OpenFlow communications protocol [9] is used for setting up the flow rules and handling the flow of packets between switches.

In a typical Mininet setup, both hosts and switches run on a single machine. In SNEAC, we modify this architecture to reflect that of Modelnet — a single machine (the emulator)

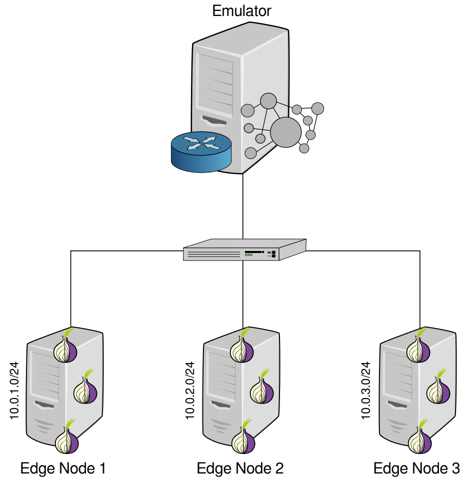

Fig. 1. SNEAC setup with the emulator and three edge nodes running Tor

runs only switches while the multiple edge nodes connected to it run the processes running on the emulated network (see Figure 1). Using Mininet, we load a GraphML network topology file that sets up the switches and creates the links with network performance parameters. On the edge nodes, processes are bound to virtual interfaces and the emulator is set as their default gateway. On the emulator, static routing is configured between switches, such that packets from the edge nodes traverse through the emulator and then exit to one of the edge nodes, thus emulating an Internet-like environment.

Our emulator is open source, easy to configure (single-click setup), and runs on all major Linux distributions without requiring kernel modification.

IV. Evaluation

To show that SNEAC is an effective replacement for Modelnet, we ran a network topology with 36 switches and 80 links on both the emulators. In each case, a single edge node running a small Tor network with three directory authorities, four relays and seven clients was connected to the emulator. HTTP clients on the edge node were connected to the Tor clients to fetch files over the local network from a HTTP server. Figure 2 shows the total number of seconds it take for seven clients running concurrently to download a 1MiB file on a topology with a latency of 10ms, bandwidth of 1 Mbps and a queue length of 10. We note that the performance of Modelnet and SNEAC is statistically equivalent.

While not discussed here, we also evaluated SNEAC's scalability where we were able to run a topology with 183 switches and 16000 links, which is impossible to emulate on a Modelnet setup. Due to concerns with the scalability of OVS, such as the number of switches it supports and the startup time [10], we are currently working on replacing the switches with LXC hosts that act as packet forwarders.

In Modelnet, there is a limit on the number of Tor clients that can be run, which is equal to the number of switches in the topology. There is no such restriction in SNEAC and it is possible to run a network with a few hundred switches but hundreds of thousands of clients connected to it.

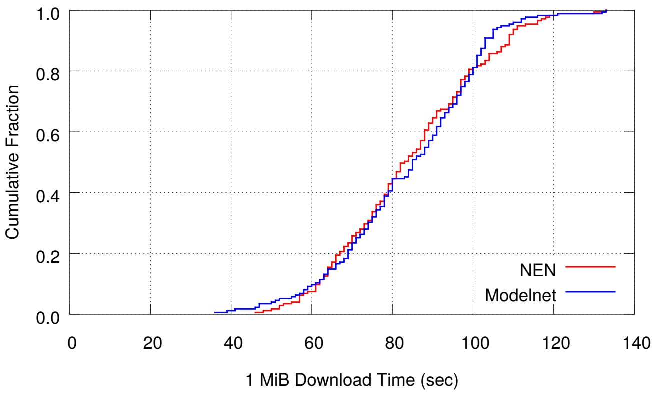

Fig. 2. Download times for 1 MiB file on SNEAC and Modelnet

V. Conclusion

We presented SNEAC, a network emulator that is easy to configure and can scale to a topology with thousands of switches and hundreds of thousands of clients running on multiple edge nodes. We also showed that it is an effective replacement for existing emulator testbeds like Modelnet. Our hope is that SNEAC will allow network and security researchers to effectively model a real-world Tor network with half a million users. Though our focus is on emulating the Tor network, we also plan to make SNEAC available as a general-purpose large-scale network emulator.

References

[1] M. G. Reed, P. F. Syverson, and D. M. Goldschlag, "Anonymous connections and onion routing," *Selected Areas in Communications, IEEE Journal on*, vol. 16, no. 4, pp. 482–494, 1998.

[2] R. Dingledine, N. Mathewson, and P. Syverson, "Tor: The second-generation onion router," in *Proceedings of the 13th Conference on USENIX Security Symposium - Volume 13*. USENIX Association, 2004.

[3] K. Bauer, M. Sherr, D. McCoy, and D. Grunwald, "Experimentor: A testbed for safe and realistic tor experimentation," in *Proceedings of the USENIX Workshop on Cyber Security Experimentation and Test (CSET 2011)*, August 2011.

[4] K. Yocum, K. Walsh, A. Vahdat, P. Mahadevan, D. Kostic, J. Chase, and D. Becker, "Scalability and accuracy in a large-scale network emulator," *ACM SIGCOMM Computer Communication Review*, vol. 32, no. 3, 2002.

[5] R. Jansen and N. Hopper, "Shadow: Running Tor in a Box for Accurate and Efficient Experimentation," in *Proceedings of the Network and Distributed System Security Symposium - NDSS'12*. Internet Society, February 2012.

[6] Modelnet. [Online]. Available: http://modelnet.ucsd.edu/

[7] B. Lantz, B. Heller, and N. McKeown, "A network in a laptop: rapid prototyping for software-defined networks," in *Proceedings of the 9th ACM SIGCOMM Workshop on Hot Topics in Networks*. ACM, 2010, p. 19.

[8] B. Pfaff, J. Pettit, K. Amidon, M. Casado, T. Koponen, and S. Shenker, "Extending networking into the virtualization layer." in *Hotnets*, 2009.

[9] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, "Openflow: enabling innovation in campus networks," *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 2, pp. 69–74, 2008.

[10] OVS ovs-vswitchd limitations. [Online]. Available: http://openvswitch.org/cgi-bin/ovsman.cgi?page=vswitchd%2Fovs-vswitchd.8