Building Block for the Internet Digital Twin: Scalable and Automated Internetwork Emulator

Joshua Levett, Vassilios Vassilakis, Poonam Yadav
Department of Computer Science, University of York

Abstract

Even minor changes to Internet routing protocols or their configurations can result in severe disruptions to the stability and security of global Internet routing. However, current test environments remain limited in scope and fail to reflect the true complexity and scale of the Internet. This highlights the urgent need for a high-fidelity Internet Digital Twin - a large-scale, emulated environment that can safely replicate and experiment with real-world Internet conditions. In this extended abstract, we present our approach to building such an Internet-scale digital twin for internetwork routing protocols. By capturing the current state of Internet topology, we generate a virtual testbed using a container orchestration framework, with each container simulating an ISP-level network. This infrastructure enables rigorous testing and enhancement of Internet routing protocols under realistic conditions. Looking ahead, our vision is to scale these emulations further to encompass the entire observable Internet, establishing the digital twin as a critical tool for future Internet resilience and innovation.

I. Introduction

Comprising of over 86,000 Autonomous Systems (ASes), each making policy-based routing decisions, the Internet is formed of largely independent networks each utilising one or more border routers, each connecting their internal network to others across the globe. It is this heterogeneous nature - wherein each network operator can optimise their networks to their needs - which characterises the Internet. It can also be an inherent limitation, such as when individual ASes make small policy changes that have unintended consequences for other networks. It is particularly an issue wherein deliberate border router misconfiguration allows for the hijacking or non-routability of traffic. In this work we take a step toward our vision of creating a digital twin, a virtual replica of the underlying infrastructure of the 'physical' Internet, which if constructed accurately can provide insight into the potential consequences of making policy and configuration changes on the wider Internet [1].

For many years there has been a concerted effort to mitigate the impact of these misconfigurations. The de-facto Internet routing protocol, BGP, faced its last major revision in 1994 [2], and there have been several attempts to introduce protocol extensions, such as S-BGP [3] and more recently, BGPsec [4]. There have also been other attempts to replace the protocol altogether to place a greater emphasis on security and complete route verification, such as the SCION suite of protocols [5]. None of these have received as much traction, however, as RPKI [6], a system for connecting Internet resources (AS numbers and IP address ranges) to a trust anchor, requiring resource owners to record the ASes from which they are advertising IP routes, and for each AS to verify that these pairings match when receiving routes in BGP. It remains the case that the majority of ASes have not made this effort [7], thus limiting the benefit of adoption. Limited testing environments for routing protocols lead to unpredictable behaviour across vendors [8] and allow minor bugs to cause widespread routing failures [9], undermining Internet reliability.

The use of physical testbeds to replicate or experiment with large topologies is costly and can be architecturally limiting, meaning they are infeasible for replicating topologies at substantive scale. In the case of SCIONLab [10], the scale is restricted to an extent that it cannot reasonably be considered representative of the Internet. The largest physical testbed available to researchers is PEERING [11], a partially virtualised platform which provides full-scale BGP research but is limited by its need to enforce non-breaking changes resulting from its direct connection to the wider Internet.

An alternative is to mimic such environments as much as possible through simulation or emulation, substantially lowering the build costs and providing opportunities to scale or make configuration changes with comparatively little effort. There has been some work undertaken to develop simulations loosely based on topology data [12], however the limitation of simulation (which by not using 'real-world' software but instead abstractions of it) means the visibility of interoperability changes – either in differing vendor software implementations, or the heterogenous configurations of the Internet – is limited. Previous work on the emulation of Internet routers and topologies includes works such as [13] or [14], which focus specifically on the replication of BGP and its extensions within subsets of the topology, rather than the existing Internet.

In this extended abstract, we explore the potential for container virtualisation in enabling large-scale emulations of Internet routers, allowing vendors to evaluate the interoperability of their routing products and operators to validate their router policy configurations and resolve external routing issues introduced by these changes. By replicating subsets of existing Internet topology, we demonstrate that containerisation is a feasible approach to Internet border router emulation.

II. Related Work

There exist a small number of other more detail-oriented emulation-based approaches, including SONiC, a framework designed for abstracting network software from the underlying physical platform, which provides some virtualisation capabilities [15]. Similarly, other works have focused on the emulated twinning of large production networks [16], [17]. None, however, have focused on the challenge of emulation at even larger scale, with networks consisting of

tens of thousands of heterogenous routers implementing unique BGP configurations and policies. Indeed, the closest work [13] utilised a similar graph-driven approach but assumed requirements of 37 GB RAM to replicate only 1158 routers and 1470 links, a resource requirement infeasible for Internet-scale emulations. Our contribution remains novel in emulating a more complete Internet snapshot based directly on more complete views of Internet topology, with an automated pipeline developing a substantively larger scale Internet emulation.

III. Methodology & Implementation

Our approach (as shown in Figure 1) captures the state of Internet topology and undertakes an automated process to generate a container-based virtual testbed to emulate the captured topology. We deliberately design the underlying tooling to allow for users to input their own graph topologies which can allow for the community to replicate Internet topology in more depth or complexity (for instance, with replication of some internal network structure). This also allows for the input of richer graphs produced in our previous work [18].

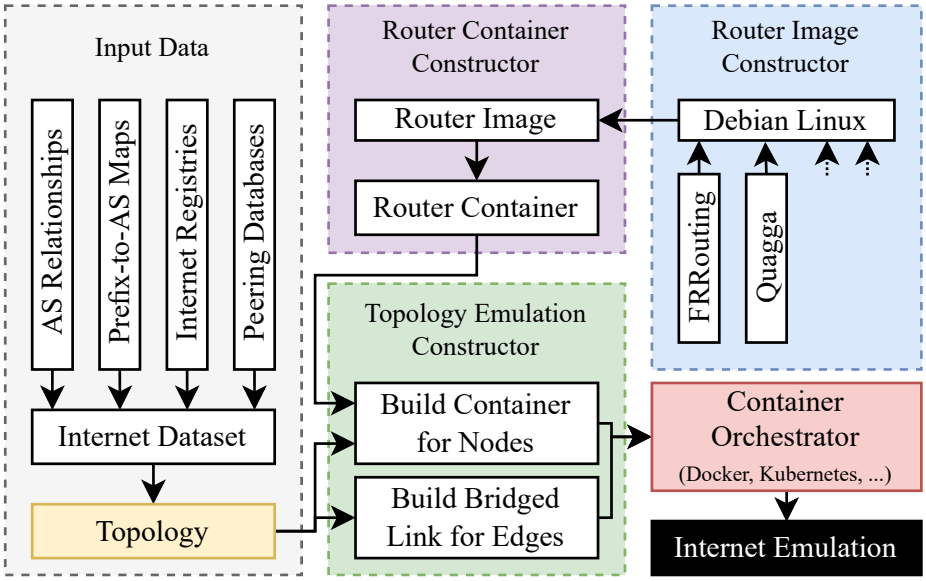

Fig. 1. Overview of our methodology. We capture a snapshot of the Internet and produce a topology by fusing a variety of sources. We extend this, automating the creation of container images for each router in the topology and their associated routing configuration files. We further automate the configuration of inter-container networking using bridged links. The complete emulation is then deployed using a container orchestrator.

This work focuses on implementing each Internet AS as an individual router. Not necessarily representative of all Internet routing dynamics, this provides significant scale and variety of routing configurations whilst helping to keep emulations from becoming unduly complex. We generate a graph-based Internet topology representation using data from a variety of sources for a fixed point in time. This aspect of work is not particularly novel and is inspired by previous works by a variety of authors, including [19], [20].

We create a Python library underpinned by `networkx` [21], introducing a variety of additional attributes for graph nodes (ASes), and enabling specification of router image and dynamics relevant to individual router configurations. Similarly, for each graph edge we supplement the source and destination attributes with those relevant to the specification of individual routers in addition to link characteristics.

We allow for the automated generation of these topology graphs by implementing loaders for CAIDA's *AS Relationships* [22] dataset to collect a snapshot of inferred Internet topology structure for a specified day, wherein for each catalogued relationship we create a labelled 'edge' between two AS nodes in our graph-based representation. We additionally allow the input of data from PeeringDB snapshots [23], which provide self-reported and community generated information about ASes, such as detail about the network type and presence at Internet Exchanges, including information about IXP-assigned router IP addresses. This data is extracted into additional attributes available on each AS, and can further be used to generate more complex routing configurations based on IXP presence or network type (for instance, transit networks will carry traffic bound for other ASes, whilst content delivery networks will likely provide little transit but remain connected to a substantive number of other ASes). Further, we allow for the loading of IP prefixes advertised by each AS by reversing the CAIDA *Prefix-to-AS* dataset [24] or the BGP table as seen by *bgp.tools* [25]. For each AS within the topology, we collate a list of prefixes which we assume it will advertise to its neighbours. We separate this aspect of the automation from the remaining aspects of emulation, enabling others to use a customised `.graphml` format file to customise emulations.

Router Emulation. For each router software we construct a router container image based on the *Quagga* routing suite [26] and its Linux Foundation fork *FRRouting* [27]. For each router we instantiate a list of network interfaces between direct routing neighbours based on the captured real-world Internet topology and then create and populate the necessary configuration files for the chosen software suite. In this work, the diversity of configuration varies only by topological information. The primary advantage of our container-based emulation approach is its scalability. By reducing resource duplication, such as a router's underlying image and filesystem, and replicating only necessary elements (router memory, routing configuration and routing policies), our approach reduces the resource constraints required for large-scale emulation.

Generating Router Configurations. Irrespective of the specific routing suite implementation, there are a number of shared aspects to any BGP router configuration. Each of the router containers requires a network interface through which to be able to speak to its direct neighbours, thereby creating a virtual LAN. In 'real' instances, this would be by connecting two networks physically, such as at an Internet Exchange Point (IXP). In the emulated environment, there is an added challenge in that it is imperative for later policy configuration (where each BGP neighbour needs a static IP address) that we assign addresses unique for each interface and that do not clash with the addresses used by direct neighbours. We therefore generate a unique identifier for each graph edge which is used as an index for the network interface and to inform the generation of the subnets assigned to each such interface. We use the non-global 'future use' `240.0.0.0/4` IPv4 and 'undefined' `fc00::/8` IPv6 address spaces for this, divided between the network interface derived address and local

host identifier bits, both of which should not feature in the global routing table. We leave typical internal address ranges available for future extensibility – such as instances where multiple routers may serve a single AS.

Inter-Container Networking. The implementation of inter-container networking differs slightly between container orchestration frameworks. We generate emulation configurations for *Megalos* [28], a Kubernetes-based container orchestration framework that implements eBGP routing across multiple nodes and the deployment of virtual LANs, each representing a container-to-container connection. This approach enables use of a host cluster, thus providing a foundation for a large heterogeneous emulation without some of the resource restrictions imposed by a single-host Docker deployment.

For each node within the topology graph we produce a single Kubernetes pod. These are natively assigned an 'IP-per-pod' address accessible through the onboard `eth0` interface, which we then disable in the router configurations. Each topology edge becomes a distinct virtual LAN, implemented as a Megalos VXLAN segment, connecting two or more containers. Each VXLAN segment is identified using a VXLAN network identifier, a four-character code comprised of numbers and letters (base-36) and providing a theoretical maximum far in excess of the true number of Internet topology adjacencies. This implementation produces a topology wherein there are as many containers as nodes and vLANs as edges. For each pod, this means introducing virtual network interfaces with a local IP address assigned for each participant of every LAN which is disjoint from the IP addresses used by a router or its neighbours, and simultaneously not globally addressable.

Container Orchestration. Finally, we deploy our Internet emulation scenarios, creating containers for each node of the topology and a bridged link for each edge. This is managed by the chosen container orchestration framework, using underlying router images and implemented inter-container networking interfaces. When deployed, it is possible to interact with router instances through the container orchestrator, with login details for each node predefined in the applicable router images.

IV. Preliminary Results

We emulate the Internet topologies for two smaller countries, Luxembourg (with 96 ASes) and Zambia (11 ASes), to demonstrate the potential for a digital twin in replicating domestic infrastructure – which could be of benefit to explore the potential changes should a global transit provider join at a national Internet Exchange, or the potential impact on route length should the commercial relationship between two national Internet service providers change.

We construct subgraphs of the complete Internet topology for each of the two emulations, including only the ASes registered to that country. We deploy the scenarios using each of the Quagga and FRRouting suites (which differ slightly in the implementation of BGP policies). The resulting memory usage distribution for each scenario is shown in Fig. 2.

Despite utilising the BGP protocol with functionally the same routing policies, we see that after convergence the memory usage is still substantially higher in the FRRouting BGP implementation despite the identical emulated topology and routing tables. Significantly, we also see that unlike in the Quagga suite implementations where memory usage increases with emulation scale. This seems unusual given the increased routing table size but could

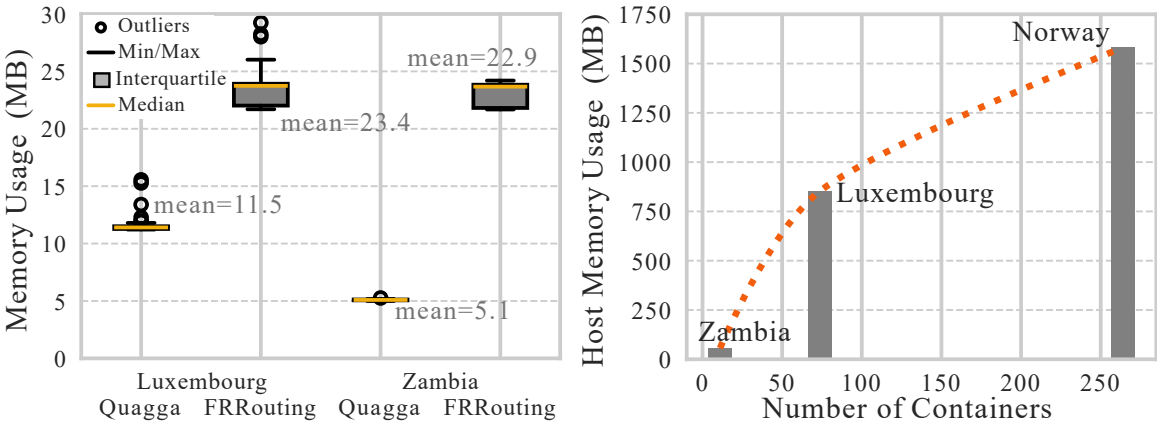

Fig. 2. Emulation memory usage. **(Left)** Per-container memory usage for Luxembourg and Zambia under different routing suites. Despite a tenfold increase in emulation size, memory usage under FRRouting demonstrates little change. **(Right)** The increase in host memory usage is non-linear, demonstrated over a small number of emulation sizes.

be demonstrative of the minimum memory requirements as the table size (which sits in the single-digit KB range) remains insignificant. Notably, however, the Quagga emulation represents an average container memory usage reduction of over 63% (and FRRouting 47%) relative to previous work [13].

We also emulate the country topology of Norway using the Quagga suite. This requires 282 AS containers, but despite the increased emulation size, the host memory requirements (as shown in Fig. 2) increase less than linearly, and in a potential logarithmic relationship. This would be particularly suited to emulating at an increased scale.

V. Conclusion and Future Work

We present a new Internet emulation approach providing a promising building block for an Internet digital twin, with preliminary results suggesting the feasibility of Internet scale replication with relatively low per-container resource usage relative to virtual machine or physical testbeds which require greater expense or higher resource overheads. We incorporate real-world Internet topology and produce a container-based emulation tested using two country-level Internet topologies.

In future work, we intend to evaluate our approach at substantially increased scale using a more complete Internet topology and an wider variety of routing suites and BGP configurations to take a further step towards building an Internet digital twin which can, for example, be used to test different BGP vendor implementations and configurations.

Acknowledgments

This work is supported, in part, by EPSRC and DSIT TMF-uplift: CHEDDAR: Communications Hub For Empowering Distributed ClouD Computing Applications And Research (EP/X040518/1), (EP/Y037421/1), and EPSRC REMOTE (EP/Y019229/1).

References

[1] C. Zhou, H. Yang, X. Duan, D. Lopez, A. Pastor, Q. Wu, M. Boucadair, and C. Jacquenet, "Network Digital Twin: Concepts and Reference Architecture," Internet Engineering Task Force, Internet Draft draft-irtf-nmrg-network-digital-twin-arch-05, Mar. 2024. [Online]. Available: https://datatracker.ietf.org/doc/draft-irtf-nmrg-network-digital-twin-arch-05

[2] Y. Rekhter and T. Li, "A Border Gateway Protocol 4 (BGP-4)," Internet Engineering Task Force, Request for Comments RFC 1654, Jul. 1994. [Online]. Available: https://datatracker.ietf.org/doc/rfc1654

[3] S. Kent, C. Lynn, and K. Seo, "Secure Border Gateway Protocol (S-BGP)," *IEEE Journal on Selected Areas in Communications*, vol. 18, no. 4, pp. 582–592, 2000.

[4] M. Lepinski and K. Sriram, "BGPsec Protocol Specification," Internet Engineering Task Force, Request for Comments RFC 8205, Sep. 2017. [Online]. Available: https://datatracker.ietf.org/doc/rfc8205

[5] X. Zhang, H.-C. Hsiao, G. Hasker, H. Chan, A. Perrig, and D. G. Andersen, "SCION: Scalability, control, and isolation on next-generation networks," in *Proceedings of the 2011 IEEE Symposium on Security and Privacy*, ser. SP '11. USA: IEEE Computer Society, 2011, p. 212–227. [Online]. Available: https://doi.org/10.1109/SP.2011.45

[6] R. Bush and R. Austein, "The Resource Public Key Infrastructure (RPKI) to Router Protocol," Internet Engineering Task Force, Request for Comments RFC 6810, Jan. 2013. [Online]. Available: https://datatracker.ietf.org/doc/rfc6810

[7] C. Testart, J. Wolff, D. Gouda, and R. Fontugne, "Identifying current barriers in RPKI adoption," in *Proceedings of the TPRC2024 The Research Conference on Communications, Information and Internet Policy*, 2024.

[8] B. Cartwright-Cox. (2023, Aug.) Grave flaws in BGP error handling. [Online]. Available: https://blog.benjojo.co.uk/post/bgp-path-attributes-grave-error-handling

[9] T. Strickx. (2019, Jun.) How Verizon and a BGP optimizer knocked large parts of the Internet offline today. [Online]. Available: https://blog.cloudflare.com/how-verizon-and-a-bgp-optimizer-knocked-large-parts-of-the-internet-offline-today/

[10] J. Kwon, J. A. García-Pardo, M. Legner, F. Wirz, M. Frei, D. Hausheer, and A. Perrig, "SCIONLab: A next-generation Internet testbed," in *Proceedings of the IEEE International Conference on Network Protocols (ICNP)*, 2020. [Online]. Available: https://netsec.ethz.ch/publications/papers/icnp2020_scionlab.pdf

[11] B. Schlinker, T. Arnold, I. Cunha, and E. Katz-Bassett, "PEERING: virtualizing BGP at the edge for research," in *Proceedings of the 15th International Conference on Emerging Networking Experiments And Technologies*, ser. CoNEXT '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 51–67. [Online]. Available: https://doi.org/10.1145/3359989.3365414

[12] S. Tabaeiaghdaei, S. Scherrer, J. Kwon, and A. Perrig, "Carbon-aware global routing in path-aware networks," in *Proceedings of the 14th ACM International Conference on Future Energy Systems*, ser. e-Energy '23. New York, NY, USA: Association for Computing Machinery, 2023, p. 144–158. [Online]. Available: https://doi.org/10.1145/3575813.3595192

[13] S. Knight, H. Nguyen, O. Maennel, I. Phillips, N. Falkner, R. Bush, and M. Roughan, "An automated system for emulated network experimentation," in *Proceedings of the ninth ACM conference on Emerging networking experiments and technologies*. Santa Barbara California USA: ACM, Dec. 2013, pp. 235–246. [Online]. Available: https://dl.acm.org/doi/10.1145/2535372.2535378

[14] T. Wirtgen, T. Rousseaux, Q. De Coninck, N. Rybowski, R. Bush, L. Vanbever, A. Legay, and O. Bonaventure, "xBGP: Faster innovation in routing protocols," in *20th USENIX Symposium on Networked Systems Design and Implementation (NSDI 23)*. Boston, USA: USENIX, 2023, pp. 575–592.

[15] Sonic Foundation, "Software for Open Networking in the Cloud (SONiC)," 2023. [Online]. Available: https://sonicfoundation.dev/
[16] H. H. Liu, Y. Zhu, J. Padhye, J. Cao, S. Tallapragada, N. P. Lopes, A. Rybalchenko, G. Lu, and L. Yuan, "CrystalNet: Faithfully emulating large production networks," in *Proceedings of the 26th Symposium on Operating Systems Principles*, ser. SOSP '17. New York, NY, USA: Association for Computing Machinery, 2017, pp. 599–613. [Online]. Available: https://doi.org/10.1145/3132747.3132759
[17] Z. Gao, A. Abhashkumar, Z. Sun, W. Jiang, and Y. Wang, "Crescent: Emulating heterogeneous production network at scale," in *21st USENIX Symposium on Networked Systems Design and Implementation (NSDI 24)*. Santa Clara, CA: USENIX Association, Apr. 2024, pp. 1045–1062. [Online]. Available: https://www.usenix.org/conference/nsdi24/presentation/gao-zhaoyu
[18] J. Levett, V. Vassilakis, and P. Yadav, "Unveiling internet censorship: Analysing the impact of nation states' content control efforts on internet architecture and routing patterns," Feb. 2024.
[19] R. Fontugne, M. Tashiro, R. Sommese, M. Jonker, Z. S. Bischof, and E. Aben, "The wisdom of the measurement crowd: Building the Internet Yellow Pages a knowledge graph for the internet," in *Proceedings of the 2024 ACM on Internet Measurement Conference*, ser. IMC '24. New York, NY, USA: Association for Computing Machinery, 2024, p. 183–198. [Online]. Available: https://doi.org/10.1145/3646547.3688444
[20] S. Anderson, L. Salamatian, Z. S. Bischof, A. Dainotti, and P. Barford, "iGDB: connecting the physical and logical layers of the internet," in *Proceedings of the 2022 ACM on Internet Measurement Conference*, ser. IMC '22. New York, NY, USA: Association for Computing Machinery, 2022, p. 433–448. [Online]. Available: https://doi.org/10.1145/3517745.3561443
[21] NetworkX, "NetworkX," 2005. [Online]. Available: https://networkx.org/
[22] CAIDA, "The CAIDA AS relationships dataset, Jan 2025." [Online]. Available: https://www.caida.org/catalog/datasets/as-relationships/
[23] PeeringDB, "The CAIDA UCSD PeeringDB dataset, Jan 2025." [Online]. Available: https://www.caida.org/catalog/datasets/peeringdb/
[24] CAIDA, "RouteViews prefix to AS mappings dataset for IPv4 and IPv6, jan 2025," 2025. [Online]. Available: https://www.caida.org/catalog/datasets/routeviews-prefix2as/
[25] Port 179 Ltd, "bgp.tools," 2018. [Online]. Available: https://bgp.tools/
[26] P. Jakma and D. Lamparter, "Introduction to the quagga routing suite," *IEEE Network*, vol. 28, no. 2, pp. 42–48, 2014.
[27] L. Foundation. (2017) Frrouting project. [Online]. Available: https://frrouting.org/
[28] M. Scazzariello, L. Ariemma, G. Di Battista, and M. Patrignani, "Megalos: A scalable architecture for the virtualization of large network scenarios," *Future Internet*, vol. 13, no. 9, 2021. [Online]. Available: https://www.mdpi.com/1999-5903/13/9/227