






# EdgeVPN.io: Seamless Software-defined Layer 2 Virtual Networking for Edge Computing

Kensworth Subratie <sup>1\*</sup> and Renato Figueiredo <sup>1\*</sup>

<sup>1</sup> University of Florida, Gainesville, FL, USA \* These authors contributed equally.

DOI: [10.xxxxxx/draft](https://doi.org/10.xxxxxx/draft)

## Software

- [Review](#) 
- [Repository](#) 
- [Archive](#) 

Editor: 

Submitted: 15 February 2024

Published: unpublished

## License

Authors of papers retain copyright<sup>12</sup> and release the work under a Creative Commons Attribution 4.0 International License ([CC BY 4.0](https://creativecommons.org/licenses/by/4.0/))<sup>8</sup>.

## Summary

This paper describes the EdgeVPN.io ([Subratie, 2024](#)) software, a novel technique that enables virtual private Ethernet networks that span edge and cloud resources – including those constrained by NATs and firewalls. EdgeVPN.io has been implemented as an open-source virtual network software solution, and experiments with the software have demonstrated its functionality and scalability. The design and evaluation are discussed further in ([Subratie et al., 2023](#)).

## Statement of need

The advent of virtualization and cloud computing has fundamentally changed how distributed applications and services are deployed and managed. Emerging IoT and smart-spaces applications exhibit requirements that are difficult to meet using existing cloud computing models ([Zhang et al., 2015](#)). With the proliferation of IoT and mobile devices, virtualized systems akin to those offered by cloud providers are increasingly needed geographically near the network's edge to perform processing tasks in proximity to the data sources and sinks. Latency-sensitive, bandwidth-intensive applications can be decomposed into workflows that leverage resources at the edge – a model referred to as fog computing - to bring compute and short-term storage closer to the data sources and sinks. This eliminates the latency and throughput penalties from moving data across large geographic distances and through high contention, bandwidth-limited links. However, it introduces an operation and management problem: it is necessary to interconnect all widely distributed components to create a virtualized computing environment. Unfortunately, software and methodologies designed for the data center are typically poorly suited for fog computing operations along the Internet's edge due to Internet Protocol (IP) constraints. Network virtualization stands at a unique point to address these challenges. While existing Virtual Private Networks (VPNs) can mitigate hurdles such as endpoint addressing and secure communication, current models are infeasible for operation and management at the proposed scale of future IoT applications. A decentralized, scalable system that supports dynamic membership, virtualizes addressable endpoints and provides secure communication is needed.

An illustrative use case of EdgeVPN.io is a software service that improves safety and effectiveness during multi-agency emergency response operations by enabling data-driven strategic and tactical decision-making. The networking core, web services, and applications facilitate the definition, deployment, and creation of ad hoc overlay networks. These virtual networks will span multiple organizations collaborating towards a specific goal, regardless of location, providing the necessary connectivity and confidentiality for intra-group communication across the public Internet. The EdgeVPN.io virtual network aggregates and integrates heterogeneous resources such as IoT sensors and actuators, analytic compute engines, and operation personnel via their client devices (tablets, laptops, phones, etc.) across multiple organizations' networks

for seamless connectivity and interactions.

## Features

The EdgeVPN.io software integrates the following features:

- Self-assembling and self-maintaining overlays that require only the definition of authorized participants.
- Software-defined switching via the OpenFlow (McKeown et al., 2008) protocol and Open vSwitch (Pfaff & Davie, 2013).
- Dynamic membership within overlays allows nodes to join or leave an active overlay.
- Concurrent active and independent layer 2 overlays within a single node.
- Hybrid overlays supporting two tunneling technologies: user-mode WebRTC (Tincan) and kernel-mode GENEVE (Gross et al., 2020) tunnels.
- Tincan tunnels are encrypted and support ICE (Rosenberg, 2010) bootstrapping for NAT traversal.
- GENEVE tunnels provide low latency communications with lower overheads.
- Role selection allows a node to act as a switch or pendant device anchored to a switching node when joining an overlay.

## Design

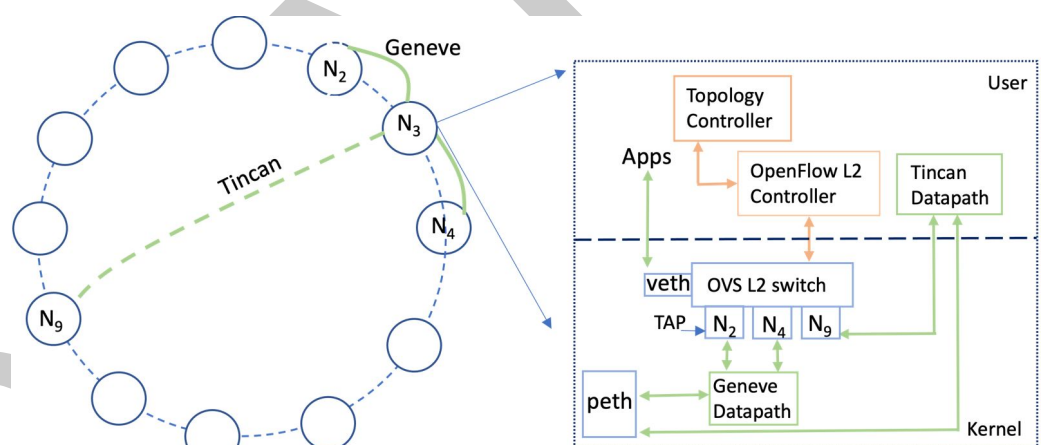


Figure 1: Fig. 1. System Overview.

The goal of EdgeVPN.io is to deliver scalable layer 2 forwarding for dynamic edge and cloud network environments where the peer nodes act as software-defined bridges. EdgeVPN.io integrates a Symphony (1-D Kleinberg routable small-world network (Manku et al., 2003)) topology and a decentralized layer-2 switching into a network fabric. Each node runs three components as depicted in Figure 1: (1) the topology controller creates and maintains the overlay, while (2) the OpenFlow layer 2 controller programs the corresponding switching rules, and (3) Tincan, the default datapath. While each node is parameterized to be independently tuned, they are true peers with identical functional capabilities, and each is independently maintained by its local controllers. While there are no centralized components for overlay management and SDN-programmed switching, it uses XMPP (Saint-Andre, 2004) for peer authentication and messaging, and ICE for endpoint discovery and tunnel bootstrapping.

## Source Code and Packaging

EdgeVPN.io is available as an MIT-licensed open-source project hosted on [Github](#). The two primary repositories are [EdgeVPNio/evio](#) and [EdgeVPNio/tincan](#).

Evio is the Python implementation of SDN controllers for topology, layer 2 switching, and other auxillary functions. Tincan is the EdgeVPN.io default datapath. It is implemented in C++ and creates the fundamental tunnel abstraction consisting of a Linux TAP device and a WebRTC data link. Tincan requires WebRTC source code or prebuilt libraries for compiling. The tools repo provides several scripts that assist with building and packaging.

EdgeVPN.io releases are distributed as a Debian Package for Ubuntu 20 and 22, and hosted for installation via apt-get. A ready-to-run docker image is also hosted publicly for retrieval using docker pull.

## Acknowledgements

This material is based upon work supported by the National Science Foundation, USA under Grants OAC-2004441, OAC-2004323, and CNS-1951816. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

## References

- Gross, J., Ganga, I., & Sridhar, T. (2020). *Geneve: Generic Network Virtualization Encapsulation* (No. 8926). RFC 8926; RFC Editor. <https://doi.org/10.17487/RFC8926>
- Manku, G. S., Bawa, M., & Raghavan, P. (2003). Symphony: Distributed hashing in a small world. *Proceedings of the 4th Conference on USENIX Symposium on Internet Technologies and Systems - Volume 4*, 10–10. <http://dl.acm.org/citation.cfm?id=1251460.1251470>
- McKeown, N., Anderson, T., Balakrishnan, H., Parulkar, G., Peterson, L., Rexford, J., Shenker, S., & Turner, J. (2008). OpenFlow: Enabling innovation in campus networks. *SIGCOMM Comput. Commun. Rev.*, 38(2), 69–74. <https://doi.org/10.1145/1355734.1355746>
- Pfaff, B., & Davie, B. (2013). *The Open vSwitch Database Management Protocol* (No. 7047). RFC 7047; RFC Editor. <https://doi.org/10.17487/RFC7047>
- Rosenberg, J. (2010). *Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal for Offer/Answer Protocols* (No. 5245). RFC 5245; RFC Editor. <https://doi.org/10.17487/RFC5245>
- Saint-Andre, P. (2004). *Extensible Messaging and Presence Protocol (XMPP): Core* (No. 3920). RFC 3920; RFC Editor. <https://doi.org/10.17487/RFC3920>
- Subratie, K. (2024). *EdgeVPNio/evio: Release 24.1.2.1061*. Zenodo. <https://doi.org/10.5281/zenodo.10655929>
- Subratie, K., Aditya, S., & Figueiredo, R. J. (2023). EdgeVPN: Self-organizing layer-2 virtual edge networks. *Future Generation Computer Systems*, 140, 104–116. <https://doi.org/10.1016/j.future.2022.10.007>
- Zhang, B., Mor, N., Kolb, J., Chan, D. S., Lutz, K., Allman, E., Wawrzynek, J., Lee, E., & Kubiatowicz, J. (2015). The cloud is not enough: Saving {IoT} from the cloud. *7th USENIX Workshop on Hot Topics in Cloud Computing (HotCloud 15)*.