Poster: Opportunistic Edge Computing with EVPNs

Stephan Recker
University of Applied Science and Arts
Dortmund, Germany
stephan.recker@fh-dortmund.de

Abstract

Opportunistic Edge Computing forms a pool of resources for delivery of edge computing services leveraging existing nearby infrastructure. The pooled resources are generally owned by different entities and unreliable. Workload hosted on such resources must be movable in case of infrastructure down-times. Additionally, distributed applications are requiring a one-to-many connectivity model rather than client-to-server connectivity. Therefore, the underlay transport network needs to cater for these requirements. This article proposes a concept that meets these based on standard EVPN/VXLAN tunnels. In order to avoid additional latency resulting from NAT and tromboning, EVPNs will need to be extended to the client. Besides introducing the networking concept also a critical assessment of resulting constraints for clients will be provided.

CCS Concepts: • **Networks** → **Network design principles**; • **Computing methodologies** → **Distributed algorithms**.

Keywords: edge computing, opportunistic computing, EVPN, VXLAN, multicast DNS, peer-to-peer networks

ACM Reference Format:
Stephan Recker. 2023. Poster: Opportunistic Edge Computing with EVPNs. In *The Eighth ACM/IEEE Symposium on Edge Computing (SEC '23), December 6–9, 2023, Wilmington, DE, USA.* ACM, New York, NY, USA, 3 pages. https://doi.org/10.1145/3583740.3626618

1 Motivation

Opportunistic Edge Computing (OEC) provides instant access to a shared pool of computing resources opportunistically built using nearby infrastructures[6]. Nearby infrastructure can include any device with computing capacity which are generally considered unreliable with planned and unplanned off-periods. Individual compute resources provided within an OEC framework are owned and operated by different entities. Workload processed by a particular OEC resource must be movable in the event of a compute resource off-period. Standard workload mobility within virtualization platforms requires one virtualization management platform across distributed (micro) data centers, which cannot be assumed for individually owned OEC resources. Therefore manual workload transfer needs to be supported without altering the network address. Additionally, distributed applications with multiple micro-services provided through different compute resources demand for a one-to-many connectivity model rather than a one-to-one client-server relationship.

SEC '23, December 6–9, 2023, Wilmington, DE, USA

ACM ISBN 979-8-4007-0123-8/23/12.
https://doi.org/10.1145/3583740.3626618

2 Service Discovery for Opportunistic Edge Computing

Service Discovery in OEC can be based on centralized infrastructure or implemented in a distributed fashion, the latter often through deploying *peer-to-peer* (p2p) networks. For example, for inter-pod connectivity between kubernetes pods located at different sites, a p2p network based on Kademlia[5] is used to build an overlay of hosts with layer 3 tunnels. This article follows the approach of [2] by deploying a p2p network to facilitate computing capacity providers to instantly join the p2p network representing the pool of shared OEC resources. However, in order to limit the number of peers in the p2p network, in the proposed solution clients requesting OEC services shall first explore available resources in the access domain they are connected to. Discovery of a *Local Edge Server* (LES) shall be based on *Multicast DNS-Service Discovery* (mDNS-SD). Only when an existing LES rejects workload acceptance, a wider service discovery using the p2p network shall be invoked, in which the LES performs the discovery on behalf of the client. Figure 1 depicts the segregation into short-haul mDNS-SD discovery and backhaul p2p network discovery.

A peer joins the p2p network requesting a list of logically "close" peers from a known bootstrap function. The bootstrap node can be implemented by any peer in the p2p network and the bootstrap function shall regularly migrate to other peers to a avoid quasi-centralized instances carrying the burden of delivering this capability. Thus, the bootstrap workload also needs to be movable. The LES leverages latency-based peer selection to determine the optimal peer for a client. Then these peers are contacted by the LES to check for available capacity and the closest peer with available capacity is reported back to the client, who establishes a direct service relationship with the identified peer. In cases where an access domain does not have a LES a default *Proxy-Local Edge*

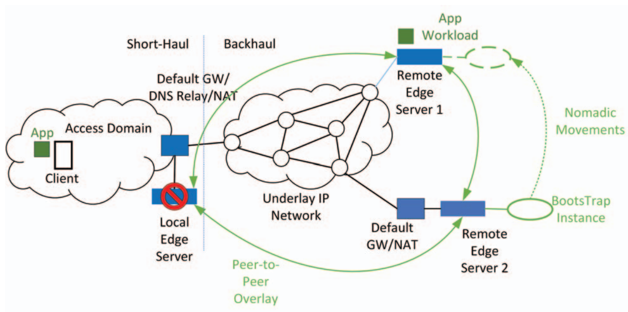

Figure 1. Backhaul Layer Peer-to-Peer Network with Bootstrap Function

function can be contacted. This proxy can be implemented in the same nomadic way as the bootstrap function to avoid one peer permantly having a quasi-centralized role.

3 EVPN/VXLAN Network

One standard network solution for workload mobility is based on *Virtual Extended LAN* (VXLAN) for extending layer 2 domains to distant locations and *Ethernet Virtual Private Network* (EVPN) for the corresponding address learning through *Multi Protocol BGP* (MP-BGP)[3]. Many hypervisors support both protocols natively so that the virtualisation host becomes a *Virtual Tunneling Endpoint* (VTEP). These VTEPs being the peers of the overlay p2p network establish VXLAN tunnels among themselves for the VLANs that host the bootstrap and Proxy-LE functions. Each edge server can have additional VLANs for workload hosted from clients. Figure 2 depicts the logical VXLAN virtual tunnels denoted by the corresponding *Virtual Network Identifier* (VNI).

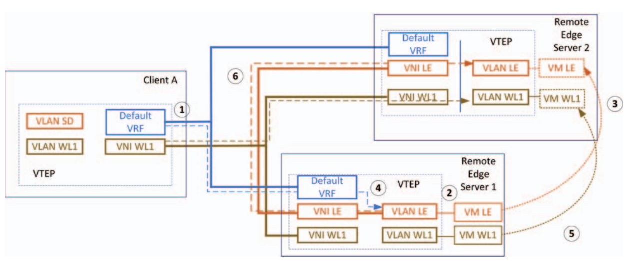

Figure 2. VXLAN Concept for Proxy-LE and Workload

For service discovery client A contacts the Proxy-LE (VM-LE) at *Remote Edge Server 1* (RES1) using the default *Virtual Routing Function* (VRF) demarcated with (1) and (2) in Figure 2. The Proxy-LE responds with edge server RES1 being identified for hosting the client's workload WL1 also through the global routing domain. To connect WL1 to client A, VLAN WL1 will be extended through a tunnel with VNI WL1 to the client together with a MP-BGP neighborship for address learning. In the event of a planned downtime of RES1, the client's workload can be moved to a different RES. RES1 first negotiates with RES2, who shall be the closest RES, to take over the workload and informs the client about the upcoming workload movement. RES2 then creates the corresponding VLAN WL1 and establishes a MP-BGP neighborship with the client. Route Targets for the extended VLAN can be set such that a hub and spoke topology is created with the client being the hub. Upon workload movement (demarcated with (5) in Figure 2), the VTEP in RES2 learns about the arrival of VM WL1 e.g. through *Gratuitous ARP* and sends a BGP EVPN type 2 update to all neighbors. Upon receipt of the update client A will send all traffic for WL1 to VTEP2 (6). Without including the client in the EVPN workload tromboning from RES1 to RES2 would have occurred leading to additional latency.

The bootstrap and Proxy-LE instances with static IP addresses move nomadically from peer to peer, in the illustrated example the Proxy-LE function moves with its VM to RES2. The corresponding VLAN LE is extended through the tunnel with VNI LE to RES2. The public IP address of the Proxy-LE is still known in the global routing domain to be available through RES1. Therefore, after LE migration a client service discovery request will still reach RES1, the local VTEP will identify VLAN LE for the Proxy-LE addresses subnet. From the previously received BGP update RES 1 knows that the MAC-Adress corresponding to the Proxy-LE is now located at RES 2 and forwards the packet accordingly through tunnel VNI LE (demarcated with (4) in Figure 2). The response is sent the same way back as RES1 still hosts VLAN LE's layer 3 gateway. This extended VLAN facilitates instantaneous mobility of the LE-VM however at the expense of tromboning of service discovery signalling flows.

4 Critical Assessment and Outlook

The proposed solution provides the needed capabilities for workload mobility and one-to-many client connections. The burden on the client of running VXLAN and MP-BGP needs to be carefully assessed. For powerful terminals Linux *Free Range Routing* (FRR)[1] can be instantly depoyed while for small form factor *IOT* devices deployments may not be feasible. Bandwidth efficiency is slightly impacted by the additional 8 Byte VXLAN header. The currently deployed resource allocation is overly simple, more sophisticated approaches as surveyed in [4] could be deployed. Security aspects such as tunnel encryption or client authentication have not been considered so far. Future work will need to look into these aspects in detail.

Acknowledgments

This work was conducted as part of project EMULATE funded by the German Federal Ministry of Economics and Climate Action under research grant 13IPC012.

References

[1] 2023. FRR Routing Project.

[2] Renato Figueiredo and Kensworth Subratie. 2020. Edgevpn. io: Open-source virtual private network for seamless edge computing with kubernetes. In *2020 IEEE/ACM Symposium on Edge Computing (SEC)*. IEEE, 190–192.

[3] A Shaji George and AS Hovan George. 2021. A Brief Overview of VXLAN EVPN. *Ijireeiceinternational Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering* 9, 7 (2021), 1–12.

[4] Quyuan Luo, Shihong Hu, Changle Li, Guanghui Li, and Weisong Shi. 2021. Resource Scheduling in Edge Computing: A Survey. *IEEE Communications Surveys & Tutorials* 23 (2021), 2131–2165.

[5] Petar Maymounkov and David Mazieres. 2002. Kademlia: A peer-to-peer information system based on the xor metric. In *International Workshop on Peer-to-Peer Systems*. Springer, 53–65.

[6] Richard Olaniyan, Olamilekan Fadahunsi, Muthucumaru Maheswaran, and Mohamed Faten Zhani. 2018. Opportunistic edge computing: concepts, opportunities and research challenges. *Future Generation Computer Systems* 89 (2018), 633–645.