2021 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS) | 978-1-6654-4893-2/21/$31.00 ©2021 IEEE | DOI: 10.1109/ANTS52808.2021.9936995

# Improving Service Performance through Multilayer Routing and Service Intelligence in a Network Service Mesh

Boyang Hu*, Deepak Nadig† and Byrav Ramamurthy*
*Dept. of Computer Science & Engg, University of Nebraska-Lincoln, Lincoln, NE, USA
{bhu, byrav}@cse.unl.edu
†Dept. of Computer and Information Technology, Purdue University, West Lafayette, IN, USA
nadig@purdue.edu

***Abstract*—Network service mesh architectures, by interconnecting cloud clusters, provide access to services across distributed infrastructures. Typically, services are replicated across clusters to ensure resilience. However, end-to-end service performance varies mainly depending on the service loads experienced by individual clusters. Therefore, a key challenge is to optimize end-to-end service performance by routing service requests to clusters with the least service processing/response times. We present a two-phase approach that combines an optimized multi-layer optical routing system with service mesh performance costs to improve end-to-end service performance. Our experimental strategy shows that leveraging a multi-layer architecture in combination with service performance information improves end-to-end performance. We evaluate our approach by testing our strategy on a service mesh layer overlay on a modified continental united states (CONUS) network topology.**

***Keywords*—service mesh, multi-layer optical network, service performance**

## I. INTRODUCTION

The advent of cloud computing has led to radical change through the rapid development and delivery of applications. Through cloud-native applications and microservice architectures, operators can quickly provide services, increase agility, and increase flexibility in distributed cloud environments. To fully exploit the benefits of cloud-native systems, automating the deployment and orchestration of microservices is necessary. Cloud-native architectures ensure flexibility, programmability, and resilience by disaggregating cloud computing systems' control and data planes. A key challenge of such an approach is to ensure reliable service-to-service communication between distributed infrastructures.

A network service mesh (or service mesh) [1] is a software infrastructure layer for controlling and monitoring internal, service-to-service traffic in microservices applications. Typically, it consists of a "data plane" of network proxies that have been deployed alongside the application code and a "control plane" that is used to interact with the proxies. This new model empowers cloud platform operators with a set of new tools for ensuring the reliability, security, and visibility of their network, while the service owners, typically the application developers, are unaware of the existence of the service mesh. Network service meshes have emerged as a valuable pattern for inter-service communication by providing an addressable infrastructure layer. Network service meshes provide access to services across distributed infrastructures. Services are replicated across cloud clusters to ensure resilience. However, end-to-end service performance varies mainly depending on the service loads experienced by individual clusters.

Thus, a key challenge is to ensure that end-to-end service performance is optimized by routing service requests to clusters with the least service processing/response times. Typically, networked services across distributed infrastructures employ a dedicated high-speed optical layer backbone to ensure service connectivity. Large cloud infrastructure providers rely on carrier networks for high-speed wide area network (WAN) connectivity. Today, cloud providers and carrier networks manage and optimize their infrastructures independently. While numerous cloud-carrier integration efforts (e.g., Microsoft-AT&T [2], Amazon-Verizon [3]), numerous challenges exist. Therefore, a two-phase approach that combines routing optimizations at the optical network backbone layer and service performance optimization at the service mesh layer is essential to ensure predictable end-to-end service performance.

Numerous research efforts have been directed towards inter-service communication for networked services, including service composition, service monitoring, load balancing (client-side), and traffic management. The management of inter-service communication traffic relies on network proxies, specialized packet encapsulation (e.g., network service header (NSH) [4]), or overlay networks (e.g., Open vSwitch [5], Tungsten Fabric [6]). These solutions either require modified packet processing libraries (specific packet headers) or full overlay network implementations, which add complexity to application and service development.

In this paper, we propose a multi-layer architecture for service mesh network that considers both the service mesh network costs and the optical network routing costs. We develop a strategy to improve end-to-end service performance by combining route optimizations of a multi-layer optical network with service performance insights from individual

cloud clusters.

This paper is organized as follows: in Section II, we provide a background of service mesh networks and layered router simulator used in this paper; we also discuss the related work; in Section III, we detail our multi-layer service mesh network architecture, associated cost models, our experimental network topology and present our solution approach; in Section IV, we present the simulation results on different network scenarios; in Section V, we conclude our work and discuss the future work.

## II. Background and Related Work

### A. Network Service Mesh

Network service mesh (NSM) is an addressable infrastructure layer that facilitates inter-services communication across distributed clouds. In a modern, cloud-native application, the service mesh ensures reliable service request delivery through a complex services network. This is typically achieved by deploying a number of lightweight network proxies alongside application code; the proxies are transparent to the applications running in the service mesh. Service mesh architectures, through declarative control, allow network management, monitoring, and policy-based network control. Service meshes ensure uniform observability of traffic through granular traffic control, service discovery, and resilience features.

Inter-cluster service-to-service communication operation between two clusters with replicated control planes is shown in Figure 1. The process is as follows: (i) The ingress gateways control inbound traffic into the service mesh. The ingress proxy is akin to a reverse proxy and forwards inbound traffic to the appropriate service. (ii) Service proxies (placed between the service and the application traffic) ensure traffic observability, control, and policy enforcement. The proxies hand over the traffic to the services. (iii) The service routes the application traffic to the workloads for processing, and the response is forwarded to an egress gateway. (iv) The egress gateway forwards the traffic to the ingress gateway controller of a neighboring cluster. Mutual transport layer security (mTLS) is used to encrypt the traffic between the two clusters. (v) The ingress gateway, like before, forwards the traffic to the appropriate service. (vi) Service proxies ensure uniform observability across the clusters. (vii) The egress gateway sends the response back to the client/application. While we assume that the clusters provide the services, we note that using a similar approach we can integrate services provided by application monoliths, virtual machines (VM) and baremetal workloads.

### B. Related Work

Network service mesh architectures are employed in various application domains, including internet of things (IoT) [7], next-generation mobile networks [8], network security [9], [10], network monitoring [11], resource management [12], and network virtualization [13], [14]. The authors in [15], [16] present an overview of service meshes and provide architectural guidance for deploying NSMs. Numerous works focus on employing NSM architectures for domain-specific solutions. The authors in [17] proposed a skewness-aware matrix factorization (SMF) method to model pairwise RTTs for inter-service communication. They demonstrate that SMF finds a good balance between low-rank matrix factorization and skewed distributions. However, the work focuses only on monitoring pairwise RTTs for performance. Other approaches focusing on latency predictions using the matrix factorization approach include [18]–[23].

Numerous studies focus on modeling optical multi-layer networks, and different cost models have been proposed based on specific network architecture designs. In [24], the authors study the network function virtualization (NFV) placement problem on an optical metro network. The proposed cost model considers both capital and operational expenditures, and the simulation results show that an efficient NFV placement strategy can reduce the cost of service provisioning. The cost of IP backbone networks continues to be an increasing concern for network service providers. The use of dual routers for redundancy at each point of presence (PoP) and the core backbone routers (BR) result in significant cost in the daily operation of the IP backbone network. In [25], the authors aim to address the reliability challenges due to the failures and the planned outages of the backbone network. They fundamentally redesign the IP backbones by reducing the redundant backbone routers and leveraging the agile optical transport layer to carry traffic to the remote BRs using an integer linear programming (ILP) solution. Based on their cost and performance evaluation, the proposed designs show a significant cost reduction for an approximately equal level of reliability to current designs. The authors in [26] propose dynamic resource allocation techniques for addressing and routing user demands using wavelength allocation in multi-layer optical backbone networks. Their approach employs a dynamic resource provisioning in cloud networks where the requested data center resources are unknown. The cost model in their study considers both the operating expenditure and the capital expenditure. Our work focuses on developing a multi-layer solution for routing service requests across distributed service mesh architectures by optimizing costs at both the optical and the service planes.

## III. multi-layer Service Mesh Architecture

In this section, we introduce a multi-layer service mesh architecture that incorporates the IP/MPLS layer and WDM layer. The service mesh architecture comprises two planes, (i) the data plane and (ii) the control plane. The data plane handles routing, packet forwarding, load balancing, and policy enforcement, and the control plane for monitoring, network state management, service discovery, identity management, policy, and configuration.

First, we present an exemplary service mesh topology and its operation in Fig. 1. Each service mesh comprises numerous services that are typically separated by trust boundaries (within or across clusters). Service proxies or sidecar containers transparently intercept the service traffic and are

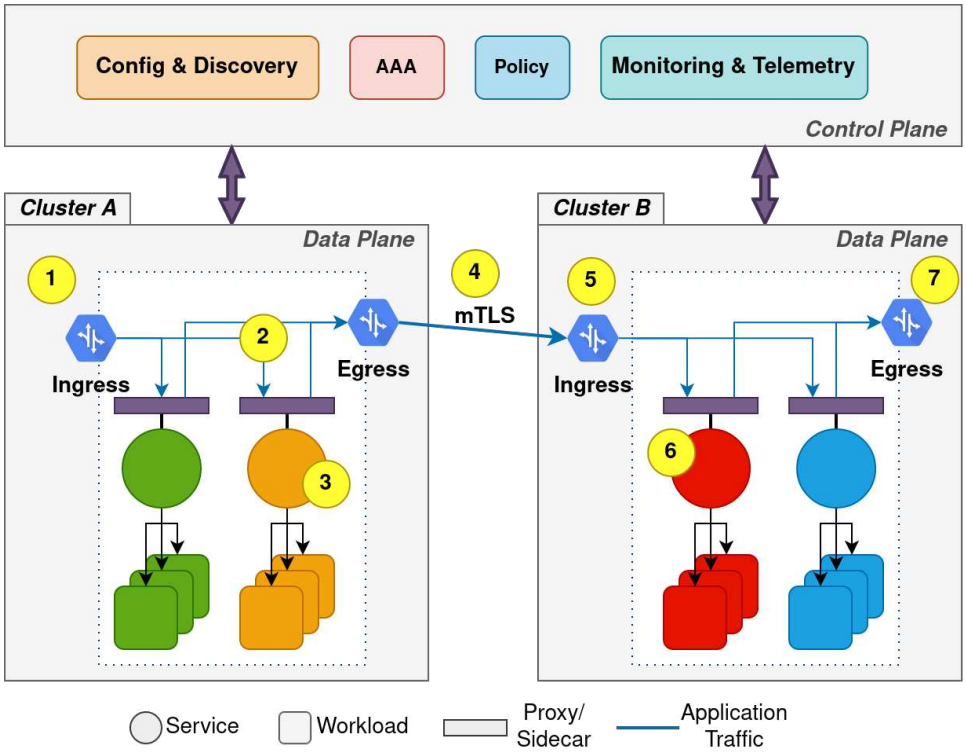


Fig. 1: Service Mesh Architecture and Components.

responsible for routing, monitoring, authorization, authentication, and auditing. The service mesh's data plane manages both inter-cluster and intra-cluster traffic using ingress and egress gateways.

Figure 2 shows the architecture and components of a typical service mesh network that is overlayed on an multilayeroptical network (with IP/MPLS and WDM layers). While the IP/MPLS layer forwards packets based on either IP addresses or MPLS labels, the WDM layer relies on reconfigurable optical add-drop multiplexer (ROADMs) to forward traffic from the IP/MPLS nodes (or routers) that are connected to it. The key advantages of the IP/MPLS-over-WDM network can be summarized as follows. The IP/MPLS-over-WDM network inherits the flexibility provided by IP protocols. IP/MPLS-over-WDM aims to achieve real-time (on-demand) provisioning in optical networks. The WDM technology multiplexes a group of wavelengths and transmits them through a single optical fiber. Currently, each wavelength usually carries 40 Gbps or 100 Gbps data streams, and the capacity of the WDM networks is slated to increase in the future. As the traffic reaches the IP/MPLS node, it will be forwarded to the corresponding node in the lower layer (WDM node). Then, the traffic will be forwarded into the destination WDM node through a lightpath circuit in the WDM layer. The WDM node forwards the traffic into the corresponding node in the upper layer (IP/MPLS node). In the following, we present the cost models for both the optical and service mesh layers.

### A. Optical Layer Cost Model

In this section, we discuss the two different cost sources that are used in our proposed architecture. The cost of the IP-optical backbone network and the other is the cost associated with the service mesh network.

Backbone network costs include both the cost of optical transport equipment used to connect routers as well as the cost of the routers themselves (such as chassis or line cards). The unit cost of each optical component is based on the detailed multi-layer cost model proposed by Huelsermann et al., in [27]. These costs can be further grouped into three different categories: (i) transport layer cost, (ii) router cost, and (iii) network cost. We discuss the cost information in detail below:

*1) Transport Layer Cost:* For a given optical circuit, we compute the costs associated with each transport layer element between two routers encountered along that path. This includes optical transponders, regenerators, and muxponders. In addition, for some of the transport equipment shared by multiple circuits like amplifiers, ROADMs, and pre-deployed fibers, we use an amortized common cost contribution to each circuit on a per wavelength-km basis.

For a typical 40G circuit, the cost includes the transponder cost, generator cost, and thecost of the WDM links on the circuits' path. Thus, the circuit cost is given by 2 * (Cost of 40G transponders + 40G generators).

*2) Router Cost:* For each circuit in our network topology, we evaluate the number of ports required on each router to obtain an approximate the required number of line cards and

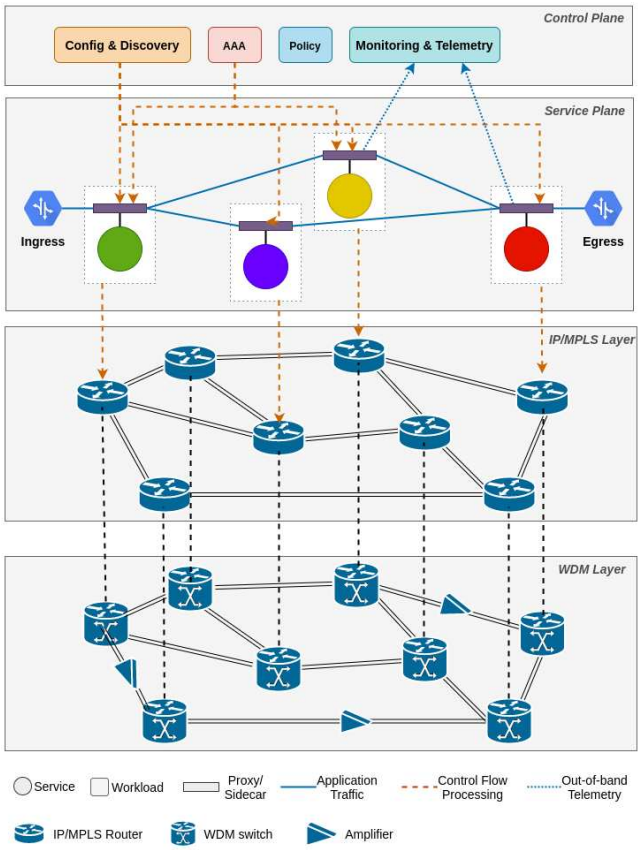


Fig. 2: multi-layer service mesh network topology

chassis.

*3) Network Cost:* Equipment prices (normalized) are obtained from [27] to compute the network cost. While these prices vary over time, it will not affect our cost model and is used only as an example.

### B. Network Service Mesh Cost Model

We develop our service mesh and cluster cost model on pricing data from major cloud providers to obtain real-time cost information of various application and service workloads running in the cloud. We evaluate the total per-hour cost for each application/service instance using CPU, GPU, RAM, and provisioned storage information within each cluster or data center. Typically, similar applications hosted in different cloud data centers can exhibit cost characteristics based on resource availability, service loads, and scheduling. Our cost model relies on the normalized sum of the base cost of the cloud resources used by an application instance. Thus, we define the total instance hourly cost as:

$$I_{\text{total}} = C_{\text{cpu}} * N_{\text{cpu}} + C_{\text{gpu}} * N_{\text{gpu}} + C_{\text{ram}} * N_{\text{ram}} + C_{\text{pv}} * N_{\text{pv}} + C_{\text{net}} * N_{\text{net}} \quad (1)$$

where, *pv* and *net* represents the persistent storage volume and network resources, respectively. The total cost of each resource in the cloud data center is computes as the product of the normalized cost of the resource with the total number of resources used by the application.

### C. Network Topology

Our experimental network topology is shown in Figure 3. It consists of a service mesh layer overlayed on a multi-layer optical network. Our experiment network topology is modified based on CORONET Continental United States (CONUS) network topology, including 43 nodes and 51 links. The upper layer represents the service mesh network layer which consists of two clients (Seattle and Houston) and six service clusters in different locations. The lower layer represents the optical network layer.

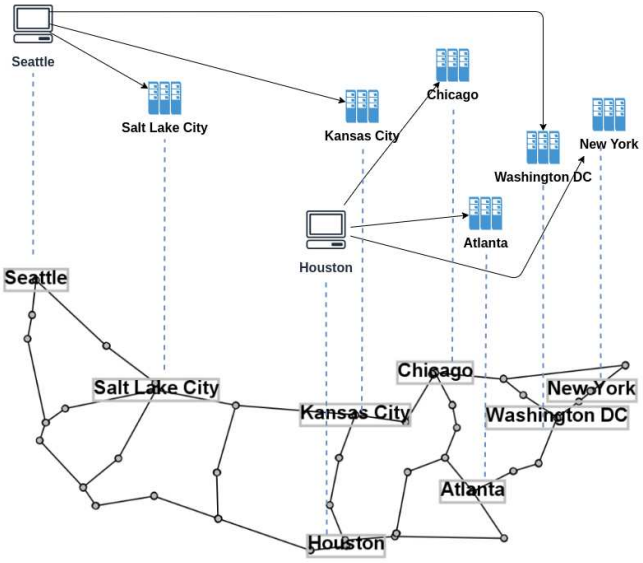


Fig. 3: Example of Service Requests in multi-layer Network

### D. Solution Approach

In order to optimize the service mesh network requests in a multi-layer network, we proposed a three phases solution approach. First, finding the optimal route in the optical layer; second, compute the optimal route in the service mesh network layer; and last, choose the lowest cost route for the service mesh requests across the multi-layer architecture based on the first and second phases.

As an illustrative example, consider a service request that can be processed by two clusters, *DC1* and *DC2*, with $\text{RTT}_{DC1} < RTT_{DC2}$. As optical path choices are not transparent to the service mesh layer, the service request is always sent to *DC1*. However, as the service load on *DC1* increases, the service processing delays exceed the transmission delays of the provisioned optical path. This results in increased service processing cost for all service requests that are routed to *DC1*. Thus, we develop a two-phase strategy in our work, where we choose an alternative (often longer) path to improve end-to-end service performance.

Algorithm 1 shows the proposed optimal route choice strategy. The algorithm processes service requests **S** from our

TABLE I: Cost between clients and service clusters in optical layer

| Request | Client | Service Cluster | Cost ($O_1$) | Client | Service Cluster | Cost ($O_2$) | Total Cost ($O_1 + O_2$) |
|---|---|---|---|---|---|---|---|
| 1 | Houston | Atlanta | 220.65 | Seattle | Salt Lake City | 187.35 | 408.01 |
| 2 | Houston | Atlanta | 220.65 | Seattle | Kansas City | 399.08 | 619.74 |
| 3 | Houston | Atlanta | 220.65 | Seattle | Washington DC | 596.31 | 816.97 |
| 4 | Houston | New York | 421.63 | Seattle | Salt Lake City | 187.35 | 608.98 |
| 5 | Houston | New York | 421.63 | Seattle | Kansas City | 399.08 | 820.71 |
| 6 | Houston | New York | 421.63 | Seattle | Washington DC | 596.31 | 1,017.94 |
| 7 | Houston | Chicago | 287.44 | Seattle | Salt Lake City | 187.35 | 474.79 |
| 8 | Houston | Chicago | 287.44 | Seattle | Washington DC | 399.08 | 686.52 |
| 9 | Houston | Chicago | 287.44 | Seattle | Washington DC | 596.31 | 883.75 |

experimental topology $\mathbf{N}$. The algorithm will first compute the optimal route in the physical layer based on the multi-layer optical network cost model. For each client $\mathbf{c}$ and service clusters $\mathbf{s}$ pairs, we compute the total cost $\mathbf{I_{cs}}$, sorted them in ascending order, and stored the results in $\mathbf{I}$ for phase 3. For each service request in the service network layer, if the service is presented in a cluster, we compute the total cost $\mathbf{I_{total}}$ of the request from the client to that cluster based on the service mesh network cost model we proposed. We then sorted and returned all the costs between client and service cluster pairs in $\mathbf{I}'$ at the end. In the final phase, based on different network load in the service mesh layer, we check the min-cost in both $\mathbf{I}$ and $\mathbf{I}'$ and output the best route for that service request.

**Algorithm 1** Optimal Route Choice Strategy

**Input:** Network topology $\mathbf{N}$, Service requests $\mathbf{S}$
**Output:** Optimal route for each service requests $\mathbf{R}$

Network initialization
Service requests initialization

*Phase 1 – Finding Optimal Route in Optical Layer*

**for** each client server pairs **do**
  Compute cost $I_{cs}(i), \forall i \in S_i$
  $I \leftarrow I_{cs}(i)$
**end for**
sort and return $\mathbf{I}$

*Phase 2 – Finding Optimal Route in Service Mesh Layer*

**for** each service requests $S$ **do**
  **if** $j \neq 0$ **then** ▷ if service available in cluster j
    Compute $I_{total}(i,j), \forall i \in S_i, \forall j \in \{0,1\}$
    $I' \leftarrow I_{total}(i,j)$
  **end if**
**end for**
sort and return $\mathbf{I}'$

*Phase 3 – Final phase*

**for** each best route in Phase 1 & 2 **do**
  $OPTIMALset(R) = \min(\mathbf{I} + \mathbf{I}')$
**end for**
`OPTIMALroute` $= [R]$

## IV. Results and Discussion

In our experiment, we demonstrate a scenario where two clients request services simultaneously from six different potential service clusters. The service that client Seattle requests are located at Salt Lake City, Kansas City, and Washington DC clusters, and the service that Houston requests is located at Chicago, Atlanta, and New York clusters. These two clients are requesting services at the same time. First, we compute the cost between each client and its respective clusters in the optical layer based on the detailed cost model from [27], the total cost is normalized based on the 10G, LH WDM transponder cost. Detailed results as shown in Table I. There are nine different client-server combinations in our experiments. The total cost of the Houston and Atlanta pair and the Seattle and Salt Lake City pair is the lowest among all the experiments. Final results are stored in a cost set in ascending order for later use. Then, we compute the cost of the service mesh layer.

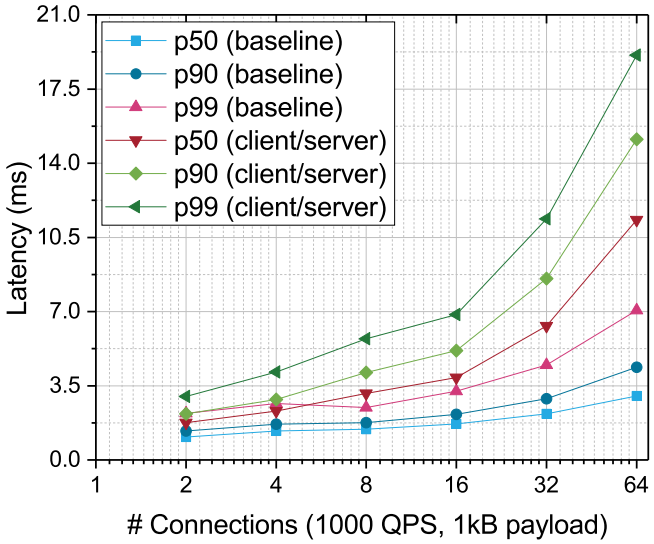


Fig. 4: Service performance with increasing loads.

The service performance at a given cluster increases exponentially with increasing service loads, as shown in Figure 4. Thus, in our final phase, we pick the lowest cost combination from both phase 1 and phase 2 and get the optimal routes to improve end-to-end service performance.

Our experiments demonstrate that the baseline approach that relies only on layered router costs or the service mesh network costs is not optimal when considering the network load in the system or the complexity of the network topology and service requests. Incorporating the optimal combination of service performance costs with the layered router costs can result in better end-to-end service performance.

## V. Conclusions

In this work, we proposed a multi-layer service mesh network model which considered the optical routing layer for optimizing the performance of the service mesh network and an optimal route choice strategy algorithm to choose the best route for the service request. As far as we know, this is the first attempt to integrate the service mesh network with the lower physical optical network layer. Based on our simulation results, we can optimize the path choice of service mesh requests, ensure the end-to-end service performance, and lower the overall costs of the service mesh network by using the multi-layer architecture. In our future work, we plan to run more experiments towards different network topologies with different network load and service requests and automate the whole process of choosing the optimal route.

## VI. Acknowledgements

This material is based upon work supported by the National Science Foundation under Grant Number CNS-1817105.

## References

[1] W. Li, Y. Lemieux, J. Gao, Z. Zhao, and Y. Han, "Service mesh: Challenges, state of the art, and future research opportunities," in *2019 IEEE International Conference on Service-Oriented System Engineering (SOSE)*. IEEE, 2019, pp. 122–1225.

[2] Microsoft, "AT&T and Microsoft announce a strategic alliance to deliver innovation with cloud, AI and 5G," *Microsoft News Center*, Jul 2019. [Online]. Available: https://news.microsoft.com/2019/07/17/att-and-microsoft-announce-a-strategic-alliance-to-deliver-innovation-with-cloud-ai-and-5g/

[3] G. Elissaios, "AWS and Verizon Expand 5G Collaboration with Private MEC Solution," *Amazon AWS*, Apr 2021. [Online]. Available: https://aws.amazon.com/blogs/industries/aws-and-verizon-expand-5g-collaboration-with-private-mec-solution/

[4] P. Quinn, U. Elzur, and C. Pignataro, "Network service header (NSH)," in *RFC 8300*. RFC Editor, 2018.

[5] B. Pfaff, J. Pettit, T. Koponen, E. J. Jackson, A. Zhou, J. Rajahalme, J. Gross, A. Wang, J. Stringer, P. Shelar, K. Amidon, and M. Casado, "The Design and Implementation of Open VSwitch," in *Proceedings of the 12th USENIX Conference on Networked Systems Design and Implementation*, ser. NSDI'15. USA: USENIX Association, 2015, p. 117–130.

[6] "Tungsten Fabric," Available: https://tungsten.io, Last accessed on August 19, 2020.

[7] H.-L. Truong, L. Gao, and M. Hammerer, "Service architectures and dynamic solutions for interoperability of IoT, network functions and cloud resources," in *Proc. 12th European Conference on Software Architecture: Companion Proceedings*, ser. ECSA '18. New York, NY, USA: Association for Computing Machinery, Sep. 2018, pp. 1–4.

[8] B. Dab, I. Fajjari, M. Rohon, C. Auboin, and A. Diquélou, "Cloud-native Service Function Chaining for 5G based on Network Service Mesh," in *ICC 2020 - 2020 IEEE International Conference on Communications (ICC)*, Jun. 2020, pp. 1–7.

[9] R. Chandramouli and Z. Butcher, "Building Secure Microservices-based Applications Using Service-Mesh Architecture," 2020-05-27 2020.

[10] M. Kang, J.-S. Shin, and J. Kim, "Protected Coordination of Service Mesh for Container-Based 3-Tier Service Traffic," in *2019 International Conference on Information Networking (ICOIN)*, Jan. 2019, pp. 427–429.

[11] H. Johng, A. K. Kalia, J. Xiao, M. Vuković, and L. Chung, "Harmonia: A Continuous Service Monitoring Framework Using DevOps and Service Mesh in a Complementary Manner," in *Service-Oriented Computing*, S. Yangui, I. Bouassida Rodriguez, K. Drira, and Z. Tari, Eds. Cham: Springer International Publishing, 2019, pp. 151–168.

[12] X. Xie and S. S. Govardhan, "A Service Mesh-Based Load Balancing and Task Scheduling System for Deep Learning Applications," in *2020 20th IEEE/ACM International Symposium on Cluster, Cloud and Internet Computing (CCGRID)*, May 2020, pp. 843–849.

[13] R. Loomba, T. Metsch, L. Feehan, and J. Butler, "A Hybrid Fitness-Utility Algorithm for Improved Service Chain Placement," in *2018 IEEE Global Communications Conference (GLOBECOM)*, Dec. 2018, pp. 1–7.

[14] G. Sun, Y. Li, D. Liao, and V. Chang, "Service Function Chain Orchestration Across Multiple Domains: A Full Mesh Aggregation Approach," *IEEE Transactions on Network and Service Management*, vol. 15, no. 3, pp. 1175–1191, Sep. 2018.

[15] W. Li, Y. Lemieux, J. Gao, Z. Zhao, and Y. Han, "Service Mesh: Challenges, State of the Art, and Future Research Opportunities," in *2019 IEEE International Conference on Service-Oriented System Engineering (SOSE)*, Apr. 2019, pp. 122–1225.

[16] A. El Malki and U. Zdun, "Guiding Architectural Decision Making on Service Mesh Based Microservice Architectures," in *Software Architecture*, T. Bures, L. Duchien, and P. Inverardi, Eds. Cham: Springer International Publishing, 2019, pp. 3–19.

[17] Y. Fu, D. Li, P. Barlet-Ros, C. Huang, Z. Huang, S. Shen, and H. Su, "A Skewness-Aware Matrix Factorization Approach for Mesh-Structured Cloud Services," *IEEE/ACM Transactions on Networking*, vol. 27, no. 4, pp. 1598–1611, Aug. 2019.

[18] Y. Mao, L. K. Saul, and J. M. Smith, "IDES: An Internet Distance Estimation Service for Large Networks," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 12, pp. 2273–2284, Dec. 2006.

[19] J. Zhu, P. He, Z. Zheng, and M. R. Lyu, "Online QoS Prediction for Runtime Service Adaptation via Adaptive Matrix Factorization," *IEEE Transactions on Parallel and Distributed Systems*, vol. 28, no. 10, pp. 2911–2924, Oct. 2017.

[20] R. Zhu, D. Niu, and Z. Li, "Robust web service recommendation via quantile matrix factorization," in *IEEE INFOCOM 2017 - IEEE Conference on Computer Communications*, May 2017, pp. 1–9.

[21] Y. Fu and X. Xiaoping, "Self-Stabilized Distributed Network Distance Prediction," *IEEE/ACM Transactions on Networking*, vol. 25, no. 1, pp. 451–464, Feb. 2017.

[22] B. Liu, D. Niu, Z. Li, and H. V. Zhao, "Network latency prediction for personal devices: Distance-feature decomposition from 3D sampling," in *2015 IEEE Conference on Computer Communications (INFOCOM)*, Apr. 2015, pp. 307–315.

[23] Y. Liao, W. Du, P. Geurts, and G. Leduc, "DMFSGD: a decentralized matrix factorization algorithm for network distance prediction," *IEEE/ACM Transactions on Networking*, vol. 21, no. 5, pp. 1511–1524, Oct. 2013. [Online]. Available: https://doi.org/10.1109/TNET.2012.2228881

[24] L. Askari, F. Musumeci, and M. Tornatore, "A techno-economic evaluation of vnf placement strategies in optical metro networks," in *2019 4th International Conference on Computing, Communications and Security (ICCCS)*, 2019, pp. 1–8.

[25] B. Ramamurthy, R. K. Sinha, and K. K. Ramakrishnan, "Balancing Cost and Reliability in the Design of Internet Protocol Backbone Using Agile Optical Networking," *IEEE Transactions on Reliability*, vol. 63, no. 2, pp. 427–442, 2014.

[26] M. Alhowaidi, P. Yi, and B. Ramamurthy, "Dynamic provisioning in virtualized Cloud infrastructure in IP/MPLS-over-WDM networks," in *2017 International Conference on Computing, Networking and Communications (ICNC)*, 2017, pp. 83–87.

[27] R. Huelsermann, M. Gunkel, C. Meusburger, and D. A. Schupke, "Cost modeling and evaluation of capital expenditures in optical multilayer networks," *Journal of Optical Networking*, vol. 7, no. 9, pp. 814–833, 2008.