DBvLEA: A Demand-Based Approach to Virtual Link Mapping for Multi-Service Industrial Applications

Frimpong Ansah*, Hermann de Meer†,
*Corporate Technology, Research in Digitalization and Automation, Siemens AG, Munich, Germany
*†Chair of Computer Networks and Computer Communications, University of Passau, Passau, Germany
Emails: *ansah.frimpong@siemens.com, †hermann.demeer@uni-passau.de,

***Abstract*—Network virtualization is proposed in several research work as a means to overcome the ossification of the Internet. Its application relies on embedding algorithms to instantiate virtual networks on substrate infrastructures. Notably, those considered in the scope of traffic-engineering are developed to focus on efficient resource utilization with the aim of increasing the acceptance ratio of the algorithms. In this paper, a demand-based virtual link embedding approach for multi-service mapping in programmable industrial networks is proposed. The approach aims at increasing the overall acceptance ratio of virtual link embedding algorithms by increasing the acceptance of demand critical requests. The goal is achieved by minimizing the deviation between requested demands and the resources satisfying the demand. The approach, when analyzed against state-of-the-art shortest path approaches under the same simulation conditions, shows good results in terms of utilization of the network resources, acceptance of delay-critical traffic demands and overall acceptance ratio.**

***Index Terms*—Software Defined Networks, Traffic Engineering, Resource Orchestration, QoS, Virtual Networks, Network Slicing, Virtual Link Mapping Algorithms.**

I. Introduction

Virtualization enables the coexistence of multiple tenants expressed as Virtual Networks (VNs) on a shared infrastructure [1]–[3]. It is proposed as a means to overcome the Internet impasse [4] as well as evaluate new and existing protocols, and services .

Virtual networks are logical abstract representation of services which usually consist of key service node points and their relations [3]. The service node points are represented as virtual nodes while virtual links represent their relations. In network virtualization, virtual nodes consume resources such as memory and processing power of substrate nodes. Likewise, virtual links consume resources (e.g. bandwidth and switching capacity) of the substrate link they traverse.

Mapping virtual resource demands onto substrate networks is a nontrivial resource allocation challenge usually referred to as the Virtual Network Embedding (VNE) problem [5]. VNE is a process of creating and allocating Virtual Networks (VNs) on shared infrastructures such that the resource constraints of the VNs expressed as demands are guaranteed on the substrate network.

The VNE problem can be split into two sub-problems namely: Virtual Node Mapping (VNoM) which deals with mapping virtual node requirements on substrate nodes, and Virtual Link Mapping (VLM) which deals with mapping virtual link requirements on substrate links within paths connecting the corresponding substrate nodes where virtual nodes are mapped. The VLM problem presents an interesting research use-case that has found application in Traffic Engineering (TE) contemporary with the concept of network and service slicing [6]–[9]. The research direction has gained significant attention due to the advent of programmable network concepts such as Software Defined Network (SDN) which enables the creation of multi-tenant service networks [10].

Often in the VLM research domain, algorithms are designed with the primary goal of embedding more virtual links by minimizing the substrate link resource (mainly bandwidth) utilization. The algorithms that show higher acceptance ratio are usually integer-linear-programming-optimization based [6]–[9], [11], while others are based on heuristics. The inherent design of the algorithms (both heuristic and optimization (Mixed Integer programming (MIP)) result in solutions that seek to find shortest or best paths first for requested demands that are delivered early in the mapping process. We refer to this kind of design goal as Shortest Path First Selection Approach (SPFSA) algorithms. This is because the inherent path selection decision is based on Shortest Path First (SPF).

The SPFSA algorithms when applied to use-cases that require honoring a similar set of requirements as in [6], [7] (e.g. only bandwidth), show good performance gains in terms of acceptance ratio and substrate link utilization. However, when applied to use-cases that require honoring heterogeneous Quality of Service (QoS) requirements (e.g. bandwidth, delay, etc.) [12], their performance decrease significantly [8]. This is mostly due to the observation that critical requirements are penalized more than less critical requirements [8], [11].

In the context of VLM, SPFSA often turns out to be an inadequate solution, especially when considering the per class and overall acceptance ratio of the algorithms in the case of multi-service demands where QoS requirements impose very different resource allocation constraints.

Contrary to SPFSA used in state of the art, we propose VLM algorithm centered on a Demand Based Approach (DBA). The goal of DBA algorithm is to provide embedding solutions for different services with emphasis on how the demands of

978-3-903176-24-9 © 2019 IFIP

the services compare to the solutions obtained as a result of the real-time network resource conditions by minimizing the deviation between the demands and solutions.

The remainder of the paper is organized as follows; Section II gives an overview of the problem and related work, with focus on the path selection design objectives of VLM algorithms while highlighting the challenges in state-of-the-art SPFSA based VLM design objectives together with some mitigating approaches used to increase acceptance ratio of VLM algorithms. Sections III presents the problem formulation followed by an introduction to the DBA for multi-service VLM in Section IV. Section V describes the simulation and results from the analysis. Finally, Section VI concludes the paper and sheds light on further research directions.

II. Problem Overview and related work

Efficient utilization of substrate network resources in the VNE problem is dependent on effective techniques for virtual network embedding, which maps virtual networks on substrate network resources [13]. This therefore makes path selection in the VLM problem a vital step. In state of the art, though often not very obvious, SPFSA is a commonly used approach upon critical analysis of the objective functions of the algorithms. The subsequent sections examine SPF selection approach used in related work.

Shortest Path First Selection Approach (SPFSA)

SPF is most often perceived as the optimal path selection approach in many VLM algorithms. Some algorithms that implement constraint based routing with respect to different constraints [14], [15] use the same path selection methods from list of available substrate paths.

It offers two main advantages: fast processing time of requests and reduced bandwidth utilization across a network [6]–[8], [16], [17]. However, the reduced resource utilization is only effective along a path. Thereby, making it efficient for VNE problems in the sub-graph isomorphic space [18], [19]. This is because in this group of use-cases, the node mappings are intentionally designed to be ambiguous so that virtual nodes are mapped to substrate nodes based solely on the availability of resources (CPU and memory). This is contrary to most TE use-cases that use VLM where the substrate nodes selected for mapping VNs are unambiguously defined [6]–[8], [11]. In this group, bandwidth demand must be guaranteed along every substrate link within a selected path. Thus, optimal bandwidth minimization is achieved when a path contains the least links or traverses less hops. This is an observation with all SPF algorithms used in VLM. Similar observation can be seen with critical analysis of the design objectives in state-of-the-art optimization algorithms such as those based on Mixed Integer Programming (MIP) ([6]–[8], [16], [17]). Thus, they are designed to increase the overall acceptance ratio by minimizing bandwidth utilization. The bandwidth minimization also implies selecting paths with least links (hops) ahead of longer paths.

$$\max\{\sum_{r=1}^{R} y^r - \epsilon \cdot \sum_{i=1}^{N}\sum_{j=1}^{N} b_{ij} \cdot x_{ij}\} \tag{1}$$

For example, consider MIP based VLM algorithms proposed in [6]–[8], [11]. The objective takes the form described in Equ. (1). The minimization part ($\sum_{i=1}^{N}\sum_{j=1}^{N} b_{ij} \cdot x_{ij}$) focuses on finding the shortest paths as it inherently adds up bandwidth (b_{ij}) along the path to determine the shortest path. The minimization aspect of the MIP based optimization algorithm is not different from the SPF path selection of its heuristic counterpart described in [14] for the VLM problem. This is because both have an inherent design objective to find the shortest path first for every Virtual Link (VL) request. The SPF path selection design objective whether MIP optimization or heuristic based, still comes at a cost of greater rejections in future requests due to its greedy nature. Particularly, those with stringent requirements when considered in the scope of multi-service VLM as illustrated by Trivisonno et al. in [8], [11].

Challenges with SPFSA for Multi-Service VLM

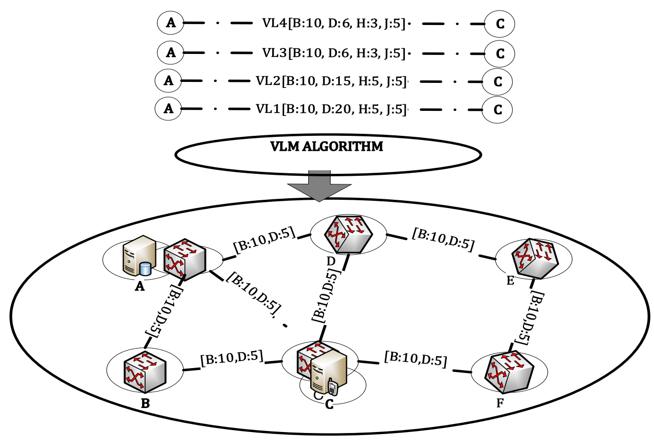

Fig. 1: Example of VL mapping illustrating challenges in state-of-the-art SPFSA and DBA VLM solutions.

The VNE problem is in general *NP-hard*, making it computationally intractable even in the off-line use-case where conditions are more relaxed with respect to the characteristics of the requests. It becomes even more challenging in the on-line use-case where the characteristics of the incoming requests are generally unpredictable [13] and the search space for every request differs due to the different and multiple constraints imposed by the demands of the requests. Thus, results in so many decision variables which makes VLM a sure *NP-complete* problem.

However, On-line VLM approaches are contrived for the TE use-case where foreknowledge of the demands of a VL request and it's arrival order are unknown until delivered for processing [6]–[8], [16]. Thus, in state-of-the-art on-line VLM, mappings are performed one after the other on a first-come-first-processed basis. The paths are computed and the shortest path (highest quality solutions) in terms of hop-count or delay are used for requests that arrive early in the processing pipe. Often, the degree of quality of the solutions used for embedding requests are overly good for the kind of demands requested. This is because the most optimal solutions are greedily chosen for every instance of the demand being processed, hence, leading to the following issues:

- Rejection of future delay critical requests and subsequently lower VL embedding ratios [8];
- Under utilization of the physical infrastructure due to congestion of critical substrate links (delay critical bandwidth) [8], [11];
- High computational time due to the need for re-mapping of requests.

Fig. 1 further illustrates the problem as described above with a simple example.

As shown, the problem requires mapping of VL requests with the demands; *bandwidth (B), delay (D), hop-count (H) and jitter (J)* on the substrate network. The VL requests are processed using an on-line VLM algorithm hosted as a network function in a network controller. The controller provides up-to-date substrate network resource and attributes to enable the embedding functions to compute mapping solutions for VL requests. Each VL request is considered as a service that is configurable on the substrate network after a mapping solution is found.

A SPFSA VLM design objective results in the mapping of `VL1` on substrate link AC and `VL2` on substrate links AB and BC. In the event of `VL3` and `VL4` arriving, there are no suitable resources that can be used to serve them. Therefore, they are rejected despite available bandwidth on links `AD, DC, DE, EF` and `FC`. This results in low substrate bandwidth utilization which subsequently leads to low embedding ratio of requests with tighter demands and the overall embedding ratios of the algorithms.

As a result of the aforementioned challenges, several mitigating approaches have been proposed [20]–[23]. Yu et al. [20] argue on the point that re-optimizing link embedding solutions provides better results and further proves it by proposing VL path splitting and migration as a means to improve the acceptance ratios and substrate utilization. The approach achieves its objective but at additional time overhead for path re-computation. While the concept of path splitting though suitable for data-center networks (packet switched networks), it is not the case for communication services in industrial networks as they require paths traversed by frames to be generally deterministic.

Also, Yao et al. [21] makes a compelling case for the use of reinforcement learning for making decisions on historical data of previously accepted VLs to optimize the acceptance ratios. The approach requires large historical data to reach meaningful decisions thus. makes it suitable for only scenarios with very large historical data. This is because the core principle behind the learning processes require a lot of historical data to determine paths that were inefficiently used for the initial path iteration.

III. Network Model and Problem Formulation

The next sections discuss the substrate and Virtual Network (VN) model for the VLM problem formulation.

A. *Substrate Network Model*

The substrate network is modeled as a non-reflexive directed graph $G^s = (\mathcal{V}^s, \mathcal{E}^s)$, where $\mathcal{V}^s = \{v_1^s, v_2^s, \cdots, v_N^s\}$ is a set of N substrate vertices and $\mathcal{E}^s = \{e_{ij}^s\}$, $i, j = 1, 2, \cdots$, is a set of substrate edges with e_{ij}^s connecting vertex v_i^s to vertex v_j^s.

The substrate vertex v_i^s and edge e_{ij}^s are modeled as entities consisting of a set of resources denoted by $\mathcal{R}_v^s$ and $\mathcal{R}_e^s$ respectively. The set of resources $\mathcal{R}_v^s$ and $\mathcal{R}_e^s$ are associated with a set of attributes denoted by $\mathcal{A}_{vR}^s$ and $\mathcal{A}_{eR}^s$ respectively. The resources of the vertex $\mathcal{R}_v^s$ include storage capacity, number of CPUs etc. Resources of the edge include bandwidth, queue buffer size with attributes such as delay, jitter, etc. In this paper, we focus on the edge resources and attributes. We denote the bandwidth and queue size of an edge e_{ij}^s as $b_{e_{ij}^s}$, and $q_{e_{ij}^s}$ respectively. Also, delay and jitter are denoted by $d_{e_{ij}^s}$ and $j_{e_{ij}^s}$ respectively such that $b_{e_{ij}^s}$, $q_{e_{ij}^s} \in \mathcal{R}_e^s$ and $d_{e_{ij}^s}$, $j_{e_{ij}^s} \in \mathcal{A}_{eR}^s$.

B. *Virtual Network Request Model*

Similarly, the VN is modeled as a non-reflexive directed graph $G^v = (\mathcal{V}^v, \mathcal{L}^v)$, where $\mathcal{V}^v = \{v_1^v, v_2^v, \cdots, v_N^v\}$ is a set of N virtual endpoints which map unambiguously to substrate vertices in the substrate network G^s. $\mathcal{L}^v = \{l_{ij}^v\}$, $i, j = 1, 2, \cdots, N$ is a set of edges such that $\mathcal{L}^v \longmapsto \mathcal{P} \subseteq \mathcal{E}^s$, $\mathcal{P} \neq \mathcal{E}^s$, where l_{ij}^v shows that a connection exist from virtual vertex v_i^v to virtual vertex v_j^v. By the non-reflexive property of G^v, $i \neq j$ (i.e. $i = j = 0$). Also, associated with each VN is a set of demands denoted by $\mathcal{D}^g$, where $\mathcal{D}^g$ represents a set of QoS requirements of the virtual node such as maximum CPU (CPU_{max}), storage (ROM_{max}, RAM_{max}) and virtual edge such as maximum delay (D_{max}), jitter (J_{max}), hop-count (H_{max}) and bandwidth (B_{max}) required by the k^{th} VN such that $ROM_{max}, RAM_{max}, CPU_{max}, D_{max}, J_{max}$, $H_{max}, B_{max} \in \mathcal{D}^g$. The list of mathematical notations for the network model are shown in Table I.

IV. Demand Based Approach (DBA)

The goal of DBA is to embed VL requests representing different communication services such that demands imposed as embedding constraints are strictly adhered while at the same time, increasing the utilization of the substrate network which should lead to subsequent increase in the acceptance ratio of delay critical communication services. DBA algorithm (Demand Based Virtual Linking Embedding Algorithm (DBvLEA)) finds the solution with the minimum deviation to the requested QoS demands among several paths that conform to the demand.

TABLE I: List of mathematical notations for graph models

Graph notation	Description
G^s, G^v	substrate and virtual graph resp.
$\mathcal{V}^s$, $\mathcal{E}^s$	set of physical bridges and links resp.
$\mathcal{V}^v$, $\mathcal{L}^v$	set of VN nodes and links resp.
$B^{l^k_{ij}}_{max}$, $D^{l^k_{ij}}_{max}$,$J^{l^k_{ij}}_{max}$,$H^{l^k_{ij}}_{max}$	max guaranteed bandwidth (burst), delay, jitter and hop-count requirements of the k^{th} VN link between node i and j resp.
CPU^i_{max},ROM^i_{max},RAM^i_{max}	max guaranteed CPU, ROM, RAM requirement of the i^{th} node of the VN
$\mathcal{R}$, $\mathcal{A}_{\mathcal{R}}$	set of physical resources and attributes
$\mathcal{P}$	physical path

Still referring to Fig. 1, upon the arrival of *VL1*, a DBA algorithm computes all possible paths between node A and C that satisfy the hop-count, delay, bandwidth, and jitter demands i.e. ([B: 10, D: 20, H: 5, J: 5]). It then chooses, for example, if delay is the focus of the communication service, the path which minimizes the difference of the actual path delay and the delay demand of *VL1*. At the same time it needs to ensure that the solution adhere to the jitter and hop-count demands. In a situation where jitter is not in focus, it only chooses the path with the minimum delay difference. In the case of *VL1*, path *ADEFC* is chosen. It then applies the same approach upon arrival of *VL2, VL3*, and *VL4*, which results in path ABC and AC being chosen respectively for the VL requests so that at the arrival of *VL4*, path AC can accommodate the request. In this regard, a DBA algorithm achieves 100% acceptance ratio compared to 60% acceptance ratio of an SPFSA algorithm described in Section II.

Designing Objective Function for DBA MIP Algorithms

Following the general methodologies used in state of the art, we model a MIP version of the DBvLEA and heuristic version. We emphasize the use of a combination of heuristic and MIP to reduce the number of decision variables which makes the MIP problem *NP-complete* in other to find solutions in polynomial time. That is, reducing the solution space to only feasible paths before optimizing for the embedding path. This is done by using Yen's or Epstein's loop-free *k-path* algorithms to compute a set of paths that meet the demands imposed as constraints and then perform the minimization of the deviation of the paths that fit the solution space of a VL request. In effect, we reduce the number of linear equations ($\approx N^2 \cdot G$) that is required to solve the problem (to $G.K$), resulting in a time complexity of $O(K(G + (M + NlogN)$ where K is the number of feasible paths, G the total number of VLs, M and N are the total number of links and nodes in the substrate graph respectively.

The general MIP objective function is shown in Equ. (2). It can be observed that contrary to [7], [8], [11], we use the variable d_{ij} which represents the delay in the objective function instead of b_{ij} the capacity of the substrate link. This is because capacity demands impose non-additive constraints and must be fulfilled on every link along a path. It is therefore logical to use an additive constraint such as delay, jitter or hop-count since additive demands impose constraints that are aggregated for the whole path rather than a single link in the path.

$$\max\{\sum_{g=1}^{G} y^g - \{(\sum_{i=0}^{N}\sum_{j=1}^{N} d_{ij} \cdot x_{ij}) - D^g_{max}\} \tag{2}$$

$i = j = 0, i, j = 1, 2, \cdots, N$ where N is the number of nodes, G is the total number of VL requests, y^g and x_{ij} are decisions variables such that:

$$y^g = \begin{cases} 1 & g^{th} \text{ VL is mapped} \\ 0 & \text{otherwise} \end{cases}$$

and

$$x_{ij} = \begin{cases} 1 & y^g \longmapsto e^s_{ij} \\ 0 & \text{otherwise} \end{cases}$$

Based on the reduction technique discussed, Equ. (2) can be reformulated as Equ. (3). Where all paths are in the feasible solution space for the g^{th} VL.

$$\max\{\sum_{g=1}^{G} y^g - \sum_{j=1,\mathcal{P}_j \in \{\mathcal{P}\}}^{K} (d(\mathcal{P}_j) - D^g_{max})\} \tag{3}$$

where $\mathcal{P}$ is the set of paths within the solution space of the g^{th} VL request, K being the total number of candidate paths and $j = 1, 2, \cdots, K$.
The objective function for VL mapping for Equ. (2) and (3) are subject to the following constraints:

$$e^s_{ij} \cdot x_{i,j} \neq 0 \tag{4}$$

$$B^k_{max} \leq b_{e^s_{ij}} \cdot x_{ij} \tag{5}$$

$$\sum_{i=0}^{N}\sum_{j=1}^{N} d_{e^s_{ij}} \cdot x_{ij} \leq D^g_{max} \tag{6}$$

$$\sum_{i=0}^{N}\sum_{j=1}^{N} e^s_{ij} \cdot x_{ij} \leq H^g_{max} \tag{7}$$

$$\sum_{i=0}^{N}\sum_{j=1}^{N} j_{e^s_{ij}} \cdot x_{ij} \leq J^g_{max} \tag{8}$$

$\forall\ e^s_{ij}, d_{ij}, j_{ij} \in E^s, A^s_{e^s_{ij}}$ and J^g_{max}, D^g_{max}, $H^g_{max} \in D^g$.

Equ. (4) is a topology constraint which imposes that a VL request can be mapped to a substrate link only if the substrate link exist. Inequations (5) to (8) are constraints imposed by the QoS demands $(D^g_{max}, H^g_{max}, J^g_{max})$ of the g^{th} VL respectively.

It must be emphasized that the accuracy of solutions obtained is generally dependent on the model of the underlying substrate network. For this reason, dynamic delay estimation based on the trajectory approach [24] is considered though other methodologies such as Network Calculus (NC) [25], [26] exist. Regardless of the methodology used, models that reflect the consequence of device and technology choices on estimating constraints that ensure guarantees in terms of delay bounds (guaranteed maximum e2e delay) are required anyway. The algorithm takes into account the most significant sources of delay and sums them up on the path traversed by a VL. The delay components considered for the simulation which is discussed in Section V are the queuing and transmission delay of substrate links. The queuing delay is considered as the ratio of loads (bits) of all previously embedded VLs and the substrate link speed (bits per second). The transmission delay is considered as the ratio of the load of the g^{th} VL and substrate link speed. Though propagation delay is another component of delay, it is generally not relevant since it is independent of the link load but rather varies with respect to the cable length. Generally in industrial networks, the propagation delays are in the order of nano-seconds due to short cable lengths hence, can be neglected in this kind of analysis [27].

The heuristic version of the DBvLEA algorithm which uses an objective function similar to Equ. (3) is illustrated in Algorithm 1. In line 8, a variant of the Yen's *k-path* algorithm is used to compute all paths that meet the demands of the g^{th} VL. This ensures that all paths are within the solution space of the g^{th} VL request before demand based optimization is performed. Line 10 to 14 computes the optimal demand embedding solution as described in Equ. (3). We refer to the optimal embedding solution as Optimal Demand Path (ODP). Lines 15 to 17 embeds the g^{th} VL request and updates the resources and attributes of the substrate network respectively.

Algorithm 1: DBvLEA:ODP

Input: G^s, G^v
Output: y^g
1 **begin**
2   **Initialization** G^v, G^s
3     $\mathcal{R} \leftarrow$ `setOf(`G^v`);`
4     $G^s \leftarrow$ `getNetwork(`G^s`);`
5   **for** $g \in G^v$ **do**
6     $y^g = 0$;
7     $\delta^g \leftarrow \inf$;
    `// using, Yen's K-path algorithm,`
8     $P \leftarrow$ `computePath(`G^v, *g.src*, *g.dst*, D^g`);`
    `// paths are verified against load, and delay demands of the request.`
9     $ODP \leftarrow 0$;
10   **for** $i \in \mathcal{P}$ **do**
11       $\phi \leftarrow (d(i) - D^g_{max})$;
12   **if** $\phi < \delta^g$ **then**
13         $\delta^g = \phi$;
14         $ODP = i$;
15     $y^g \leftarrow$ `embed(`g, ODP`);`
16   **if** $y^g = 1$ **then**
17   `updateResource(`G^s, D^g`);`
      `// update all resource after` g^{th} `request is embedded`

V. Evaluation and Performance Analysis

The following section describes the simulation platform used to evaluate DBA with state-of-the-art embedding approaches. The simulation results are analyzed, highlighting the differences in results obtained for DBvLEA and that of state of the art when the same conditions are applied.

A. Use-case and Simulation Description

As previous stated, the use-case analysis focuses on a fixed source and destination pair. This is because the node mappings are always unambiguously defined in the VLM TE use-case (see [6]–[9], [11]). The use-case considered for our simulation is similar to the use-cases proposed for SDN-enabled 5G virtual core network in [8] and critical utility networks in [28] with an additional integration of Time Sensitive Networks (TSN) as proposed by 5G-ACIA ([29]) targeted at industrial communication services. Where multiple communication services with very different QoS requirements are provisioned on the same 5G networks for verticals.

Communication between applications such as motion-control systems and Input/Output device (IODs) or actuators can be achieved on the same network together with video surveillance, sensors and alarm monitoring communication services. The use-case considered for the evaluation is depicted in Fig. 2 where the realization of communication between field applications and control applications over a backbone TSN network are examined.
Here, communication services for real-time control traffic such as between `App-A` and `App-B` requiring very stringent end-to-end delay and jitter. Video surveillance and alarm monitoring services, which are also delay as well as bandwidth centric, are realized on the same network.

B. Simulation Environment and Description

The simulation is carried out using the open source ALEVIN [13] on a 64 bit windows machine with 4x2.6GHz processor cores and 16GB RAM capacity. The choice of ALEVIN stems from the possibility of integrating and testing new VNE algorithms against already implemented state-of-the-art algorithms. However, the open-source ALEVIN platform requires some extensions to enable QoS-awareness for

TABLE II: Traffic class and VL QoS Profile

QoS Class	Traffic Class	Cycle (mS)	Bandwidth (bytes)	Delay (μS)	Jitter (μS)	Hop-count
1	Real-time high (RT-H)	250	$64 - 68$	$1000 - 2500$	$\leq \pm 2750$	< 5
2	Real-time low (RT-L)	500	$68 - 128$	$2500 - 5000$	$\leq \pm 5750$	< 8
3	Video surveillance	1000	$512 - 1500$	≥ 50000	$\geq \pm 50000$	> 8

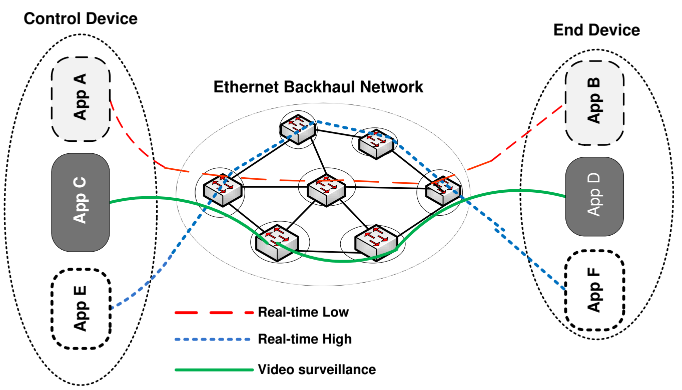

Fig. 2: Multi-service communication over backbone Ethernet network

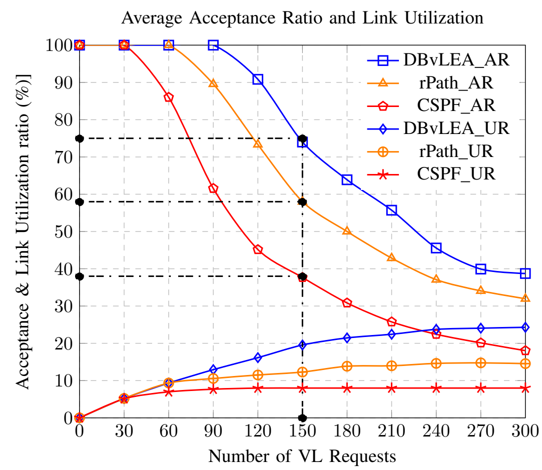

Fig. 3: Average cumulative VL embedding ratio and substrate link utilization for network size of 22 nodes

the TE use-case. This is because the current open-source version does not support QoS-aware VLM. The simulation requires definition of resource-demand pairs, where resource describes the substrate network resources and related attributes as discussed in section III. The demands represent QoS requirements of the VLs.

For the use-case examined, three traffic classes requiring very different QoS demands on bandwidth, delay, jitter and hop-count are considered. Two *real-time* traffic classes representing QoS class 1 and QoS class 2 for high and low cyclic (low and high transmitting) applications. For example sensor/alarm monitoring applications. The third traffic class is for *video surveillance or monitoring* designated for QoS class 3. For each traffic class, minimum and maximum demands for the load (bandwidth), delay, jitter and path length are defined. VLs are generated to represent traffic emanating from each traffic class. For each QoS class representing a traffic class, a set of demands are defined by a random variable uniformly generated between the range of minimum and maximum demand limits for bandwidth, delay, jitter and hop-count requirements per traffic class. These are shown in Table II.

The substrate topology is generated randomly with an average node degree set to three with node connectivity probability set to 0.5 for core nodes. Two edge nodes are defined and connected at two or more core nodes with the highest node degree in the network. This is done to ensure that the bottleneck in the network occurs within the core of the network instead of the points at which the control and field device connect to the network.

Contrary to [8], [11], we consider homogeneous link characteristics between nodes. A Gigabit Ethernet link is used between all core nodes and assume the propagation delay to be negligible. This is because the propagation delay is independent of the load on the link but rather varies with respect to the length of the cable, as the cable lengths do not vary, it makes no difference. It therefor focuses the simulation on the effect of transmission delay and queuing delay contributing to the overall end-to-end path delay fluctuations due to admission of VLs into the different traffic classes on the network. Regardless, the propagation delays in most industrial networks are in orders of nano seconds and therefore have no effect on the result.

The delay per link is calculated as the sum of queuing and transmission delay occurring on each link due to VLs using the link. The transmission delay is defined as the time taken to transmit the load (capacity) of a VL yet to be embedded on a substrate link. The queuing delay is the time taken to transmit the load of already embedded VLs on a substrate link. At any point of time, the delay on the link is equal to the queuing delay since the transmission delay of previously mapped requests are inherent the queuing delay. The queuing delay of a link has a default value set to zero if no request has been embedded on the link. The sum of the two delay components on all links traversed by a VL constitutes the end-to-end delay experienced by a VL on a substrate path. Also, we define a traffic-mix ratio, which determines the percentage

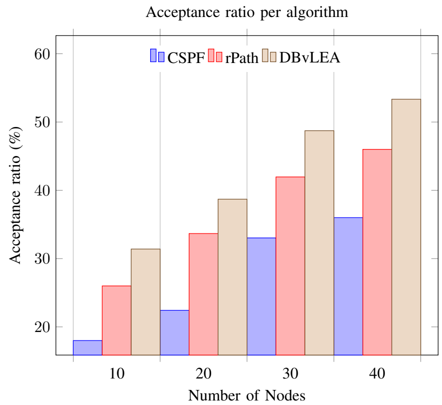

Fig. 4: the acceptance ratio per network size for 300 VLs request

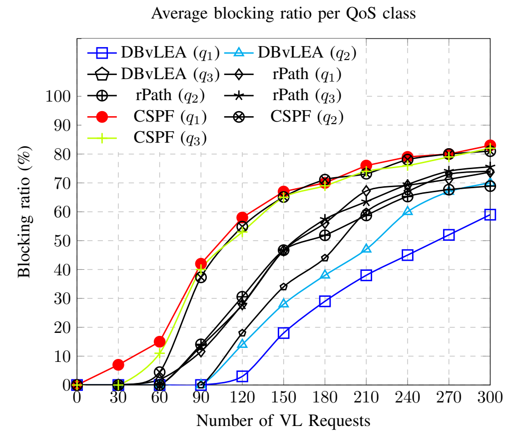

Fig. 5: Average rejection/blocking ratio per QoS class

of the total VLs that are generated per QoS class by the virtual network generator of ALEVIN. For our simulation purpose the traffic-mix ratio is set to 35% for QoS class 1 and 2 each, 30% for QoS class 3. The traffic mix ratio of QoS class 1 and 2 is intentionally chosen to be a bit more than QoS classes 3 to compensate for the low bandwidth demand of the VLs in the class. This therefore allows the analysis to focus more on the additive constraints such as delay rather than bandwidth. It must be emphasized that the simulation considers an on-line VLM, therefore, all algorithms receive the VLs in the same order. Hence, the traffic-ratio of the Qos class 1 and 2 does not affect the efficacy of the process as all algorithms receive the VL requests in the same order.

C. Simulation Results

A total of 10 simulation runs are taken per network size and the average results are evaluated at a confidence level of 95%. The goal of the analysis is to compare SPFSA and DBA based embedding approaches on the following metrics as usually done in state of the art [6]–[9], [11]:

1. Overall acceptance ratio: is the ratio of successfully embedded VLs and the total number of requests that arrived during the simulation.
2. Efficiency in utilization of the substrate links: is a measure of the ratio of load accepted and the overall bandwidth (and effective bandwidth) within the network.
3. Rejection or Blocking ratio per QoS class: measures the number of VLs per QoS class that were rejected, expressed as a ratio of the number VLs that arrived per QoS class during the simulation.
4. Convergence time: is the time required for an algorithm to process a VL.

The DBvLEA is compared to a constrained SPF VLM algorithm which epitomizes the SPFSA design goal used in state of the art. It must be emphasized that SPFSA MIP versions used in [7], [8] from the analysis presented in Section II produce the same results as CSPF algorithms. As rightly noted by Trivisonno et al. in [7], [8], the gains in the case of their MIP version are as a result of the bulk processing of VLs which may lead to situations where more VLs belonging to a critical demand class may exist in a bulk processing.

To show this effect, we include a version of the VLM algorithm where an embedding solution is selected randomly from the feasible set of paths computed by the Yen's *k-paths* in Algorithm 1 line 8 tagged as rPath.

Fig. 3 shows the results for the overall average acceptance ratio and link utilization for a network size of 22 nodes and 300 VLs over 10 simulation runs with a confidence interval of (2.26, 2.24), (2.32, 2.25), (2.21, 2.16) for DBvLEA, rPath and CSPF respectively. While Fig. 4 shows similar results for increasing network size plotted as a histogram.

Referring to Fig. 3, it can be observed that due a high performance substrate network conditions, all algorithms show good acceptance ratio in the beginning of the embedding process but gradually dwindles as the substrate network begins to saturate. This is made evident when we consider the acceptance ratio midway at which point the 150^{th} VL has arrived for processing for all algorithms. It can be observed that CSPF acceptance ratio declines sharply compared to that of DBvLEA, with both rPath and DBvLEA showing approximately 20% and 36% increment in acceptance respectively compared to CSPF. Also comparing DBvLEA which embeds request using an ODP as to random path selection in rPath, significant gain in acceptance of about 16% is observed midway through the embedding process. The reason for such gains in the acceptance ratio is further explained by examining Fig. 5 which shows the VL rejection ratio per QoS class for each algorithm.

From the figure, it can be observed that QoS class 1 experiences less blocking compared to the other QoS class for DBvLEA. With QoS class 3 being the most rejected QoS class due its bandwidth requirements. However, compared to rPath and CSPF though not so significant, it can be observed that QoS class 1 and 2 are the most rejected QoS classes for rPath and CSPF. One possibility for such results can be explained as; CSPF and rPath reach early saturation for certain QoS classes as resources capable of embedding VLs from QoS class 1 and

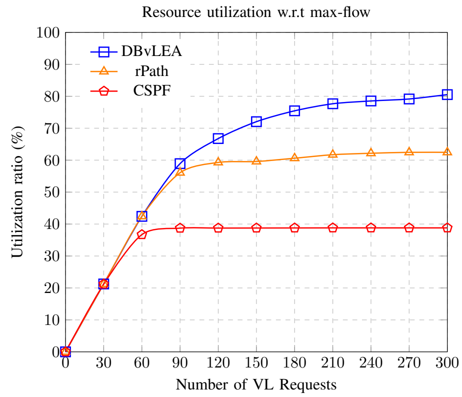

Fig. 6: Average link utilization with rep. to the maximum VLs allowed in the substrate network

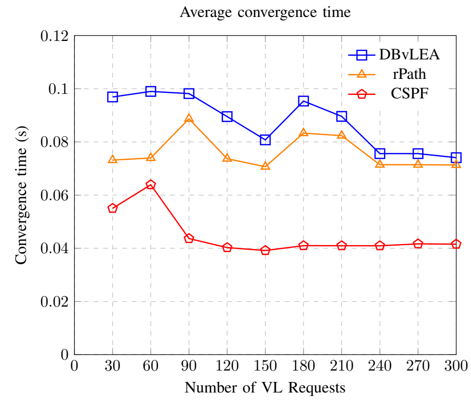

Fig. 7: Average computation time per algorithm

2 are used up in the early stages by QoS class 3 VLs thus, leads to high blocking ratio. This is confirmed by examining how the substrate links are used by each of the algorithms in Fig. 3. It can be observed that CSPF reach early saturation consequently resulting in lower acceptance ratio likewise rPath.

Furthermore, very low substrate utilization is observed by close examination of Fig. 3. The reason for the low utilization as shown in Fig. 3 is solely due to the fact that the link utilization metric is expressed as a ratio of total accepted capacity and the total substrate capacity as presented in [7], [8]. However when considering VLM problem in the TE context with unambiguously mapped source and destination, some of the substrate links automatically fall out of the solution space thus, may never be used but still considered in calculating the utilization ratio.

To better evaluate the link utilization metric, we propose *max-flow* utilization metric. The metric defines the link utilization ratio as a ratio of the accepted capacity and the maximum flow capacity that is required to separate the unambiguously mapped source and destination into two disjoint graphs. This defines the theoretical maximum capacity that can be attained by an ideal VLM algorithm. This new metric is shown in Fig. 6 where the cumulative capacity of VLs accepted into the network is expressed as a ratio of the *max-flow* capacity within the network. From the figure, we clearly see how the different algorithms use the substrate resources thus contributing to their global acceptance ratios.

Also, a critical examination of Fig. 6 shows a sharp rise in capacity utilization of all algorithms at the beginning of simulation process but begin to decline slowly as the process reaches mid-point. It can be seen that at the arrival of the 100^{th} VL, the CSPF can barely accept additional VLs as it reaches saturation due to the SPFSA, leading to sub-optimal utilization. However, DBvLEA continues to accept more VL due to its purposeful selection of paths. This as well confirms the efficacy of DBvLEA to the DBA. rPath results can be attributed to the fact that some randomly selected embedding paths from the feasible paths may sometimes coincide with similar solutions as the *DBvLEA* which is also the case when MIP versions embed VLs in bulk. The overall acceptance ratio for the different path selection approaches are shown in Fig. 4, where we observe a general increase in the overall acceptance ratio of the algorithms with increasing network size. This however can only be attributed the fact that additional substrate nodes and links only enriches the solution space thus, more VLs are accepted on the substrate network.
Upon further examination of Fig. 6, it can be seen that even at the arrival of the 150^{th} VL, DBvLEA shows about 11% and 32% utilization gains over rPath and CSPF respectively while rPath also shows a gain of about 20% over CSPF. This further highlights the efficiency of the DBA in the multi-service VLM use-case.

Finally, we compare the algorithms in terms of convergence time. Fig. 7 shows the average convergence time for embedding a single VL considering all the constraints involved. It can be seen that CSPF shows lower convergence time than the DBvLEA and Random Path Selection (rPath) which is expected given that the DBvLEA and rPath deal with finding several paths from which one is chosen as a solution while the CSPF only computes a single shortest path for every VL. It can also be seen that the convergence time gap between DBvLEA and rPath algorithms reduces gradually to almost the same times as the embedding process approaches completion. This is because the number of candidate paths that has to be traversed to find the ODP in the beginning decreases as the substrate network approaches saturation. This explains the closeness of the convergence time as the embedding process approaches completion. Regardless, the convergence time for DBvLEA is well below acceptable ranges compared to those in state of the art [8], [11]

VI. Conclusion and Future Work

In this paper, challenges for QoS-aware on-line VLM for the TE use-case are discussed. The deficiency of state-of-the-art approaches that adopt a SPFSA for the multi-service QoS-aware on-line VLM problem were highlighted. A demand-based approach to path selection was proposed. Subsequently a MIP and heuristic algorithm based on the approach are defined, discussed and evaluated against SPFSA based algorithm by way of simulations to ascertain the efficacy and efficiency of the different approaches.

The results show that DBA ameliorates the problem of low acceptance ratio of delay critical VLs, the overall acceptance ratio and substrate under utilization while ensuring adherence to QoS demands when compared to state-of-the-art virtual link mapping approaches that adopt SPFSA.

Also, the impact of the various QoS demands on the acceptance ratio and computation time were analyzed. The results show computation times well within the ranges of state of the art. To the best of our knowledge, this work is the first to consider such an approach to path selection for the VLM problem.

Acknowledgement

This work is supported by the German Federal Ministry for Education and Research (BMBF) within the project Future Industrial Network Architecture (FIND).

References

[1] N. M. K. Chowdhury and R. Boutaba, "A survey of network virtualization," *Comput. Netw.*, vol. 54, no. 5, pp. 862–876, Apr. 2010. [Online]. Available: http://dx.doi.org/10.1016/j.comnet.2009.10.017

[2] A. Berl, R. Weidlich, M. Schrank, H. Hlavacs, and H. de Meer, "Network virtualization in future home environments," in *Integrated Management of Systems, Services, Processes and People in IT*, C. Bartolini and L. P. Gaspary, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2009, pp. 177–190.

[3] "Recommendation ITU-T Y.3011, F*ramework of network virtualization for future networks*," ITU, 2012. [Online]. Available: https://www.itu.int/rec/T-REC-Y.3011/en

[4] T. Anderson, L. Peterson, S. Shenker, and J. Turner, "Overcoming the internet impasse through virtualization," *Computer*, vol. 38, no. 4, pp. 34–41, Apr. 2005. [Online]. Available: http://dx.doi.org/10.1109/MC.2005.136

[5] A. Haider, R. Potter, A. Nakao, and N. nakao, "Challenges in resource allocation in network virtualization," 2009.

[6] Z. Despotovic, A. Hecker, A. N. Malik, R. Guerzoni, I. Vaishnavi, R. Trivisonno, and S. A. Beker, "Vnetmapper: A fast and scalable approach to virtual networks embedding," in *2014 23rd International Conference on Computer Communication and Networks (ICCCN)*, Aug 2014, pp. 1–6.

[7] R. Trivisonno, I. Vaishnavi, R. Guerzoni, Z. Despotovic, A. Hecker, S. Beker, and D. Soldani, "Virtual links mapping in future sdn-enabled networks," in *2013 IEEE SDN for Future Networks and Services (SDN4FNS)*, Nov 2013, pp. 1–5.

[8] R. Guerzoni, I. Vaishnavi, A. Frimpong, and R. Trivisonno, "Virtual link mapping for delay critical services in sdn-enabled 5g networks," in *Proceedings of the 2015 1st IEEE Conference on Network Softwarization (NetSoft)*, April 2015, pp. 1–9.

[9] T. Trinh, H. Esaki, and C. Aswakul, "Quality of service using careful overbooking for optimal virtual network resource allocation," in *The 8th Electrical Engineering/ Electronics, Computer, Telecommunications and Information Technology (ECTI) Association of Thailand - Conference 2011*, May 2011, pp. 296–299.

[10] N. McKeown, T. Anderson, H. Balakrishnan, G. M. Parulkar, L. L. Peterson, J. Rexford, S. Shenker, and J. S. Turner, "Openflow: enabling innovation in campus networks," *Computer Communication Review*, vol. 38, pp. 69–74, 2008.

[11] R. Trivisonno, R. Guerzoni, I. Vaishnavi, and A. Frimpong, "Network resource management and qos in SDN-enabled 5G systems," *2015 IEEE Global Communications Conference (GLOBECOM)*, pp. 1–7, 2014.

[12] A. E. Kalr, R. Guillaume, J. J. Nielsen, A. Mueller, and P. Popovski, "Network slicing in industry 4.0 applications: Abstraction methods and end-to-end analysis," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 12, pp. 5419–5427, Dec 2018.

[13] A. Fischer, J. F. Botero, M. Duelli, D. Schlosser, X. Hesselbach, and H. De Meer, "ALEVIN - a framework to develop, compare, and analyze virtual network embedding algorithms," *Electronic Communications of the EASST*, vol. 37, pp. 1–12, 2011. [Online]. Available: http://www.net.fim.uni-passau.de/pdf/Fischer2011a.pdf

[14] T. Korkmaz and M. Krunz, "Multi-constrained optimal path selection," in *Proceedings IEEE INFOCOM 2001. Conference on Computer Communications. Twentieth Annual Joint Conference of the IEEE Computer and Communications Society (Cat. No.01CH37213)*, vol. 2, 2001, pp. 834–843 vol.2.

[15] Y. Yang, L. Zhang, J. K. Muppala, and S. T. Chanson, "Bandwidth-delay constrained routing algorithms," *Comput. Netw.*, vol. 42, no. 4, pp. 503–520, Jul. 2003. [Online]. Available: http://dx.doi.org/10.1016/S1389-1286(03)00199-3

[16] M. Capelle, S. Abdellatif, M. J. Huguet, and P. Berthou, "Online virtual links resource allocation in software-defined networks," in *2015 IFIP Networking Conference (IFIP Networking)*, May 2015, pp. 1–9.

[17] L. Gong, Y. Wen, Z. Zhu, and T. Lee, "Toward profit-seeking virtual network embedding algorithm via global resource capacity," in *IEEE INFOCOM 2014 - IEEE Conference on Computer Communications*, April 2014, pp. 1–9.

[18] M. Chowdhury, M. R. Rahman, and R. Boutaba, "Vineyard: Virtual network embedding algorithms with coordinated node and link mapping," *IEEE/ACM Transactions on Networking*, vol. 20, no. 1, pp. 206–219, Feb 2012.

[19] J. Lischka and H. Karl, "A virtual network mapping algorithm based on subgraph isomorphism detection," in *Proceedings of the 1st ACM Workshop on Virtualized Infrastructure Systems and Architectures*, ser. VISA '09. New York, NY, USA: ACM, 2009, pp. 81–88. [Online]. Available: http://doi.acm.org/10.1145/1592648.1592662

[20] M. Yu, Y. Yi, J. Rexford, and M. Chiang, "Rethinking virtual network embedding: Substrate support for path splitting and migration," *SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 2, pp. 17–29, Mar. 2008. [Online]. Available: http://doi.acm.org/10.1145/1355734.1355737

[21] H. Yao, X. Chen, M. Li, P. Zhang, and L. Wang, "A novel reinforcement learning algorithm for virtual network embedding," *Neurocomputing*, vol. 284, pp. 1 – 9, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0925231218300420

[22] A. Blenk, P. Kalmbach, P. van der Smagt, and W. Kellerer, "Boost online virtual network embedding: Using neural networks for admission control," in *2016 12th International Conference on Network and Service Management (CNSM)*, Oct 2016, pp. 10–18.

[23] M. He, L. Zhuang, S. Tian, G. Wang, and K. Zhang, "Multi-objective virtual network embedding algorithm based on q-learning and curiosity-driven," *EURASIP Journal on Wireless Communications and Networking*, vol. 2018, no. 1, p. 150, Jun 2018. [Online]. Available: https://doi.org/10.1186/s13638-018-1170-x

[24] J. Diemer, D. Thiele, and R. Ernst, "Formal worst-case timing analysis of ethernet topologies with strict-priority and AVB switching," in *7th IEEE International Symposium on Industrial Embedded Systems (SIES'12)*, June 2012, pp. 1–10.

[25] J. W. Guck and W. Kellerer, "Achieving end-to-end real-time quality of service with software defined networking," in *2014 IEEE 3rd International Conference on Cloud Networking (CloudNet)*, Oct 2014, pp. 70–76.

[26] A. Van Bemten and W. Kellerer, "Network calculus: A comprehensive guide," Lehrstuhl fr Kommunikationsnetze, Tech. Rep., 2016.

[27] M. T. Beck and C. Linnhoff-Popien, "On delay-aware embedding of virtual networks," in *The sixth international conference on advances in future internet, AFIN*. Citeseer, 2014.

[28] E. Sakic, V. Kulkarni, V. Theodorou, A. Matsiuk, S. Kuenzer, N. E. Petroulakis, and K. Fysarakis, "Virtuwind - an sdn- and nfv-based architecture for softwarized industrial networks," in *GI/ITG Measurement, Modelling and Evaluation of Computing Systems (MMB) 2018*, Erlangen, Germany, Feb 2018.

[29] 5G-ACIA, "5G for connected industries and automation," 2018, zVEI - German Electrical and Electronic Manufacturers' Association.