Traffic Pattern based Virtual Network Embedding

Andreas Blenk
Institute for Communication Networks
Technische Universität München
München, Germany
andreas.blenk@tum.de

Wolfgang Kellerer
Institute for Communication Networks
Technische Universität München
München, Germany
wolfgang.kellerer@tum.de

ABSTRACT
Virtual Network Embedding (VNE) algorithms are used to find the best embedding of multiple Virtual Networks (VNs) with respect to a performance metric such as network utilization. However, most algorithms assume that the demands of the VNs are static over time. We focus on VNE algorithms for VNs whose demands are changing according to traffic patterns of their provided services. In order to always guarantee high performance and network resource efficiency, the embedding of the VNs has to change, i.e., the VNs have to be re-embedded. By considering traffic patterns of different services, we propose an algorithm that looks ahead and minimize the impact of reconfigurations that are the result of a re-embedding. First simulation results show the potential of VNE algorithms that consider the behavior of evolving service demands.

Categories and Subject Descriptors
C.2.1 [Computer-communication networks]: Network Architecture and Design

Keywords
Virtual network embedding; traffic pattern; path migration

1. INTRODUCTION

Network Virtualization is a level of indirection that may solve problems such as the inflexibility of inventing new service in the Internet [1,4]. It abstracts physical resources and allows tenants of virtual resources, i.e. Service Providers (SPs), to utilize them according to their demands. To increase utilization, Infrastructure Providers (InPs) have to deploy VNE algorithms to efficiently allocate physical resources according to the demands of VNs. Many approaches to VNE have emerged, such as including QoS-aware solutions, resilient embedding, or maximizing economical profit, which are mostly designed for embedding static virtual net-

work demands [6]. Less attention, however, has been paid to dynamically changing VN demands.

As users do not consume services, such as YouTube, constantly over the day, the aggregated traffic of the used applications may also change over time [7–9,12]. Further, the use of different services may also depend on the schedule of social events. The men’s 100m track race, for instance, caused the largest traffic peak during the Summer Olympics 2012 [14]. This behavior of the users may result in partly predictable traffic-patterns [16].

While running virtual networks to provide access to their services, SPs should consider the time-varying behavior of traffic patterns and update their virtual networks, e.g. the virtual node locations or the demanded bandwidth, to always achieve the best possible service quality. Consequently, InPs need to dynamically update the embedding of the VNs to avoid the waste of resources. However, changing the embedding may lead to costly resource reconfigurations.

Existing work has already considered reconfigurations while solving the VNE problem [2,5,15,17,19–21]. In general, all authors jointly optimize the embedding for load and reconfigurations. Further, they assume VN demands to be fixed. We propose a VNE algorithm based on Mixed Integer Linear Programming (MILP) that looks ahead and finds a balance between an instantaneously optimal embedding and any cost caused by future reconfigurations. The algorithm assumes that the changes of future VN demands are predictable as they may reoccur on a regular basis. First simulation results present the potential of our algorithm to improve the network performance in terms of reconfigurations and network utilization.

2. IMPACT OF RECONFIGURATIONS

Dynamically changing VNs may force InPs to change an existing embedding. In order to deploy a new embedding, InP may have to reconfigure the physical resources, which may lead to migrating virtual links or virtual nodes. As the technology used in different network environments, such as Data Centers (DCs) or Wide Area Networks (WANs), may affect the impact of reconfigurations, VNE algorithms should pay attention to the used virtualization technology.

Figure 1 illustrates an example where a traffic pattern aware algorithm may decrease the amount of link migrations. For the first virtual network configuration, two solutions with the same path lengths are possible. However, solution two, which already considers the upcoming virtual network configuration, may reduce the total amount of link
migrations. The figure shows that the amount of reconfigurations is reduced even for a simple example.

For a virtualized WAN, MPLS could be used to interconnect virtualized nodes [3,10,18]. In MPLS networks, a frequently rerouting of traffic may lead to latency inflation [13]. Considering virtual link migrations, a traffic pattern based solution could avoid such a frequently traffic rerouting.

3. OPTIMIZING VNE FOR MIGRATIONS

We assume a shared WAN where, e.g., MPLS could be used for interconnecting virtual nodes. The problem is to embed multiple VNs, whose link demands may change over time according to predictable patterns, in such a way that a frequently rerouting of the traffic could be avoided.

In a first step, the optimal embedding per time slot is determined with focus on a balanced network load. Balanced network load means to have an equal link utilization for all physical links. The objective is given by

\[
\text{minimize } (\text{load}_{\text{max},t} + \text{load}_{\text{avg},t})
\]

where \(\text{load}_{\text{max},t}\) is the load of the link carrying the maximum load of all physical links and \(\text{load}_{\text{avg},t}\) is the average of all physical links at time \(t\). They are considered simultaneously as minimizing only \(\text{load}_{\text{avg},t}\) would lead to a greedy solution taking only the shortest paths, whereas minimizing only \(\text{load}_{\text{max},t}\) would lead to solutions that arbitrarily take longer over shorter paths. The resulting \(\text{load}_{\text{avg},t}\) and \(\text{load}_{\text{max},t}\) are used as input for a traffic pattern aware embedding extension.

In a second step, the costs for reconfigurations are minimized. The objective is given by

\[
\text{minimize } \sum_{t \in T} \sum \text{Virtual Networks } \text{mig}_t
\]

where \(\text{mig}_t\) is the amount of link migrations for one time slot \(t \in T\). The load of the embedding solutions for each time slot, i.e. the load given by \(\text{rec}_t \cdot \text{load}_{\text{max},t}\) and \(\text{rec}_t \cdot \text{load}_{\text{avg},t}\), must be within a certain range of the load given by the optimal embedding. The following constraints

\[
\text{rec}_t \cdot \text{load}_{\text{max},t} \leq \gamma \cdot \text{load}_{\text{max},t} \ \forall t
\]

\[
\text{rec}_t \cdot \text{load}_{\text{avg},t} \leq \gamma \cdot \text{load}_{\text{avg},t} \ \forall t
\]

are added to restrict the load. The parameter \(\gamma \geq 1\) allows to specify an additional load overhead in order to decrease migrations.

First simulations were conducted for the Poland IP network [11]. In each simulation run, 16 VNs were embedded. The number of virtual nodes randomly varied from 6 to 12.

The nodes’ location was randomly chosen as well. For worst case analysis, traffic demands, i.e. traffic patterns, were randomly generated for all experiments. We compare the results of three VNE approaches: pure load balancing (PLB), consider load and reconfigurations (LR), consider traffic pattern (TP). PLB optimizes the embedding for each time slot \(t\) independently. LR extends PLB by considering the existing embedding (i.e., from previous time-slot) to minimize migrations. TP optimizes the embedding under the assumption of knowing the traffic patterns.

Figure 2 presents the results for \(\gamma = 1\). The CDF for the normalized link load for each time slot \(t\) is the same for all solutions. TP leads to the best solution and LR to the second best solution in terms of migrations. Figure 3 presents the results for \(\gamma = 1.05\). Due to a higher acceptable load, LR decreases migrations compared to the solution where \(\gamma = 1\). TP even decreases migrations to 0. Allowing a higher load additionally decreases the number of migrations whereas the increase in link load is still acceptable.

We can conclude that considering traffic patterns decreases migrations while it does not diminish the link utilization. Consequently, VNE algorithms should consider the trade off between migrations and utilization according to the impact of reconfigurations.

4. CONCLUSION AND FUTURE WORK

In case of changing VN demands, necessary re-embedding of VNs may lead to reconfigurations that may degrade the performance of the physical network. In order to avoid performance degradation in case of changing an existing embedding, VNE algorithms should consider reconfigurations. Our solution is to take predictable traffic patterns into account and to investigate the gain of reducing reconfigurations. First simulation results show that there is huge potential in reducing reconfigurations while not diminishing the network utilization.

For future work, we plan to consider virtual node constraints as well. We also want to integrate uncertainty into our traffic pattern assumption. In case of other technologies, such as SDN, different traffic patterns may demand for different controller placements. As migrating controllers may be a critical task, the trade-off between controller placements and different embeddings needs to be investigated.
5. REFERENCES


