Homa: An Efficient Topology and Route Management Approach in SD-WAN Overlays.

Diman Zad Tootaghaj[†], Faraz Ahmed[†], Puneet Sharma[†], Mihalis Yannakakis[*]
[†]Hewlett Packard Labs (USA), [*]Columbia University (USA)

***Abstract*—This paper presents an efficient topology and route management approach in Software-Defined Wide Area Networks (SD-WAN). Traditional WANs suffer from low utilization and lack of global view of the network. Therefore, during failures, topology/service/traffic changes, or new policy requirements, the system does not always converge to the global optimal state. Using Software Defined Networking architectures in WANs provides the opportunity to design WANs with higher fault tolerance, scalability, and manageability. We exploit the correlation matrix derived from monitoring system between the virtual links to infer the underlying route topology and propose a route update approach that minimizes the total route update cost on all flows. We formulate the problem as an integer linear programming optimization problem and provide a centralized control approach that minimizes the total cost while satisfying the quality of service (QoS) on all flows. Experimental results on real network topologies demonstrate the effectiveness of the proposed approach in terms of disruption cost and average disrupted flows.**

***Index Terms*—Software Defined Networking, Wide Area Networks, Network Reconfiguration, Resiliency**

I. Introduction

Enterprises have been increasingly moving from traditional Multi-protocol Label Switching (MPLS) based Wide Area Network (WAN) solutions to Software Defined WAN (SD-WAN) solutions which are transport agnostic overlays built on top of cheaper best effort Internet connections like DSL, Cable, LTE, etc [1–3]. Primary reasons for this shift from MPLS based WAN to SD-WAN are cost savings and centralized control for simpler yet flexible policy-driven management of the overlay network. It is predicted that by the end of 2019, 30% of enterprises will use SD-WAN to connect their branches [4]. Besides expensive MPLS links, the traditional enterprise WAN solutions suffer from scaling limitations arising from manual configuration of routing policies on all MPLS enabled edge routers located at thousands of branches across the globe [5]. Moreover, SD-WAN is inherently oblivious to the state of its underlay network i.e., the Internet, whose best-effort WAN links suffer from frequent and unpredictable network outages [1]. To mitigate the impact of underlay link failures and performance variations, it is necessary to have SD-WAN topology and route management mechanisms to improve the overall resilience of the SD-WAN infrastructure.

We believe that state of the art SD-WAN technologies are not able to realize full potential due to two reasons: i) limited awareness of the underlay path sharing information, and ii) lack of flexible and rich overlay topologies that can support indirect paths in SD-WAN infrastructure. Most SD-WAN deployments adopt a restricted and simple hub and spoke overlay topology where branches are connected to a hub site [2, 3]. Each branch can reach other branches through the central hub. Hub and spoke topologies are simpler to deploy and manage; they require only one or more overlay link between each branch and the hub, and traffic concentration on the hub sites. However, hub and spoke topologies lack resiliency because the hub becomes a hotspot and a single point of failure. Also, these topologies cannot realize the full capacity of the underlying network. On the other end of the SD-WAN topology spectrum, a full mesh topology connects each branch to all other branches, as a result, any two branches can directly communicate with each other using branch-to-branch tunnel [6]. Although full mesh has high resiliency to link and node failures, it has high deployment and management overheads. When branches are multi-homed, a full mesh topology can have $O(mn^2)$ overlay links between n branches with m different underlay network interfaces. For large enterprises with thousands of branches, it becomes prohibitively expensive to manage such a large number of interconnections.

In practice, SD-WAN products for enterprise networks implement network management solutions that allow enterprises to tailor it to their business intents and traffic demands [2, 3, 7, 8]. These solutions provide a hub and spoke as a building block and give enterprises functionality to further manually configure branches to connect to specific hubs or branches. Despite this flexibility, these solutions have several shortcomings. Enterprises do not have topology information or hop-by-hop performance metrics of the underlying network infrastructure, they only have information about the end-to-end performance between branches. This makes it difficult for the SD-WAN admins to diagnose issues in the underlying network and react to it. Even if issues in the underlay are known, SD-WAN admins have to manually re-configure either their SD-WAN topology or the SD-WAN routing policies to avoid the problematic underlay routes. Besides the proprietary SD-WAN products that address enterprise WANs, Google B4 and Microsoft SWAN are software defined WAN management solutions that address inter-data center WANs [7, 9–11]. The traffic characteristics of enterprise networks are widely different from inter-data center networks [12–15]. Inter-data center traffic traverses through private WAN links and network admins have full visibility of the underlay network. Whereas SD-WAN for enterprises, augments the MPLS underlays with the public Internet. The performance dynamics of the Internet makes the SD-WAN challenging to manage.

In this paper, we propose *HOMA*, an SD-WAN topology and route management approach for enterprise networks. There are two novel aspects of HOMA's algorithm and mechanism design. First, unlike state-of-art underlay agnostic SD-WANs, HOMA is underlay aware. HOMA periodically generates end-to-end probes between nodes in the overlay network to infer link failures and performance degradation in the underlay network. HOMA can also infer if two overlay paths are sharing the same underlay congested path. HOMA leverages this underlay awareness to automatically re-configure SD-WAN topology and the overlay routing policies to meet traffic demands and resiliency requirements. The automatic re-configuration is designed to minimize disruptions to the existing overlay network traffic. Second, HOMA's optimization formulation is able to support rich cost models for the best-effort underlay WAN links. Besides minimizing traffic disruptions, HOMA also minimizes the cost incurred due to the use of underlay networks such as LTE, DSL, Cable, etc. To this end, this paper makes the following main **contributions**:

- We design a network monitoring approach that exploits the end-to-end probing information to infer the underlay network topology. Experimental results on our SD-WAN testbed topology shows that our network monitoring approach provides a good approximation of the underlay topology.
- We model the underlay aware re-configuration of SD-WAN as a minimum cost network update (*Min-Cost*) problem. We formulate the *Min-Cost* problem as an integer linear programming (ILP) problem that minimizes the network update cost while satisfying the QoS constraint and show that it is NP-Hard.
- We propose a randomized greedy algorithm to solve *Min-Cost*. Our algorithm provides an approximation guarantee of $e^{O(\sqrt{(ln|H|lnln|H|)})}$ when the capacity of the links and the congestion threshold are large enough to accommodate all flows.
- We present an extensive evaluation of our approaches based on real network topologies and demonstrate their advantages in terms of network cost, disrupted flows, and satisfied demands, compared to existing update approaches that use the direct overlay links for each flow.

The remainder of this paper is organized as follows. Section II and III discusses the background and motivation. In Section IV, we explain the *Min-Cost* optimization problem and show that it is NP-Hard. Section V describes our algorithms. Section VI shows our evaluation methodology and results. Section VII concludes the paper with a summary.

II. Related Works

SD-WANs are overlay networks built on top of the Internet routing substrate. While such application layer overlays have their own overlay routing mechanisms in place to route packets in the overlay network, the path taken in the physical network is defined by intra-domain and inter-domain routing protocols such as OSPF and BGP respectively. Prior work on such application layer overlays can be divided into three categories.

Resilient Overlays: Overlay networks such as the Resilient Overlay Network (RON), were mainly proposed to enable communicating overlay nodes to re-route traffic through indirect overlay routes when the underlying network has issues [16, 17]. In order to re-route overlay traffic, routing overlays have to choose relay nodes to form an alternate overlay path. Prior work on resilient overlay routing has focused on deploying and selecting relay nodes to increase resiliency [18–20]. In [18], relay nodes are selected based on topology graph properties. In [19], the authors proposed an overlay architecture for managing flows between two overlay nodes. Cost effective relay selection has also been proposed to minimize the costs associated with deploying and maintaining an overlay infrastructure [20]. Although these approaches optimize routing, node placement, and relay selection individually, it is imperative to collectively consider these overlay management challenges. In addition to bringing resiliency to overlays, minimizing service disruptions due to re-routing of overlay traffic is crucial for enterprise networks. In this work, we minimize traffic disruptions and overlay infrastructure costs while providing resiliency to the SD-WAN overlay. Another group of work on resilient overlays has addressed combined topology and route management of L2 overlays [21, 22]. These works propose survivable routing techniques that bring resiliency to L2 overlays which are built on the physical layer. These approaches require complete knowledge of the physical layer topology. In contrast, HOMA is designed for SD-WAN overlays and does not require complete physical layer topology information.

Overlay Topology Design: Prior work on overlay construction has mainly focused on carefully choosing nodes in the underlay topology graph for overlay node placement [23–27]. These approaches first propose methods to identify underlay network topology and then find an optimal set of nodes in the underlay for overlay node placement. However, our problem setting is different in two ways. First, in an enterprise SD-WAN deployment underlay nodes are branch offices and all branch offices act as overlay nodes. Second, we make no assumptions on the underlay network topology i.e., the topology of the Internet. We consider a problem setting where we are given a set of underlay nodes that are connected to each other through some underlay topology, our goal is to design an overlay network such that all nodes are connected through the overlay network. Overlay topology design and re-configuration has been studied in the past [28, 29]. In [28], an overly mesh construction algorithm is proposed to improve the redundancy of the overlay network. In [29], the authors propose dynamic reconfiguration of overlay network topology under changing traffic demands generated by the overlay nodes. The approach minimizes two types of re-configuration costs namely; operational cost and reconfiguration costs. However, the approach does not consider the effect of changes in the underlay network performance on the overlay network. In contrast, HOMA is underlay-aware and monitors end-to-end network performance for inferring issues in the underlay network. HOMA re-configures overlay

network with three objectives; minimizing traffic disruptions, re-configuration costs, and operational costs.

Underlay-overlay interaction: A tangential yet relevant area on overlay routing is the interaction between overlay and underlay networks that has been previously studied in [30–32]. These approaches focus on analyzing the conflicts that arise due to the independent nature of overlay and underlay routing protocols, and propose improvements through cross-layer cooperation. In this work, we acknowledge that cross-layer conflicts create traffic engineering challenges for SD-WAN operators as such conflict may result in overlay performance degradation. Since SD-WAN operators do not have complete visibility and control over the public Internet underlay, we propose to use flexible re-routing and re-configuration through end-to-end performance monitoring. This allows us to re-route overlay traffic without making assumptions about the cause of performance degradation.

III. Motivation

In this section we motivate the need for HOMA through illustrative examples. Typically, SD-WAN deployments adopt three types of overlay topologies; (i) direct links where two branches have a direct overlay link and pairs of branches are connected by carefully planning the overlay network, as a result two branches which do not have a direct overlay link cannot communicate with each other (ii) full mesh topology where each branch has a direct overlay link to all other branches, or (iii) hub and spoke topology. We show that these topologies have shortcoming that can be addressed through efficient route and topology reconfiguration proposed in HOMA. We use four metrics to quantify the shortcomings of the above overlay topologies; (i) **Cost:** is defined as the total cost of all overlay links used to route the current flows. As further discussed in more details in Section IV-C, the cost of each overlay link is defined as a fixed (leased) cost and a variable cost that depends on the bandwidth usage. In our model, we assume that the cost of each overlay link is $10 \times l + l \times f$, where, l is the number of hops the overlay link used in the underlay topology and f is the total amount of flow that goes through that link. (ii) **Average disrupted flows:** shows the average number of existing flows which get disconnected when a failure happens in the underlay network. (iii) **Average path length:** is the average number of hops in the underlay network to route each source-destination pairs. (iv) **Path stretch:** is the average path length for each source-destination pair divided by the minimum path length.

To motivate our minimum cost network management approach, we use the BellCanada network topology as the underlay topology which is further described in Section VI. The underlay network uses the shortest path between two nodes to route the traffic. To compare the three traditional overlay topologies with Homa, we create overlay networks consisting of 6 nodes. For each type of overlay network, we generate traffic flows between 10 random chosen source-destination pairs from the six overlay nodes. Each flow between a pair sends 3 units of demands which is the amount of traffic.

TABLE I: Comparison between current practical SD-WAN route management solutions.

Approach	Cost	Avg disrupted flows	Avg path length	Path stretch
Homa (OPT/Greedy)	1428/1688	4.2/4.5	12.8-13.4	1.37/1.44
H&S	2530	5.8	11.5	1.23
Direct links	2046	5.3	9.3	1
Full mesh	2706	5.3	9.3	1

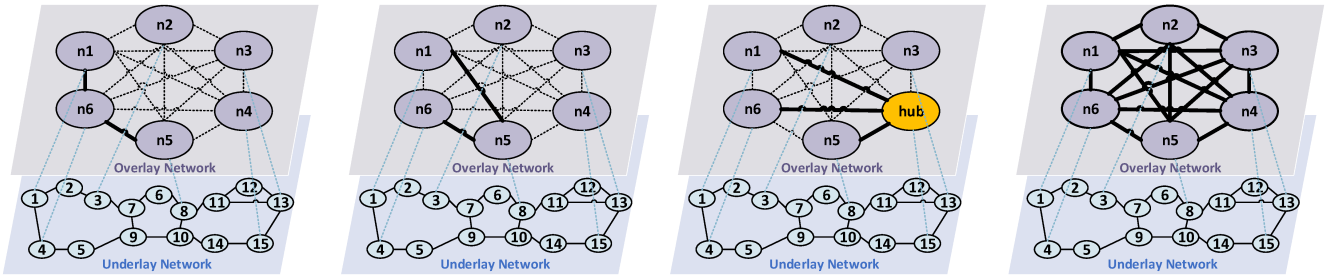
We route all demands in the overlay network, using Homa (OPT/Greedy) which is further explained in Section V, then we route these demands using the three traditional overlay topologies. We first compare the cost of sending the traffic, the average path length for each source-destination pair in the underlay network, and path stretch in each approach. For each type of overlay network, we randomly disrupt 5 links in the underlay network and compare the average disrupted flows. As shown in Table I, Homa has the lowest cost to transfer the traffic and has better failure resiliency compare to other approaches while it has higher average path length. The large gap in the network route management cost between Homa (OPT/Greedy) that uses indirect links and the current SD-WAN solutions motivates this paper.

An illustrative example: To further clarify the discussion, consider the overlay network topology shown in Figure 1 with 6 nodes. For simplicity suppose each overlay node is connected to other nodes using one interface that uses the shortest path in the underlay network topology. We want to find the minimum cost overlay topology to route two flows, $\{f_1 : n1 - n5, f_2 : n6 - n5\}$ with 1 unit of demand each. Table II shows the overlay/underlay links used and the total cost in Homa, direct link, H&S, and Full mesh approaches correspondingly. For the hub and spoke solution, we assume that the hub is placed on $n4$. In this example, Homa tries to find the solution to route all flows while minimizing the cost, and therefore, uses indirect overlay links $n1 - n6 - n5$ to route f_1 and re-uses the overlay link $n6 - n5$ to route f_2. Therefore, the total cost for each overlay link is $10 \times 1 + 1$ for $n1 - n6$ and $10 \times 4 + 2 \times 4$ for $n6 - n5$. However, the direct link approach uses the direct overlay link $n1 - n5$ to route f_1 and $n6 - n5$ to route f_2 with a total cost of $10 \times 5 + 5$ and $10 \times 4 + 4$ respectively. The H&S approach uses the overlay links $\{n1 - hub, hub - n6\}$ for the first flow and $\{n6 - hub, hub - n5\}$ for the second flow and therefore, uses three overlay links $\{n1 - hub, hub - n6, hub - n5\}$ with a total cost of $10 \times 6 + 6$, $10 \times 5 + 5$, and $10 \times 3 + 2 \times 3$ respectively. The full mesh topology leases all overlay links with a total cost of 569 and uses the direct links to route the traffic between each source-destination pair. Therefore, Homa finds a minimum cost overlay network and significantly improves the cost.

IV. Problem Formulation

A. Network Model

We consider the problem of route management with minimum cost that satisfies the QoS constraint during (i) topology

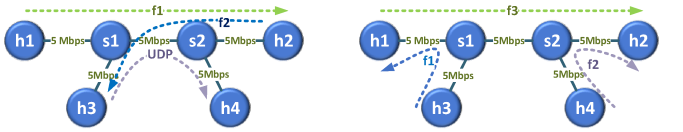


(a) Homa. (b) Direct link. (c) H&S. (d) Full mesh.

Fig. 1: Homa, direct link, H&S, and Full mesh overlay network design.

TABLE II: Comparison between current practical SD-WAN route management solutions in the example of Figure 1.

Approach	Overlay links	Underlay links	Cost
Homa	$\{n1-n6, n6-n5\}$	$\{1-4, 4-5-9-10-8\}$	59
Direct links	$\{n1-n5, n6-n5\}$	$\{1-2-3-7-6-8, 4-5-9-10-8\}$	99
H&S	$\{n1-n4-n5, n6-n4-n5\}$	$\{1-4-5-9-10-14-15\}, \{15-14-10-8\}, \{4-5-9-10-14-15\}, \{15-14-10-8\}$	157
Full mesh	all overlay links	$\{1-2-3-7-6-8, 4-5-9-10-8\}$	569



(a) Direct Correlation test bed. (b) Indirect Correlation test bed.

Fig. 2: Experimental testbeds that show (a) direct correlation, and (b) indirect correlation of flows.

changes, (ii) service/traffic changes, (iii) new policy requirements, and (iv) link/node failures on SD-WANs from a traffic engineering perspective. We are given a capacitated undirected underlay graph $G_u = (V_u, E_u)$, and a capacitated undirected overlay graph $G_o = (V_o, E_o)$ where V_u/V_o represents the underlay/overlay network nodes and E_u/E_o is the set of underlay/overlay communication links connecting them. The underlay network can consist of multiple regions including DSL, MPLS, Wifi, and cable regions. We are also given a set of flow demands H, where each flow h has a source s_h, a destination t_h, and d_h units of demand. Each pair of overlay nodes are connected using different direct interfaces (e.g. DSL, LTE, Cable, and Wireless). Each link $(i,j) \in E_o$ on interface $t \in I$ has a capacity of $c_{ij,t}$. We also assume that the cost of using interface t on link (ij) is equal to $p_{ij,t}$. We want to use a subset of overlay links that can satisfy all demands while minimizing the cost. Table III summarizes the **notations** used in our formulation.

In the following sub-sections, we first define the *correlation matrix* (CM) to infer the underlay topology. Then, we define our minimum cost network reconfiguration problem.

TABLE III: Notations used in our formulations.

Notation	Explanation
$G_u = (V_u, E_u)$	an undirected underlay graph where V_u represents the set of underlay nodes and E_u is the set of underlay links
$G_o = (V_o, E_o)$	an undirected overlay graph where V_o represents the set of overlay nodes and E_o is the set of overlay links
$H = H_{old} \cup H_{new/disrupted}$	the graph of all flow demands $H = (V_h, E_h)$ including existing and new/disrupted flows, where $E_h = \{(s_1, t_1), ..., (s_h, t_h)\}$
$c_{ij,t}$	capacity of each link $(ij) \in E_o$
$c_{i,t}$	capacity of the interface $t \in I$ on node i
$p_{ij,t}$	cost of using link (ij) on interface t in the overlay network.
b_i^h	the amount of flow generated/consumed by node i
$x_{ij,t}^h$	the new routing decision to use link (i,j) for flow h on interface t (when $x_{ij,t}^h = 1$), or not ($x_{ij,t}^h = 0$)
$\lambda_{ij,t}$	the binary variable that specifies if a link is turned on in the new routing for at least one of the flows (when $\lambda_{ij,t} = 1$) or not (when $\lambda_{ij,t} = 0$).
d_h	amount of demand flow for flow h.
D	the largest demand value.

B. Mutual Correlation Function

Network inference approaches that use network monitoring tools to provide the internal network state is crucial for many network management functions such as traffic engineering, anomaly detection, and service provisioning. Especially, when the important network performance metrics are not directly observable (e.g. due to lack of access), network inference approaches provides a solution to infer such metrics. There are many ways to infer network topology. One way is to use network tomography techniques by sending probing traffic periodically and exploiting the end-to-end performance measurements to infer network topologies [33–37]. Compared to other tomography techniques that exploit SNMP polling or traceroute, end-to-end probes do not need any special support from the underlay routers [38–42] and is, therefore, a reliable tool for monitoring the performance of our SD-WAN overlay network.

In this section, we describe a monitoring approach that uses the end-to-end performance metric of the overlay links to infer information about the uncontrollable routing nodes and the underlay network topology [43]. We first measure end-to-end delay for each flow at different time slots. If the

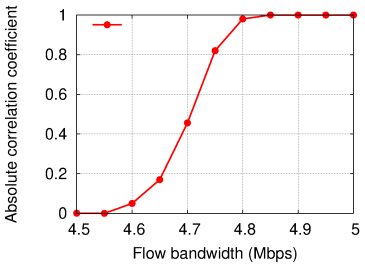
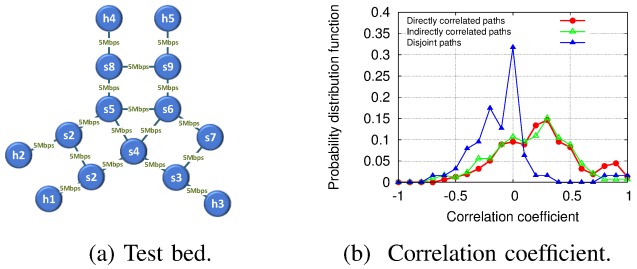


Fig. 3: An experiment that shows the correlation coefficient versus the amount of congestion.



(a) Test bed. (b) Correlation coefficient.

Fig. 4: a) Our network test bed topology, b) Probability distribution function of correlation coefficient between 1) directly correlated flows, 2) indirectly correlated flows, and 3) edge-disjoint flows.

two flows pass through the same congested link/node in the underlay network, we expect to see a correlation between the two sampled signals. On the other hand, if the two flows do not share any congested link/node, we expect little or no correlation between the two sampled signals.

Complete information about the underlay topology: If the underlay network topology was visible to the overlay network, we define the *fraction of overlapping links* (FOL) between any two overlay links as the number of shared links between the two links in the underlay network divided by the total number of underlay links.

$$FOL(e_{ij,t}, e_{nm,t}) = \frac{|l \in e_{ij,t} \cap e_{nm,t} : l \in E_u|}{|E_u|} \quad (1)$$

Inferring the underlay topology: If the underlay network topology was not visible to the overlay network, we send end-to-end ping messages between the two overlay links and use the correlation function between the two measured delay signals $(d_{ij,t}, d_{mn,t'})$ to find the *Correlation Matrix* (CM) as follows:

$$CM(e_{ij,t}, e_{nm,t'}) = corr(d_{ij,t}, d_{mn,t'}) \quad (2)$$

The computed correlation between the two sampled signal $d_{ij,t}$ and $d_{mn,t'}$ is $+1$ in the case of a perfect direct (increasing) linear relationship and -1 in the case of a perfect inverse (decreasing) linear relationship.

Direct Correlation: Figure 3 shows a preliminary experiment that characterizes the direct impact of congested shared links in our SD-WAN test bed which is shown in Figure 2a. We first establish a UDP connection between two hosts $h3$ and $h4$. We then send ICMP packets from $h1$ to $h2$ to create flow $f1$ and from $h2$ to $h3$ to create flow $f2$. The network bandwidth for all links including the shared link between $f1$ and $f2$ is set to 5 Mbps. As we increase the UDP data rate and consequently the amount of congestion on the shared link, the ICMP messages experience more correlation as shown in Figure 3. We observe that when there is little or no congestion on the shared link, the correlation coefficient is usually close to zero, while as we increase the amount of congestion on the shared link, the correlation coefficient gets closer to one.

Indirect Correlation: During our experiments, we also observed a different type of correlation. We observed a linear correlation between two flows which are edge-disjoint but share the same congested links with a third flow. Figure 2b shows our testbed experiment for the indirect correlation. We created three flows $f1 : h3 \rightarrow h1$, $f2 : h4 \rightarrow h2$, and $f3 : h1 \rightarrow h2$. We observed that while $f1$ and $f2$ do not share any links, there exists a high correlation between them since a third flow $f3$ shares a congested link with both flows. We observed that congestion on the shared link between $f1$ and $f3$ also changes the delays on $f2$.

To further characterize the impact of direct and indirect correlation, we conduct three different sets of experiments on a network topology of 6 hosts and 8 switches as shown in Figure 4a. The experiments consist of a total of 620 pairs of flows generated between different hosts in the test bed. The length of each flow is 60 seconds and the bandwidth demand of each flow is equal to the capacity of the links in the network. We first measure the correlation between pairs of flows which are on disjoint paths. We generate a pair of flows on disjoint paths and measure round trip time (RTT) in milliseconds using ICMP ping messages for each flow. While a pair of flows is active no other flow is generated. For each pair, we compute the Pearson's correlation coefficient using the RTT values. In the second set of experiments, we generate pairs of flows such that they have at least one shared link. We measure the correlation between the overlapping pairs of flows using the RTT values. In the third set of experiments, we generate pairs of flows on disjoint paths and generate a third flow which has at least one shared link with the first two flows. We measure the correlation between the pair of flows which are on disjoint paths. Figure 4b shows the distribution of the correlation values for (i) directly correlated paths, (ii) indirectly correlated paths, and (iii) disjoint paths in our experiment. As shown, the correlation coefficient of disjoint paths which do not share any congested link with any other flows, show a lower correlation coefficient and the probability distribution function shows a correlation coefficient around zero for most of the flows. However, the correlation coefficient for directly and indirectly correlated flows have similar probability distribution function and most of the flows have a correlation coefficient higher that 0.3.

C. Minimum Cost Network Reconfiguration

Our goal is to minimize the network reconfiguration cost by selecting a path for each flow $h \in H$ which minimizes the cost of overlay links for all reconfigured flows.

• **Input parameters:** Let b_i^h be the amount of flow h generated by node i which is $b_i^h = d_h$, if i is the source node ($i = s_h$), and $b_i^h = -d_h$, if i is a destination node ($i = d_h$) and $b_i^h = 0$ otherwise. Furthermore, we assume that QoS requirements are expressed in terms of the number of shared congested links if the controller has full observation of the underlay network topology and are expressed in terms of the amount of correlation between two overlay links if the underlay topology is not visible by the controller. Let Ω_h be the maximum amount of shared congested links allowed for flow h or the summation of maximum amount of correlation allowed between all links for flow h.

Prior works on overlay network designs considered two distinct cost models for leasing each overlay link: (i) fixed (leased) cost model, and (ii) variable (usage-based) cost model [44, 45]. However, most SD-WAN pricing models have both a fixed and a variable component that depends on the bandwidth usage. We take this fact into account and assume the price of each link $(i,j) \in E_o$ for interface $t \in I$ is a linear function of the amount of flow passing through that link and a fixed cost for that interface as follows:

$$p_{ij,t} = \alpha_{ij,t} + \zeta_{ij,t} \cdot f_{ij,t}, \quad \forall (i,j) \in E_o, \forall t \in I, \tag{3}$$

where $\alpha_{ij,t}$ is a fixed price of using link (i,j) on interface t, $\zeta_{ij,t}$ is the bandwidth cost rate of link (i,j) on interface t, and $f_{ij,t}$ is the total amount of flow that goes through link (i,j) on interface t, i.e. ($f_{ij,t} = \sum_{h \in H} (x_{ij,t}^h) \cdot d_h$). The fixed price of an overlay link corresponds to the price charged by the ISPs for maintaining each overlay link and the variable (bandwidth cost rate) corresponds to the fee paid to the ISPs for consuming bandwidth to carry traffic on the overlay link [46].

• **Decision variables:** With $x_{ij,t}^h$ we represent the current routing decision to use link (i,j) for flow h on interface t when $x_{ij,t}^h = 1$ or not $x_{ij,t}^h = 0$. Also the binary variable $\lambda_{ij,t}$ represents if a link is turned on in the new routing for at least one of the flows ($\lambda_{ij,t} = 1$) or not ($\lambda_{ij,t} = 0$).

• **Min-Cost network reconfiguration:** We formulate the minimum cost network reconfiguration (*Min-Cost*) optimization problem as follows:

$$\text{minimize} \quad \sum_{t \in I} \sum_{(i,j) \in E_o} \alpha_{ij,t} \lambda_{ij,t} + \sum_{t \in I} \sum_{(i,j) \in E_o} \zeta_{ij,t} \sum_{h \in H} (x_{ij,t}^h + x_{ji,t}^h) \cdot d_h$$

$$\text{subject to} \sum_{h \in H} (x_{ij,t}^h + x_{ji,t}^h) \cdot d_h \leq c_{ij,t}, \quad \forall (i,j) \in E_o, \quad \forall t \in I \tag{4a}$$

$$\sum_{t \in I} \sum_{j \in V_o} x_{ij,t}^h = \sum_{t \in I} \sum_{k \in V_o} x_{ki,t}^h + sign(b_i^h), \quad \forall i \in V_o, \quad \forall h \in H \tag{4b}$$

$$\sum_{h \in H} \sum_{j \in V_o} (x_{ij,t}^h) \cdot d_h \leq c_{i,t}, \quad \forall i \in V_o, \quad \forall t \in I \tag{4c}$$

$$\sum_{h \in H} \sum_{j \in V_o} (x_{ji,t}^h) \cdot d_h \leq c_{i,t}, \quad \forall i \in V_o, \quad \forall t \in I \tag{4d}$$

$$\lambda_{ij,t} \geq \frac{\sum_{h \in H} [x_{ij,t}^h + x_{ji,t}^h]}{|H|} \quad \forall t \in I \ \ \forall (ij) \in E_o \tag{4e}$$

$$\sum_{t \in I} \sum_{t' \in I} \sum_{h' \in H, h' \neq h} \sum_{(m,n) \in E_o} \sum_{(i,j) \in E_o} x_{ij,t}^h \cdot x_{mn,t'}^{h'} \cdot CM((ij,t),(mn,t')) \leq \Omega_h \ \ \forall h \in H \tag{4f}$$

$$x_{ij,t}^h, \lambda_{ij,t} \in \{0,1\}, \ \ \forall (ij) \in E_o, \ \ \forall h \in H \ \ \forall t \in I \tag{4g}$$

The left hand side of the objective function corresponds to the fixed cost of turning each link on and the right hand side of the objective corresponds to the variable bandwidth cost of each link for all flows that pass through it. The first constraint shows the link capacity constraint and the second constraint is the flow balance constraint, i.e. the total flow out of a node is equal to the summation of total flows that comes into a node and the net flow generated/consumed at the node. Constraint 4c and 4d specifies that the total flow going out/in a node for interface t is bounded by the capacity of interface t on node i, i.e. $c_{i,t}$. In our setting we set the capacity of the overlay link ij for interface t to the minimum of $c_{i,t}$ and $c_{j,t}$. In addition to the link capacity and flow balance constraints (4a and 4b), the binary variable $\lambda_{ij,t}$ specifies if a link is being used in the new route for at least one of the flows (when $\lambda_{ij,t} = 1$), or not (when $\lambda_{ij,t} = 0$). Constraint 4e ensures that $\lambda_{ij,t}$ is set to 1 when at least one of the flows use link (i,j) on interface t. Constraint 4f is the QoS constraint and indicates the maximum amount of correlation (Ω_h) between the selected overlay links in the new setting.

In words, formulation (4) aims at minimizing the network reconfiguration cost such that all the flows can be routed subject to link capacity and QoS constraint.

Theorem 1. ***NP-Hardness****: The problem Min-Cost is NP-Hard.*

Assuming that the capacity of edges is large enough to accommodate the sum of all demand flows and that the QoS threshold (Ω_h) is set to infinity; the Min-Cost problem corresponds to the well-studied problem of buy-at-bulk network design [47]. Buy-at-bulk reflects the assumption that the edge cost functions have economies of scale. The buy-at-bulk problem for subadditive cost functions $p_{ij,t}$ has $O(log|H|)^2$ approximation ratio and $\Omega(log^{1/4}(|H|))$ hardness bound, if $p_{ij,t}$ is uniform over all $e = (ij,t)$. Also, Min-Cost has $O(log^3|H|)$ approximation ratio and $\Omega(log^{1/2}|H|)$ hardness bound if $p_{ij,t}$ is different from edge to edge. We focus on non-uniform subadditive cost function defined in Section IV-C.

V. HOMA: Minimum Cost Network Reconfiguration

In this section, we propose *HOMA*, our network topology and route management framework that minimizes the network reconfiguration cost while satisfying the QoS constraints and the three algorithms that solve the Min-Cost problem.

A. Minimum Cost Update Algorithms

The Min-Cost problem is hard to solve optimally and efficiently as shown in Theorem 1. Therefore, in this section, we propose a greedy algorithm that selects the overlay paths

with minimum cost greedily which approximates the optimal solution, in polynomial time.

Before delving into our proposed algorithms, we first describe a linearization approach to make the optimization problem linear to solve it using standard integer programming solvers.

Linearization: The correlation constraint (4f) in Min-Cost is non-linear and is the product of two binary variables $x^h_{ij,t}$, and $x^{h'}_{mn,t'}$.

Therefore, we define a new variable $\Delta^{h,h'}_{(ij,t),(mn,t')} = x^h_{ij,t} \cdot x^{h'}_{mn,t'}$ and add the following three constraints to make the optimization problem linear.

$$\Delta^{h,h'}_{(ij,t),(mn,t')} \leq x^{h'}_{mn,t'}, \quad \forall h \in H, h' \neq h \in H, \forall (ij),(mn) \in E_o, \forall t,t' \in I \tag{5a}$$

$$\Delta^{h,h'}_{(ij,t),(mn,t')} \leq x^{h}_{ij,t}, \quad \forall h \in H, h' \neq h \in H, \forall (ij),(mn) \in E_o, \forall t,t' \in I \tag{5b}$$

$$\Delta^{h,h'}_{(ij,t),(mn,t')} \geq x^{h}_{ij,t} + x^{h'}_{mn,t'} - 1, \quad \forall h \in H, h' \neq h \in H, \tag{5c}$$
$$\forall (ij),(mn) \in E_o, \forall t,t' \in I$$

The first two inequalities ensure that $\Delta^{h,h'}_{(ij,t),(mn,t')}$ will be zero if either $x^{h'}_{mn,t'}$ or $x^h_{ij,t}$ are zero. The last inequality will make sure that $\Delta^{h,h'}_{(ij,t),(mn,t')}$ will take value 1 if both binary variables are set to 1.

Baseline: Traditional SD-WAN with direct links: A naive approach which is used in most SD-WAN topology managements, is to connect each source-destination pair using the direct overlay links that have a lower cost for each flow independent of the other flows. However, as the routing decision for one flow is highly correlated with the routing decision of other flows and on the underlying network topology, this approach does not yield good performance results as we later see in Section VI. Further, in case of a failure in the underlay network, one or more direct links for a source-destination pair, might disrupt. As a result, using indirect working overlay links might recover the system with lower cost in such cases.

Inflated Greedy Algorithm: Inspired by the work of Charikar et al. [48], we propose a randomized greedy algorithm that solves the Min-Cost problem. When there are no capacity and correlation constraints, the algorithm provides an approximation guarantee of $e^{O(\sqrt{(ln|H|lnln|H|)})}$ when the problem has at most $|H|$ source-destination pairs with unit demands and $e^{(O(\sqrt{(ln|H|lnln|H|)}))}logD$ in the case of general demands where D is the largest demand value. Algorithm 1 shows different steps of the greedy algorithm with unit demands for all flows. The algorithm first selects a random permutation of demands (line 1). Without loss of generality, let $d_1, d_2, ..., d_{|H|}$ be the random permutation. We initialize the residual capacity of each link to the the capacity of that links (line 2). We next assign inflated demands to each pair as a function of its position in the permutation (line 4). Then the algorithm tries to find the cheapest way to connect each source-destination pair greedily using the constrained shortest path algorithm (CSP) explained in Algorithm 2 (line 5).

Algorithm 1: Greedy algorithm for *Min-Cost* problem

1 Pick a random permutation of demands (without loss of generality let $d_1, d_2, ..., d_{|H|}$ be the random permutation).
2 Initialize the residual capacity of all links: $c_{\tilde{ij},t} = c_{ij,t}\ \forall (ij) \in E_o, t \in I$
3 **for** $i = 1$ *to* $|H|$ **do**
4 Set $d'_i = \frac{|H|}{i}$.
5 Find the shortest constraint path using CSP algorithm (Algorithm 2) to find the smallest cost of routing d'_h units of demand, using the network constructed for the previous $i-1$ pairs.
6 **if** *the correlation constraint for all existing flows are satisfied by adding the new flow* **then**
7 Route a single unit of demand between s_i and t_i along the path selected at the i-th iteration.
8 Update the residual capacity $c_{\tilde{ij},t}$ of all links along the selected path: $c_{\tilde{ij},t} = c_{\tilde{ij},t} - 1$
9 **else**
10 Drop d_i.

Finally, if the correlation constraint of all existing flows are satisfied, we route a single unit of demand between s_i and t_i along the path selected at each iteration (line 7) and update the residual capacity of the links along that path (line 8). Otherwise, we drop the d_i (line 10).

Similar to [48], for general demands, we partition source-destination pairs into groups $G_0, G_1, ..., G_{logD}$ where all source-destination pairs with demand in the interval $(2^{i-1}, 2^i]$ are in G_i. The algorithm first modifies all demand pairs in group G_i to be 2^i by inflating them by no more than a factor of 2 and then each group is solved independently using Algorithm 1.

Constrained Shortest Path Algorithm: Our inflated greedy algorithm differs from the proposed approach in [48] in that we need to take into account the link capacity and QoS constraints to find the set of shortest paths that solve Min-Cost problem as explained in Section IV-C. Therefore, we propose a pseudopolynomial dynamic programming algorithm that finds the constrained shortest paths in Algorithm 1. Suppose that every edge $(ij,t) \in E_o$ has a positive cost $p_{ij,t}$ and a correlation $CM_{ij,t}$ where the correlation is integer and show the total correlation of that link with respect to all other flows which exists in the network (we assume each link has a direct/indirect correlation with another link or not, i.e $CM_{ij,t}$ shows how many links are directly or indirectly correlated with link (ij,t)). When we have full information about the underlay, we use the number of shared links in the underlay topology to specify the correlation of link (ij,t) with respect to other links which are used in the current solution. In the case, where we don't have full information about the underlay topology, we use the monitoring information and based on the correlation specify whether the link is correlated with another link (when the correlation is higher than a threshold) or not. In the case where the correlations are non-integers (or if they are very large integers), we can compute in polynomial time an approximate shortest path (within any approximation factor $1+\epsilon$) by using scaling and rounding, and applying a

Algorithm 2: Constrained Shortest Path (CSP) algorithm

for *every integer* $k = 1, ..., \Omega_h$ *and every node* $j \in V_o$ **do**
  Compute the minimum cost $C(j,k)$ of a path from s_h to j subject to the correlation being at most k and the link capacity being at least d_h.
  Basis: $C(s_h, k) = 0 \; \forall k$ and $C(j, 0) = \infty \; \forall j$
  Recurrence: **for** $j \neq s$ *and* $k > 0$ **do**
    $C(j,k) = min_{i \in V_o, t \in I}(C(i, k - CM_{ij,t}) + p_{ij,t})$, where if i is not a neighbor off j, or $k - CM_{ij,t} < 0$, or $\tilde{c_{ij,t}} \leq d_h$ then we don't include i in the minimization.

psedupolynomial algorithm [49, 50].

We want to find the minimum cost path from a source node s_h to a destination node t_h under the constraint that the total correlation is at most a given bound Ω_h and the bottleneck capacity of the path is more than the capacity of the link $c_{ij,t}$. For every integer $k = 1, ..., \Omega_h$ (in this order) and every node j, we compute the minimum cost $C(j,k)$ of a path from s_h to j subject to the correlation being at most k and the link capacity being at least d_h, using the recurrence shown in Algorithm 2.

The complexity of this algorithm is $O((|V|+|E|)\Omega_h)$ where $|V|$ and $|E|$ are the number of nodes and edges of the graph.

B. Handling Unexpected Network Events

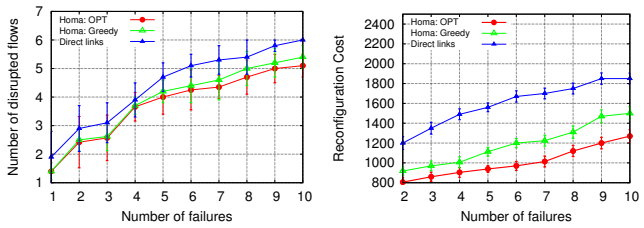
In this section, we explain how Homa reacts to unexpected network events.

1) Underlay Link/Node Failures: Any link/node failure in the underlay network that leads to disconnectivity of the current overlay network are detected and sent to the controller. The controller then removes the direct links in the overlay network which were using the failed link/node in the underlay topology. If one or more flows get disrupted as a result of the failure, the controller re-computes the network state and routing using the updated overlay network. The algorithm minimizes the update cost and provides the QoS for each flow.

2) Controller Failure: During a controller failure, the network still carries traffic if the state of the network does not change. If there was a changing network condition such as a failure or new service, Homa can still use a greedy approach to perform the route update without having a global view of the network state. In this case, each router that detects a failure or new traffic will try to find a new route based on its own state of the network. If the router was not able to find any route whose bottleneck residual capacity is larger than the amount of demand for flow h, i.e. d_h, the flow will be dropped.

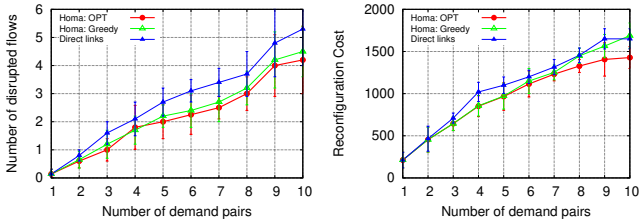
VI. Evaluation

We evaluate the performance of our proposed algorithm against the baseline solution: traditional SD-WAN direct links, where each source-destination pair in the overlay network is connected through the minimum cost direct overlay link, and optimal (OPT) under several scenarios. We evaluate each algorithm using two performance measures: 1) the network reconfiguration cost, and 2) number of disrupted flows.For each scenario, we randomize the results by running 10 different trials, where we vary the random selection of demand pairs from the entire set of nodes.



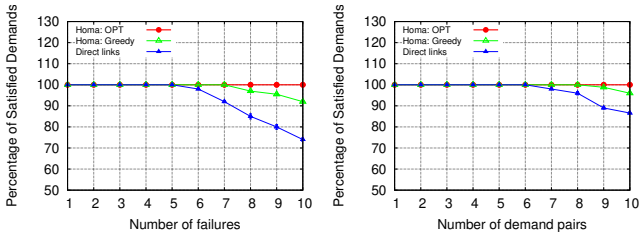
(a) Disrupted Flows. (b) Cost.

Fig. 5: Average number of disrupted flows (a), and cost of network reconfiguration (b), in Homa that uses indirect links and traditional SD-WAN that uses direct link architecture, as we increase the number of failures.



(a) Disrupted Flows. (b) Cost.

Fig. 6: Average number of disrupted flows (a), and cost of network reconfiguration (b), in Homa that uses indirect links and traditional SD-WAN that uses direct link architecture, as we increase the number of demand pairs.



(a) Increasing Failures. (b) Increasing Demands.

Fig. 7: Average percentage of satisfied demands, as we increase the number of failures and demand pairs.

We use real Internet Service Provider (ISP) topologies including the BellCanada topology with 48 nodes and 64 edges, taken from the Internet Topology Zoo [51, 52], and our SD-WAN testbed with 14 nodes and 16 edges shown in Figure 4a. We considered small size networks in order to be able to compare with the optimal solution in a reasonable time for the various operational settings, despite the NP-hardness of the problem.

A. Failure Resiliency

In this section, we perform several experiments to compare the network resiliency of our network update algorithms.

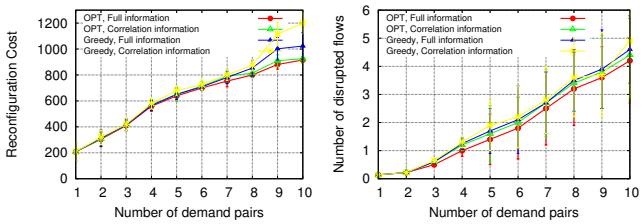
1) Direct/Indirect Links: In the first set of experiments, we quantify the failure resiliency of our route selection algorithm

with respect to the case where all source-destination flows in the overlay network are connected using direct links. We use the BellCanada topology where each link's capacity is chosen randomly in the interval $[20, 50]$. We consider six demand flows where each demand pair has a flow rate of three. We increase the random number of link failures in the underlay network topology from 1 to 10 and observe the total number of disrupted flows in Homa (Greedy) and optimal (OPT), with respect to the case when all source-destination pairs in the overlay network use a direct link to communicate. Figure 5 shows the experimental results for this scenario, where Figure 5a shows the total number of disrupted flows and Figure 5b shows the disruption cost. As shown, Homa disrupts fewer flows with lower disruption cost compared to the baseline where all source-destination pairs use the direct link. We also performed a similar experiment on the same network topology with 5 random disrupted links in the underlay network and increase the number of demand pairs from 1 to 10. As shown in Figures 6a and 6b, Homa experiences less number of disrupted flows and reconfigures the network with lower cost. This is mainly due to the fact that the non-disrupted indirect overlay links in Homa, can transfer traffic with lower cost compared to the higher cost non-disrupted direct links in the traditional SD-WAN baseline algorithm.

2) Demand Loss: In this scenario, addressed in Figure 7, we considered the BellCanada topology explained in Section VI-A1. We compare Homa (OPT), Homa (Greedy) and the baseline algorithm that always uses the direct overlay links to route each flow, to determine the amount of demand loss in the baseline and our greedy heuristic. Similar to the previous two experiments, we increase the number of failures in the underlay topology from 1 to 10 in one scenario and the number of demand pairs from 1 to 10 in another scenario where the number of failed underlay links is set to 5 in the second scenario. Figures 7a and 7b show the experimental results for this scenario. As we increase the number of failures in Figure 7a, Homa (OPT) reconfigures the network without dropping any flow, Homa (Greedy) loses only 5% of the flows, while the traditional SD-WAN baseline that uses direct links drops 30% of the flows when the number of failures is equal to 10. This is due to the fact that whenever all the three direct links fail due to an underlay link failure, the baseline algorithm cannot find any alternative paths and drops the flow, while Homa uses indirect links with minimum cost and finds an alternative path. We also observe that Homa (Greedy) only drops 2% of the demands in the second scenario as we increase the number of demand pairs to 10. Since the greedy heuristic does not explore all possible paths, it drops the flows once the residual capacity of the bottleneck link is smaller than the demand d_h. However, the greedy heuristic still performs better than the baseline approach that uses the direct overlay links.

B. Inferring the underlay topology

In this set of experiments, we use our testbed network topology with 14 nodes and 16 edges shown in Figure 4a.



(a) Reconfiguration Cost. (b) Disrupted Flows.

Fig. 8: Comparison between (a) network reconfiguration cost, and (b) number of disrupted flows, when we have full information about the underlay network topology and when using the correlation matrix from the underlay network topology.

We increase the number of demand pairs from 1 to 10. Also, we assume a single random failure in the underlay network topology and observe 1) the network reconfiguration cost, and 2) number disrupted flows, assuming 1) we have full observability about the underlay network topology, and 2) using the correlation matrix from the monitoring system. Figure 8 shows the experimental results for this scenario. As shown, the results using the correlation matrix is very close to the case when we have full observability of the underlay network topology.

VII. Conclusion

Traditional SD-WAN frameworks perform poorly on new network conditions such as failures, changing topology, service, traffic, or new policy requirements. We identified the key causes of poor performance as follows: (1) the overlay path for different flows usually share the same congested links, (2) in multi-path routing approaches the primary and backup paths usually share the same link, and (3) network reconfiguration and routing decisions are usually made locally. Addressing these issues require judicious centralized routing decisions based on the global network state to respond quickly to the changing network conditions.

In this paper, we proposed *Homa* to perform such decisions in SD-WANs while minimizing the network reconfiguration cost during the update. We formulated the problem as an integer linear programming which is NP-Hard. We proposed a greedy algorithm that solves the *Min-Cost* problem with bounded approximation on the total cost. Experimental results on real network topologies demonstrated the effectiveness of the proposed approach in terms of disruption cost, and average disrupted flows.

Acknowledgement

We thank the anonymous reviewers for their feedback on earlier drafts of this paper. This research is supported in part by NSF grant CCF-1763970.

References

[1] Ensuring High Uptime with SD-WAN. https://www.catonetworks.com/blog/ensuring-high-uptime-with-sd-wan/, 2018. [Online; accessed 28-May-2019].

[2] White paper: Aruba sd-branch overview. https://www.arubanetworks.com/assets/wp/WP_SDBranchOverview.pdf, 2018. [Online; accessed 28-May-2019].

[3] White paper: Contrail sd-wan design & architecture guide. https://www.juniper.net/documentation/en_US/release-independent/solutions/information-products/pathway-pages/sg-007-contrail-sd-wan-design-architecture.pdf, 2018. [Online; accessed 28-May-2019].

[4] B. Munch, S. Slaymaker, A. Lerner, and N. Rickard. Market guide for software defined wan. *Gartner, Inc.*, 2015.

[5] MPLS Explained. https://www.networkworld.com/article/2297171/network-security-mpls-explained.html, 2019. [Online; accessed 28-July-2019].

[6] SD-WAN Deployment Architectures. https://www.juniper.net/documentation/en_US/cso4.1/topics/concept/sd-wan-deployment-architectures.html, 2019. [Online; accessed 28-May-2019].

[7] S. Jain, A. Kumar, S. Mandal, J. Ong, L. Poutievski, A. Singh, S. Venkata, J. Wanderer, J. Zhou, M. Zhu, et al. B4: Experience with a globally-deployed software defined wan. In *ACM SIGCOMM Computer Communication Review*, 2013.

[8] Citrix SD-WAN Reference Architecture. https://docs.citrix.com/en-us/tech-zone/design/reference-architectures/sdwan.html, 2019. [Online; accessed 28-May-2019].

[9] C. Y. Hong, S. Kandula, R. Mahajan, M. Zhang, V. Gill, M. Nanduri, and R. Wattenhofer. Achieving high utilization with software-driven wan. In *ACM SIGCOMM Computer Communication Review*. ACM, 2013.

[10] S. Kandula, I. Menache, R. Schwartz, and S. R. Babbula. Calendaring for wide area networks. In *ACM SIGCOMM computer communication review*. ACM, 2014.

[11] J. Zheng, Q. Ma, C. Tian, B. Li, H. Dai, H. Xu, G. Chen, and Q. Ni. Hermes: Utility-aware network update in software-defined wans. In *IEEE ICNP*, pages 231–240, 2018.

[12] M. Casado, M. J. Freedman, J. Pettit, J. Luo, N. McKeown, and S. Shenker. Ethane: Taking control of the enterprise. In *ACM SIGCOMM Computer Communication Review*. ACM, 2007.

[13] R. Pang, M. Allman, M. Bennett, J. Lee, V. Paxson, and B. Tierney. A first look at modern enterprise traffic. In *ACM SIGCOMM IMC*, 2005.

[14] Y. Chen, S. Jain, V. K. Adhikari, Z. L. Zhang, and K. Xu. A first look at inter-data center traffic characteristics via yahoo! datasets. In *2011 Proceedings IEEE INFOCOM*, 2011.

[15] N. Laoutaris, M. Sirivianos, X. Yang, and P. Rodriguez. Inter-datacenter bulk transfers with netstitcher. In *ACM SIGCOMM Computer Communication Review*, 2011.

[16] D. Andersen, H. Balakrishnan, F. Kaashoek, and R. Morris. Resilient overlay networks. *SIGOPS Operating Systems Review*, 2001.

[17] W. Cui, I. Stoica, and R. H. Katz. Backup path allocation based on a correlated link failure probability model in overlay networks. In *10th IEEE International Conference on Network Protocols*, 2002.

[18] S. Tian, J. Liao, T. Li, J. Wang, and G. Cui. Resilient routing overlay network construction with super-relay nodes. *KSII Transactions on Internet & Information Systems*, 2017.

[19] L. Subramanian, I. Stoica, H. Balakrishnan, and R. H. Katz. Overqos: An overlay based architecture for enhancing internet qos. In *NSDI*, 2004.

[20] R. Cohen and D. Raz. Cost-effective resource allocation of overlay routing relay nodes. *IEEE/ACM Transactions on Networking (TON)*, 2014.

[21] E. Modiano and A. Narula-Tam. Survivable routing of logical topologies in wdm networks. In *Proceedings IEEE INFOCOM 2001.*, pages 348–357 vol.1, April 2001.

[22] E. Modiano and A. Narula-Tam. Survivable lightpath routing: a new approach to the design of wdm-based networks. *IEEE Journal on Selected Areas in Communications*, 20(4):800–809, May 2002.

[23] S. Roy, H. Pucha, Z. Zhang, Y. C. Hu, and L. Qiu. On the placement of infrastructure overlay nodes. *IEEE/ACM Transactions on Networking*, 2009.

[24] S. Ratnasamy, M. Handley, R. Karp, and S. Shenker. Topologically-aware overlay construction and server selection. In *Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies*, 2002.

[25] M. Kwon and S. Fahmy. Topology-aware overlay networks for group communication. In *Proceedings of the 12th International Workshop on Network and Operating Systems Support for Digital Audio and Video*, NOSSDAV '02, 2002.

[26] M. Waldvogel and R. Rinaldi. Efficient topology-aware overlay network. *SIGCOMM Computer Communication Review*, 2003.

[27] J. Han, D. Watson, and F. Jahanian. Topology aware overlay networks. In *Proceedings of IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies.*, 2005.

[28] A. Young, J. Chen, Z. Ma, A. Krishnamurthy, L. Peterson, and R. Y. Wang. Overlay mesh construction using interleaved spanning trees. In *IEEE INFOCOM*, 2004.

[29] J. Fan and M. H. Ammar. Dynamic topology configuration in service overlay networks: A study of reconfiguration policies. In *IEEE INFOCOM*, 2006.

[30] A. Nakao, L. Peterson, and A. Bavier. A routing underlay for overlay networks. In *Proceedings of the 2003 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, 2003.

[31] R. Keralapura, N. Taft, C. N. Chuah, and G. Iannaccone. Can isps take the heat from overlay networks. In *ACM SIGCOMM Workshop on Hot Topics in Networks (HotNets)*, 2004.

[32] H. Zhang, J. F. Kurose, and D. F. Towsley. Can an overlay compensate for a careless underlay? In *IEEE INFOCOM*, 2006.

[33] A. Adams, T. Bu, T. Friedman, J. Horowitz, D. Towsley, R. Caceres, N. Duffield, F. L. Presti, S. B. Moon, and V. Paxson. The use of end-to-end multicast measurements for characterizing internal network behavior. *IEEE Communications Magazine*, 2000.

[34] Y. Vardi. Network tomography: Estimating source-destination traffic intensities from link data. *Journal of the American statistical association*, 1996.

[35] D. Z. Tootaghaj, H. Khamfroush, N. Bartolini, S. Ciavarella, S. Hayes, and T. La Porta. Network recovery from massive failures under uncertain knowledge of damages. In *the Proceedings of the IFIP Networking Conference*, 2017.

[36] D. Z. Tootaghaj, T. He, and T. La Porta. Parsimonious tomography: Optimizing cost-identifiability trade-off for probing-based network monitoring. *SIGMETRICS Performance Evaluation Review*, 2018.

[37] D. Z. Tootaghaj, N. Bartolini, H. Khamfroush, and T. La Porta. Controlling cascading failures in interdependent networks under incomplete knowledge. In *The 36th IEEE International Symposium on Reliable Distributed Systems (SRDS)*, 2017.

[38] W. Stallings. Snmp and snmpv2: the infrastructure for network management. *IEEE Communications Magazine*, 1998.

[39] D. Andersen, H. Balakrishnan, F. Kaashoek, and R. Morris. *Resilient overlay networks*. ACM, 2001.

[40] N. Feamster, D. G. Andersen, H. Balakrishnan, and M. F. Kaashoek. Measuring the effects of internet path faults on reactive routing. In *ACM SIGMETRICS Performance Evaluation Review*, 2003.

[41] V. Paxson. End-to-end routing behavior in the internet. *IEEE/ACM transactions on Networking*, 1997.

[42] M. Luckie et al. Challenges in inferring internet interdomain congestion. In *IMC*. ACM, 2014.

[43] N. Fraenkel, D. Shaked, and M. C. Harel. Inferring a network topology, April 26 2018. US Patent App. 15/558,659.

[44] Z. Duan, Z. L. Zhang, and Y. T. Hou. Service overlay networks: Slas, qos, and bandwidth provisioning. *IEEE/ACM Transactions on Networking (TON)*, 2003.

[45] O. Papaemmanouil, Y. Ahmad, U. Cetintemel, and J. Jannotti. Application-aware overlay networks for data dissemination. In *Proceedings. 22nd International Conference on Data Engineering Workshops*. IEEE, 2006.

[46] C. Chekuri, M. T. Hajiaghayi, G. Kortsarz, and M. R. Salavatipour. Approximation algorithms for node-weighted buy-at-bulk network design. In *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms*. Citeseer, 2007.

[47] M. Andrews. Hardness of buy-at-bulk network design. In *45th Annual IEEE Symposium on Foundations of Computer Science*, 2004.

[48] M. Charikar and A. Karagiozova. On non-uniform multicommodity buy-at-bulk network design. In *Proceedings of the thirty-seventh annual ACM symposium on Theory of computing*. ACM, 2005.

[49] R. Hassin. Approximation schemes for the restricted shortest path problem. *Mathematics of Operations research*, 1992.

[50] D. H. Lorenz and D. Raz. A simple efficient approximation scheme for the restricted shortest path problem. *Operations Research Letters*, 2001.

[51] S. Knight, H. X. Nguyen, N. Falkner, R. Bowden, and M. Roughan. The internet topology zoo. *IEEE Journal on Selected Areas in Communications*, 2011.

[52] The internet topology zoo. `http://www.topology-zoo.org/`, accessed in May, 2015.