Incremental Deployment of SDN in Hybrid Enterprise and ISP Networks

David Ke Hong*, Yadi Ma†, Sujata Banerjee†, Z. Morley Mao*
* University of Michigan, † Hewlett Packard Labs
{kehong, zmao}@umich.edu, {yadi.ma, sujata.banerjee}@hpe.com

ABSTRACT

Introducing SDN into an existing network causes both deployment and operational issues. A systematic incremental deployment methodology as well as a hybrid operation model is needed. We present such a system for incremental deployment of hybrid SDN networks consisting of both legacy forwarding devices (i.e., traditional IP routers) and programmable SDN switches. We design the system on a production SDN controller to answer the following questions: which legacy devices to upgrade to SDN, and how legacy and SDN devices can interoperate in a hybrid environment to satisfy a variety of traffic engineering (TE) goals such as load balancing and fast failure recovery. Evaluation on real ISP and enterprise topologies shows that with only 20% devices upgraded to SDN, our system reduces the maximum link usage by an average of 32% compared with pure-legacy networks (shortest path routing), while only requiring an average of 41% of flow table capacity compared with pure-SDN networks.

Categories and Subject Descriptors

C.2.3 [**Network Operations**]: Network management

Keywords

Software-Defined Networks, incremental deployment

1. INTRODUCTION

Software-defined networking (SDN) enables dynamic programming of network-wide packet forwarding and provides flexible means for achieving centralized, fine-grained traffic engineering (TE) and fast failure recovery. This power has been unleashed by cloud data centers and private WANs [11, 18, 17]. However, many existing SDN applications can realize their potential only when SDN is fully deployed. In reality, upgrading all existing legacy devices into SDN-capable ones poses huge budget and operational burden on enterprises and Internet Service Providers (ISPs). This is particularly true in large scale legacy enterprise networks where it is common to have long lifecycles of network devices. Thus enterprise and ISP network operators resort to incrementally deploying SDN devices in their existing networks and tailor their SDN applications to work in a hybrid environment. For example, many enterprises have resorted to upgrading just their edge devices to SDN for QoS and security related applications. This upgrade strategy may not work well for TE and failure recovery applications.

Unfortunately, we find a lack of strategies from existing work that can systematically compute the best upgrade options for the maximum return of network benefits, as well as operate the hybrid infrastructure in an optimal way. Our focus is on TE benefits, such as balancing link load and recovery from network failures. Previous works do not study the trade-offs between the cost of SDN deployment and the TE benefits SDN brings. In this paper, we aim to answer the following questions: where and how many SDN switches should be deployed for meeting specific TE goals given budget and resource constraints? Is there a trade-off between the cost of SDN deployment and TE performance gain? How should legacy and SDN devices interact in a hybrid network to yield the maximum benefit?

To answer these questions, we propose to strategically select a subset of legacy devices from an existing network and upgrade to SDN[1], without changing its topology or the configuration of existing legacy components. Specifically, we tackle the following challenges: 1) Optimization-based cost-effective selection of legacy components for SDN upgrading; 2) Real-time attainment of a global network view consisting of both SDN and legacy devices; 3) A TE scheme that interoperates between SDN's rule-based forwarding and the default routing behavior of legacy components to achieve specific TE or failover goals.

As we illustrate by evaluation on real enterprise and ISP topologies, randomly picking switches for SDN upgrading cannot fully exploit SDN's centralized control. Though upgrade planning is an offline process, it is unrealistic to enumerate all possible candidate sets of switches for evaluation, given the scale of an enterprise or ISP network and its dynamic traffic demands. Even though traffic demands can be approximated beforehand, picking an optimal set of switches for upgrading while meeting budget and resource constraints

SOSR '16, March 14 - 15, 2016, Santa Clara, CA, USA

ISBN 978-1-4503-4211-7/16/03...$15.00
DOI: http://dx.doi.org/10.1145/2890955.2890959

[1]Upgrading a device to SDN today often implies being able to support a version of the OpenFlow protocol with appropriate hardware and software support. A robust, scalable and highly available SDN controller with specific applications also needs to be included in the infrastructure.

is an integer-linear programming problem and is known to be NP-complete. Fortunately, we observe that enterprise and ISP networks share some common characteristics, which motivates us to design simple but effective heuristics that work in both networks for incrementally selecting switches to upgrade. Furthermore, operating such a hybrid network is challenging in several aspects. First, it is hard to obtain a global view consisting of both SDN and legacy nodes because legacy nodes in a hybrid network do not participate in the topology discovery protocol adopted by many SDN controllers. Second, the routing of legacy nodes are limited to shortest path and equal-cost multipath (ECMP), while OpenFlow/SDN provides flexible routing features. We demonstrate that it is non-trivial to operate the hybrid infrastructure in a way that accommodates legacy routing protocols while leveraging SDN's centralized, fine-grained routing control.

Our main contributions are summarized below:

- We systematically study the incremental SDN deployment problem by formulating it as an optimization problem, and propose effective heuristics for selecting a small set of existing devices for upgrading.
- We maintain a global view of the hybrid topology through interactions between legacy and SDN devices.
- We design a hybrid traffic engineering approach, to leverage SDN's fine-grained and flexible packet forwarding in the hybrid infrastructure, which reduces the maximum link usage by an average of 32% compared with legacy shortest path routing.
- We analyze the trade-offs between the cost of incremental SDN deployment and the TE benefits it brings. With only 20% devices upgraded, close to optimal results can be achieved on real enterprise/ISP topologies.

2. CASE STUDY

We conducted a case study on real-world enterprise and ISP networks to understand their topological characteristics and protocol usage, and observe that both networks share some common characteristics in organization and routing. These similarities enable us to design converged incremental SDN deployment and traffic engineering strategies that also consider the different traffic characteristics and network usage for both networks. First, both networks are organized in a multi-layer hierarchy. Typical large enterprise networks usually span multiple sites and are organized into three layers – edge/access, distribution/aggregation and core layer. End hosts are connected to the access switches in the edge layer. The distribution layer, consisting of aggregation switches and routers, interconnects access switches to the core layer. The core layer spans multiple sites to provide inter-site connectivity. Similarly, an ISP network consists of layers including the customer edge, provider edge, and Internet backbone. Routers within each layer interconnect adjacent layers. Second, similar routing protocols are configured in both networks. In enterprise networks, intra-site routing protocols include OSPF and IS-IS, while inter-site traffic is normally routed by BGP or through MPLS cloud. In ISP networks, IGPs including OSPF are widely used for intra-AS (autonomous system) routing, while BGP and MPLS are regular configurations for inter-AS routing. Third, redundant links are provisioned between adjacent layers in both networks for load balancing and failure resilience.

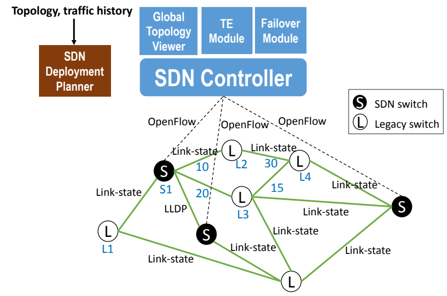


Figure 1: System architecture.

We confirmed these observations from anecdotal and informal discussions with network administrators, and existing surveys [6, 22],

High availability and bounded congestion are key requirements for enterprises and ISP networks [1, 21]. SDN's centralized control is promising in improving both metrics for existing legacy-based enterprise and ISP networks. First, SDN provides an expressive interface for performing network-wide data-plane re-configurations to accommodate network congestion and failures, and relieves network operators of error-prone, box-by-box network configurations. Second, by pushing per-flow forwarding rules to SDN switches, fine-grained TE polices can be applied to various traffic types (e.g., latency-sensitive or throughput-intensive) for improving differentiated services and meeting service level objectives. Third, compared with ECMP in legacy devices which evenly splits traffic among equal-cost paths, SDN switches flexibly support arbitrary splitting of traffic across multiple paths and fast failover with the help of *select* group tables type in OpenFlow 1.1 [7]. This feature enables traffic-aware weighted multipath load balancing and fast congestion-aware failover. Replacing hundreds or thousands of existing well-functioning legacy network devices in an enterprise or ISP network with SDN-capable ones, however, is financially and technically impractical. Incremental SDN deployment can partially and cost-effectively bring in aforementioned TE benefits for an enterprise or ISP network. We are seeking to identify a "sweet spot" that achieves close-to-optimal TE performance within a given budget of SDN investment.

3. DESIGN

Our system contains four major components, as shown in Figure 1. The first component is an offline incremental *SDN Deployment Planner* that runs periodically (e.g., when new budget is available) and decides which legacy devices to upgrade. The remaining components run in an SDN controller. The *Global Topology Viewer* maintains a global view of the hybrid network through interactions between legacy and SDN devices and reflects dynamic changes such as link/switch up or down in real-time. The *TE Module* meets TE goals by controlling forwarding paths of any flow that traverses an SDN switch. The *Failover Module* avoids link congestion under failures and ensures fast failure recovery. Hereafter, we denote a legacy switch or router as a *legacy node* and an SDN-capable switch as an *SDN node*, for the ease of writing.

3.1 SDN Deployment Planner

The SDN Deployment Planner plans in advance or periodically, to choose a subset of legacy devices to upgrade for partial SDN deployment, by analyzing enterprise or ISP network topologies, traffic demand history, and resource constraints. This information is often routinely available in network management systems, or in an approximate form from discussions with the network administrators. The analysis determines a set of legacy devices to be upgraded to SDN.

3.1.1 Problem formulation

We formulate the SDN deployment problem as an optimization problem to minimize the maximum link utilization subject to constraints such as link capacity and upgrade budget. We could also take into consideration constraints such as available SDN versions, and other hardware resource constraints (e.g., flow table size). Other pragmatic constraints could also be introduced, such as the fact that devices at the edge of the network may be easier to upgrade. Compared to traditional load balancing problems, the unknown set of switches to be upgraded makes load balanced paths nondeterministic. As a result, two unknowns need to be solved, one for selecting switches to upgrade, the other for selecting paths to balance traffic.

Given the network topology as a directed graph $G = (V, E)$, a set of end-to-end traffic demands $D = \{s, t \in V | d_{s,t}\}$, and an upgrade budget in terms of the maximum percentage γ of the total legacy nodes to be upgraded, we formulate the SDN deployment and traffic engineering problem based on path-constrained multi-commodity flow problem [14] as follows, with the goal of minimizing maximum link usage (link load over link bandwidth):

Minimize:

$$\max_{l \in E} \frac{Load_l}{BW_l} \tag{1}$$

Subject to:

$$Load_l = SLoad_l + BLoad_l - RLoad_l \tag{2}$$

$$SLoad_l = \sum_{s \in V} \sum_{t \in V} Z_{l,SP_{s,t}} Y_{SP_{s,t}} d_{s,t} \tag{3}$$

$$BLoad_l = \sum_{s \in V} \sum_{t \in V} \sum_{a \in V} \sum_{l' \in E-\{l\}} \sum_{p \in BP_{s,t,a,l'}} Z_{l',p} X_a Y_p d_{s,t} \tag{4}$$

$$RLoad_l = \sum_{s \in V} \sum_{t \in V} \sum_{a \in V} \sum_{p \in BP_{s,t,a,l}} X_a Y_p d_{s,t} \tag{5}$$

$$\sum_{a \in V} X_a \le \gamma |V| \tag{6}$$

$$\forall l \in E : Load_l \le \alpha BW_l, 0 \le \alpha \le 1 \tag{7}$$

$$\forall d_{s,t} \in D : \sum_{a \in V} \sum_{l \in E} \sum_{p \in BP_{s,t,a,l}} X_a Y_p + Y_{SP_{s,t}} \ge 1 \tag{8}$$

$$\forall a \in V, l \in E, p \in SP \cup BP : X_a \in \{0,1\}, 0 \le Y_p \le 1, Z_{l,p} \in \{0,1\} \tag{9}$$

The variables used in the formulation are summarized in table 1. Eq. 6 captures the upgrade budget constraint by ensuring the number of upgraded SDN nodes will not exceed γ of total nodes. Eq. 7 ensures each link is not overloaded, where α is a threshold usually less than 1. Eq. 8 captures the demand constraint by ensuring each traffic demand is fully routed through primary and backup paths.

We observe that a link can be traversed by a set of primary shortest paths and backup load-balanced paths. Therefore, bandwidth of a link can be consumed by traffic routed along primary paths and along other backup paths. Meanwhile, traffic can be moved away from a link along its primary path due to load balancing to backup paths. Therefore, we can decompose the traffic load imposed on a link, formulated in Eq. 2, into above three components. X_a and Y_p are unknowns in the formulation, while the other three terms can be pre-computed by Eq. 10.

Variable	Description
X_a	Indicator variable for selecting node a to upgrade
Y_p	Indicator variable for selecting path p
$Z_{l,p}$	Indicator variable for link $l \in$ path p
$Load_l$	Total traffic demands over link l
$SLoad_l$	Total traffic demands routed along primary paths that contain link l
$BLoad_l$	Total traffic demands re-routed from all links but link l along backup paths that contain l
$RLoad_l$	Total traffic demands re-routed from link l along some backup paths
$SP_{s,t}$	Primary shortest path from s to t
$BP_{s,t,a,l}$	Set of backup paths from s to t that circumvents link l using upgraded node a
$DFS_{V,E-\{l\},s,t}$	Set of paths from s to t in $G' = (V, E - \{l\})$ by depth-first search
$l << a$	Node a appears after link l in $SP_{s,t}$
$a << l$	Node a appears before link l in $SP_{s,t}$

Table 1: Summary of variables.

$$BP_{s,t,a,l} = \begin{cases} SP_{s,t} & \forall a \in V, l \notin SP_{s,t} \\ \emptyset & \forall a \notin SP_{s,t}, l \in SP_{s,t} \\ \emptyset & \forall a \in SP_{s,t}, l \in SP_{s,t}, l << a \\ DFS_{V,E-\{l\},s,t} & \forall a \in SP_{s,t}, l \in SP_{s,t}, a << l \end{cases} \tag{10}$$

Note that the formulation contains a bi-linear term of unknowns, and solving the unknowns is as hard as solving a mixed integer bi-linear program. Though it can be linearized into a mixed integer linear program by adding McCormick envelopes [15], the problem is still NP-complete [13]. Instead, we develop several simple but effective heuristics to tackle the problem.

3.1.2 Heuristic-based approach

We propose several traffic-agnostic or traffic-aware heuristics for selecting a subset of legacy nodes to upgrade based on the topological property of the network. The first heuristic greedily picks legacy nodes with higher degrees (sum of incoming and outgoing degrees) in the topology graph (i.e., S1 and L3 in Figure 1), based on the intuition that heavily connected nodes are likely to be traversed by more end-to-end routing paths. In the second heuristic, we consider link weights for legacy routing and compute a set of forwarding paths for each pair of source and destination devices using K-shortest path algorithm. Then we greedily pick nodes that appear most frequently in these paths. We also take traffic demand into consideration and select nodes with highest traffic volume. We evaluated all three heuristics on various topologies and observe no obvious differences among them, so we only report results from the first heuristic.

3.2 Global Topology Viewer

A network-wide view of topology is essential for the controller to dynamically distribute traffic for meeting TE requirements, and to detect link/switch failures in real time for fast failure recovery. Though topology information is usually available from other sources (e.g., network management system), they do not reflect dynamic changes in real time, such as link/switch up or down.

A typical topology discovery mechanism in SDN relies on Link Layer Discovery Protocol (LLDP) and Broadcast Domain Discovery Protocol (BDDP), where an SDN controller periodically commands SDN switches to flood LLDP and BDDP messages, and SDN switches that receive these messages forward them back to the controller. In this way, the controller discovers interconnection between SDN nodes. However, in a hybrid network, an SDN node can be interconnected with legacy nodes and LLDP packets flooded by the SDN node will be dropped by legacy nodes. Therefore, the SDN controller cannot discover the interconnection between these devices. Assuming link-state Interior Gateway Protocol (IGP) is configured on legacy nodes, we illustrate how to construct a global topology consisting of both SDN and legacy devices, using one IGP (i.e., OSPF) as an example.

Messages flooded by legacy routing protocol, such as OSPF link-state advertisements (LSAs), can be intercepted by intermediate SDN switches and forwarded as Packet-In to the SDN controller. We implemented a global topology module in HP Virtual Application Networks SDN controller [5], which is a production SDN controller. It parses LSAs to extract topology information and classifies between a router LSA and network LSA. Network LSA is broadcasted by a designated router which lists routers that are joined together by the network segment. Through router LSA, a router announces its presence and lists the links to other routers of the network. The topology viewer detects links between legacy devices through these LSAs and also link liveness when it sees differences between previous and current connection states. Links between SDN switches are detected by the controller's built-in link service manager using LLDP and BDDP as described above. A link between an SDN switch and legacy switch is detected by OSPF Hello message where a TIMEOUT indicates the link is down. Therefore, our topology viewer maintains a centralized global network topology that reflects topology changes in real time.

The idea can be extended to handle other routing protocols, like BGP. In Google's B4 [18], each switch has an OpenFlow Agent forwarding routing protocol packets (e.g., BGP/IS-IS) to a remote OpenFlow Controller. Though we can not change legacy switches' behavior as in B4, we could employ other approaches. For example, we could configure the SDN switches to maintain BGP sessions with any router in ASes to receive BGP route updates and forward them to the SDN controller. We could also employ the approach in the SNMP4SDN ODL project [9], where legacy switches are configured by CLI to send SNMP trap to a plugin in the controller when the switch boots up, and the plugin also queries LLDP data on legacy switches for topology discovery.

To retrieve real-time link load, we propose to leverage the meter table feature available in OpenFlow 1.3 [8] for measuring per-flow packet rate. Since the controller has global knowledge of routing paths for each pair of source and destination, if a flow traverses at least one SDN node, its packet rate can be measured by the meter entries that are attached to flow entries in the flow table of any SDN switch the flow traverses. For a flow that does not traverse any SDN node along its routing path, SNMP-based estimation of bandwidth utilization [4] can be applied to estimate its packet rate. The latter case is rare based on our experimental study on real ISP and enterprise topologies and traffic traces, which shows that around 90% of flows traverse at least on SDN node along their routing paths under 20% upgrading ratio. The controller periodically polls for the flow counters of meter entries or SNMP states and aggregates them for computing the network load of each link.

3.3 Traffic Engineering

The TE Module meets traffic engineering goals by controlling traffic forwarding paths. We start with load balancing as our TE goal, aiming to minimize the maximum link utilization to avoid link congestion. This is a common goal for WAN, ISP and enterprise networks, as you can pack more traffic without provisioning extra link capacity. SDN-based TE becomes ill-suited in such a hybrid network, because the forwarding behavior of non-SDN nodes cannot be controlled. Our strategy is to accommodate the default routing behavior in legacy nodes and apply SDN-based forwarding in a best-effort manner.

In our problem context, a flow is defined as end-to-end aggregate traffic demand from one source to a particular destination. Each new flow is first forwarded along the shortest path to be compatible with IGPs operated in legacy devices which commonly use Shortest Path First (SPF) algorithms for route calculation. Then the TE Module installs rules on SDN switches that the flow goes through to control forwarding paths based on routing policies and real-time link load. Whenever a new flow reaches an SDN switch, the TE Module implements several load balancing heuristics to forward the flow. One heuristic called *Hybrid-LLF* forwards the flow to an output port along the least loaded path where LLF stands for least loaded first. Take Figure 1 for an example, traffic flow from legacy switch L1 destined to L4 is first forwarded along its shortest path to reach SDN switch S1. S1 has two paths to reach L4, through L2 or L3. Assume the blue numbers under the links stand for the current link load, Hybrid-LLF would pick the path through L3 whose maximum link usage is smaller. Therefore, the TE module installs rules on S1 to forward the flow to the output port associated with L3.

Another heuristic called *Hybrid-Weighted* splits flows to go through multiple paths with different possibility by using the "select" group table feature supported by OpenFlow 1.1 [7]. For the same example, S1 would split flows to L4 along two paths, S1-L2-L4 or S1-L3-L4. Hybrid-Weighted could assign weights that are inversely proportional to the maximum link usage on each path (e.g. with weight 0.4, S1 forwards to L2; and with weight 0.6, it forwards to L3). We evaluated both heuristics on various topologies and find Hybrid-LLF outperforms Hybrid-Weighted, mainly because Hybrid-Weighted splits flows only at the first SDN node along IGP's routing path. Linear program can be used for determining the optimal set of forwarding paths and their splitting ratio across all SDN nodes. Our evaluation shows that it (namely **Hybrid-OPT**) outperforms LLF and achieves near-optimal performance. However, it is not suited for dynamic load balancing. We thus report results on Hybrid-LLF in the evaluation. If a flow does not reach any SDN switch, normal IGP-based routing is applied.

Besides load balancing, the Failover Module (Figure 1) aims to alleviate link congestion when failure happens and ensures fast failure recovery, which is part of our ongoing work. Other TE goals, such as minimizing the delay for a class of flows, can be conveniently supported as extensions to the TE Module and are our future work.

Topology	# Nodes	# Links	# Demands
ISP1	79	294	6162
ISP2	87	322	7441
ISP3	104	302	10593
ISP4	138	744	18790
ISP5	161	656	25330
ISP6	315	1944	96507
ENTR	419	1066	79398

Table 2: Summary of ISP and enterprise topologies.

4. EVALUATION

We evaluated the network performance and resource usage of partial SDN deployment using real-world ISP and enterprise topologies and traffic matrices. The key TE metric we focus on is the maximum link utilization of a network. Minimizing it alleviates congestion when the network is oversubscribed. We also investigated the path stretch of flows, as an approximation for network delay, and the maximum number of rules installed in an SDN switch, to estimate the required capacity of SDN switch flow tables. In addition, we studied the trade-offs between SDN upgrade cost and the performance benefit.

4.1 Experimental Setup

Topologies. Table 2 summarizes the real ISP and enterprise network topologies we evaluated and the size of end-to-end traffic demands placed into each network. We evaluated six real-world ISP topologies, labeled **ISP1** through **ISP6**, using traffic demands [16] that are publicly available at [2]. The enterprise network topology, **ENTR**, is parsed from router configurations of a real campus network [22]. Similar to Panopticon [19], access switches are pruned from the original topology, resulting in 419 switches. We generated synthetic traffic matrices using a real enterprise trace [20] from 22 subnets that are mapped to the subnets in the campus network. Each traffic demand from the enterprise trace is randomly assigned to a source and destination switch residing in the mapped subnets of the campus network. We evaluated multiple generated traffic matrices for the pruned topology and they show similar trend so we report results from one of them. We also scaled traffic demands according to the link capacity of the campus network.

Schemes evaluated. We compare our hybrid approach, **Hybrid-LLF** with two schemes in fully legacy networks, shortest path first routing **Legacy-SPF** and equal-cost multipath routing **Legacy-ECMP**. To compare with full SDN, we evaluated a scheme called **SDN-LLF** where traffic load is balanced using least-loaded-first heuristic. SDN-LLF paves end-to-end full paths rather than just controlling flow forwarding through SDN switches as in Hybrid-LLF.

To understand the optimality of minimizing the maximum link utilization, we formulated a linear programming (LP) based fractional multi-commodity flow (MCF) problem [13] and used GLPK 4.55 [3] to solve the optimal routing and traffic distribution (denoted as **OPT**). However, OPT are impractical as it assumes flows are arbitrarily split at each node and the size of all flows are known in advance. Instead, OPT serves as a lower bound to compare our results with.

4.2 Maximum Link Utilization

Figure 3 shows the maximum link utilization with various schemes: (1) full legacy (Legacy-ECMP and Legacy-SPF), (2) hybrid SDN with 20% upgrade ratio (Hybrid-LLF), (3) full SDN (SDN-LLF) and (4) OPT. Note that OPT values for some large topologies are missing as the LP formulation contains too many columns for GLPK to solve.

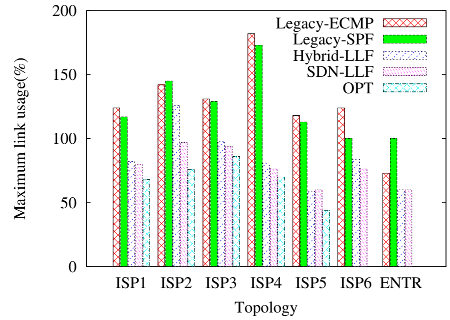
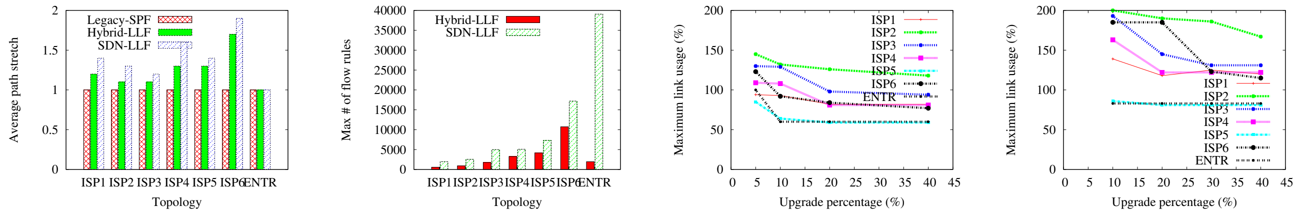


Figure 3: Maximum link usage (Hybrid-LLF with 20% SDN switches.)

We can see from Figure 3 that our hybrid SDN deployment planner and heuristic-based TE, namely Hybrid-LLF, reduces network congestion on heavily loaded links compared to the legacy schemes. Note that topologies with >100% maximum link usage indicate there are congested links. With 20% switches upgraded to SDN, Hybrid-LLF reduces the maximum link usage by an average of 32% and at most 53% (ISP4) compared to Legacy-SPF. Compared with SDN-LLF, Hybrid-LLF is only 6% higher on average. It is important to note that Legacy-SPF outperforms Legacy-ECMP for ISP topologies because we use pre-configured link costs in these topologies to calculate equal-cost paths while only a small number of such paths exist as those link costs vary a lot. For the ENTR topology, we use hop counts to calculate equal-cost paths and there are many such paths because of the high redundancy in the topology; therefore Legacy-ECMP outperforms Legacy-SPF for ENTR topology. Though maximum link utilization is the most important metric for measuring congestion, we also evaluated the average and variance of link utilization in each topology. Compared to Legacy-SPF, Hybrid-LLF's average link usage is 11% higher, while link usage variance is 13% lower. Compared to SDN-LLF, Hybrid-LLF's average link usage is 15% lower, and link usage variance is only 0.15% higher. Therefore, Hybrid-LLF effectively balances load across links and reduces link congestion.

Furthermore, we compute the optimal traffic distribution using the fractional MCF model with GLPK in our partial SDN deployment, denoted as **Hybrid-OPT**. We observe that the results of Hybrid-OPT are very close to OPT, indicating our SDN deployment planner is effective in selecting legacy nodes for upgrading. In addition, compared to the results of Hybrid-OPT, the maximum link utilization of Hybrid-LLF can be further reduced by 6.2% to 26.2% to achieve near-optimal performance. In summary, our topology-based SDN deployment scheme is effective in leveraging SDN's flexible packet forwarding for load balancing, while the LLF heuristic for load balancing still has room for further improvements. We leave this as future work.

In addition, we also evaluated a scheme which randomly selects nodes from the aggregation/distribution layer of each topology for upgrading and then calculated the optimal load distribution. Though the maximum link utilization of this random scheme is 15% lower than Legacy-SPF, it is 35% higher than Hybrid-LLF, which indicates that the current common practice of randomly upgrading SDN switches cannot exploit the full strength of SDN.



(a) Average path stretch (b) Flow table usage (c) Upgrade cost vs. max link usage (no scaling) (d) Upgrade cost vs. max link usage (1.5X scaling)

Figure 2: Performance evaluations.

4.3 Path Stretch

Figure 2(a) shows the average path stretch of flows, where the path stretch of a flow is defined as the number of hops that the flow traverses divided by the number of hops on the shortest path. Compared to Legacy-SPF, Hybrid-LLF and SDN-LLF incurs 24% and 40% larger average path stretch, respectively, while the worst-case path stretch is increased by 27% to 220% from hybrid to full SDN deployment under the LLF-based load balancing. Moreover, optimal traffic distribution under hybrid or full SDN deployment leads to 1.6 to 5.2x increase in worst-case path stretch, indicating that the MCF-based traffic distribution incurs significant increase of path stretch for dynamic load balancing.

4.4 Switch Resource Usage

Figure 2(b) shows the maximum number of rules (i.e., flow table entries) used in any SDN switch for each topology. There is a general increasing trend as topology size and traffic demands increase from left to right. Operating the LLF heuristic in a hybrid network only requires an average of 41% (ranging from 5% to 62%) of the flow table capacity required in a full SDN network. Optimal traffic distribution in a hybrid of full SDN networks requires similarly large flow tables. Our planning of hybrid SDN deployment is compatible with using lower-end SDN switches and can potentially reduce the deployment budget. In summary, Hybrid-LLF trades little loss in load balancing performance for shorter traffic delay and lower switch upgrade cost.

4.5 Trade-off: Upgrade Cost vs. TE Benefits

With more SDN switches deployed, we expect to accommodate more traffic without increasing maximum link usage much. However, if the traffic demand is mostly stable, the benefit of increased SDN deployment is somewhat limited, as illustrated in Figure 2(c). It shows the load balancing performance of Hybrid-LLF with SDN upgrade percentage ranging from 5% to 40%. When upgrade percentage increases from 5% to 20%, there is a significant reduction in the maximum link usage for most topologies, while further upgrading over 20% incurs less or intangible improvement in load balancing. We further scaled up traffic demand by 1.5 times and report the maximum link usage with increasing SDN upgrade percentage, varying from 10% to 40%. As shown in Figure 2(d), link congestion is reduced as more SDN switches are deployed in most topologies, especially those topologies containing overloaded links (e.g. ISP2 and ISP6). Therefore, it is rewarding to incrementally deploy SDN when traffic demands increase.

5. RELATED WORK

Incremental deployment of SDN has been widely adopted in enterprises, but its unique challenges require careful routing design and treatment [24]. B4 [18] is an inter-datacenter WAN that relies on incremental deployment of SDN to achieve centralized TE for throughput improvement, but its scale is not suited for a large enterprise or ISP network. Panopticon [19] targets the problem of partial SDN deployment in enterprise networks for policy enforcement, but does not consider traffic engineering on dynamic traffic demands. [10] targets TE in hybrid SDN and solves the dynamic routing problem periodically using an efficient approximation scheme. [12] guarantees traffic reachability in case of any single link failure in hybrid SDN networks, but it does not address TE issues in normal scenarios without failures. Our approach not only plans for the SDN deployment but also reconfigures routing in real time to accommodate dynamic traffic demands.

Prior work also tries to achieve SDN-like flexible path enforcement with legacy networks. Fibbing [23] injects fake nodes and links into the underlying link state routing protocol to achieve some level of load balancing and traffic engineering, but its forwarding rule matching is limited to destination-based and its expressivity is thus confined to expressivity of IP routing. Besides, with injected "lies", Fibbing could lead to debugging issues and incorrect operation. DEFO [16] leverages segment routing to control routing paths for carrier-grade traffic engineering, but shares some similar limitations as in Fibbing. Besides, its constraint programming based middle-point selection largely focuses on static traffic matrices, while our proposed TE module can load balance dynamic traffic demands.

6. CONCLUSIONS AND FUTURE WORK

We present a systematic study and design on incrementally deploying SDN in enterprise and ISP networks to improve TE performance. Our proposed deployment planner and TE heuristics effectively reduce the maximum link usage for congestion mitigation with only 20% deployment of SDN on real ISP and enterprise topologies. Directions for future work include a system implementation in a testbed with scalability studies, and factoring in historical traffic demands on SDN deployment plans. Another consideration is the impact of increasing frequency of rule updates for routing path adjustments on consistent network updates and eventually on TE performance.

Acknowledgements

We would like to thank Ying Zhang, Jeongkeun Lee, Joon-Myung Kang, Mario Sanchez, Alvin AuYoung and anonymous reviewers for useful feedback and discussions.

7. REFERENCES

[1] Campus network for high availability design guide. http://www.cisco.com/c/en/us/td/docs/solutions/Enterprise/Campus/HA_campus_DG/hacampusdg.html.

[2] Declarative and expressive forwarding optimizer. http://sites.uclouvain.be/defo/.

[3] Glpk (gnu linear programming kit). https://www.gnu.org/software/glpk/.

[4] How to calculate bandwidth utilization using snmp. http://www.cisco.com/c/en/us/support/docs/ip/simple-network-management-protocol-snmp/8141-calculate-bandwidth-snmp.html.

[5] Hp virtual application networks sdn controller. http://h17007.www1.hp.com/docs/networking/solutions/sdn/4AA4-8807ENW.PDF.

[6] Medium enterprise design profile reference guide. http://www.cisco.com/c/en/us/td/docs/solutions/Enterprise/Medium_Enterprise_Design_Profile/MEDP.html.

[7] Openflow switch specification version 1.1.0. http://archive.openflow.org/documents/openflow-spec-v1.1.0.pdf.

[8] Openflow switch specification version 1.3.1. https://www.opennetworking.org/images/stories/downloads/sdn-resources/onf-specifications/openflow/openflow-spec-v1.3.1.pdf.

[9] Snmp4sdn: Architecture and design. https://wiki.opendaylight.org/view/SNMP4SDN:Architecture_and_Design.

[10] S. Agarwal, M. Kodialam, and T. Lakshman. Traffic engineering in software defined networks. In *INFOCOM, 2013 Proceedings IEEE*, pages 2211–2219, April 2013.

[11] M. Al-Fares, S. Radhakrishnan, B. Raghavan, N. Huang, and A. Vahdat. Hedera: Dynamic flow scheduling for data center networks. In *Proceedings of the 7th USENIX Conference on Networked Systems Design and Implementation*, NSDI'10, pages 19–19, Berkeley, CA, USA, 2010. USENIX Association.

[12] C.-Y. Chu, K. Xi, M. Luo, and H. Chao. Congestion-aware single link failure recovery in hybrid sdn networks. In *Computer Communications (INFOCOM), 2015 IEEE Conference on*, pages 1086–1094, April 2015.

[13] T. H. Cormen, C. Stein, R. L. Rivest, and C. E. Leiserson. *Introduction to Algorithms*. McGraw-Hill Higher Education, 2nd edition, 2001.

[14] S. Even, A. Itai, and A. Shamir. On the complexity of time table and multi-commodity flow problems. In *Proceedings of the 16th Annual Symposium on Foundations of Computer Science*, SFCS '75, pages 184–193, Washington, DC, USA, 1975. IEEE Computer Society.

[15] A. Gupte, S. Ahmed, M. S. Cheon, and S. Dey. Solving mixed integer bilinear problems using milp formulations. *SIAM Journal on Optimization*, 23(2):721–744, 2013.

[16] R. Hartert, S. Vissicchio, P. Schaus, O. Bonaventure, C. Filsfils, T. Telkamp, and P. Francois. A declarative and expressive approach to control forwarding paths in carrier-grade networks. In *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication*, SIGCOMM '15, pages 15–28, New York, NY, USA, 2015. ACM.

[17] C.-Y. Hong, S. Kandula, R. Mahajan, M. Zhang, V. Gill, M. Nanduri, and R. Wattenhofer. Achieving high utilization with software-driven wan. In *Proceedings of the ACM SIGCOMM 2013 Conference on SIGCOMM*, SIGCOMM '13, pages 15–26, New York, NY, USA, 2013. ACM.

[18] S. Jain, A. Kumar, S. Mandal, J. Ong, L. Poutievski, A. Singh, S. Venkata, J. Wanderer, J. Zhou, M. Zhu, J. Zolla, U. Hölzle, S. Stuart, and A. Vahdat. B4: Experience with a globally-deployed software defined wan. In *Proceedings of the ACM SIGCOMM 2013 Conference on SIGCOMM*, SIGCOMM '13, pages 3–14, New York, NY, USA, 2013. ACM.

[19] D. Levin, M. Canini, S. Schmid, F. Schaffert, and A. Feldmann. Panopticon: Reaping the benefits of incremental sdn deployment in enterprise networks. In *Proceedings of the 2014 USENIX Conference on USENIX Annual Technical Conference*, USENIX ATC'14, pages 333–346, Berkeley, CA, USA, 2014. USENIX Association.

[20] R. Pang, M. Allman, M. Bennett, J. Lee, V. Paxson, and B. Tierney. A first look at modern enterprise traffic. In *Proceedings of the 5th ACM SIGCOMM Conference on Internet Measurement*, IMC '05, pages 2–2, Berkeley, CA, USA, 2005. USENIX Association.

[21] M. Suchara, D. Xu, R. Doverspike, D. Johnson, and J. Rexford. Network architecture for joint failure recovery and traffic engineering. In *Proceedings of the ACM SIGMETRICS Joint International Conference on Measurement and Modeling of Computer Systems*, SIGMETRICS '11, pages 97–108, New York, NY, USA, 2011. ACM.

[22] Y.-W. E. Sung, S. G. Rao, G. G. Xie, and D. A. Maltz. Towards systematic design of enterprise networks. In *Proceedings of the 2008 ACM CoNEXT Conference*, CoNEXT '08, pages 22:1–22:12, New York, NY, USA, 2008. ACM.

[23] S. Vissicchio, O. Tilmans, L. Vanbever, and J. Rexford. Central control over distributed routing. In *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication*, SIGCOMM '15, pages 43–56, New York, NY, USA, 2015. ACM.

[24] S. Vissicchio, L. Vanbever, and O. Bonaventure. Opportunities and research challenges of hybrid software defined networks. *SIGCOMM Comput. Commun. Rev.*, 44(2):70–75, Apr. 2014.