Capacity-Efficient and Uncertainty-Resilient Backbone Network Planning with Hose

Satyajeet Singh Ahuja
Facebook

Varun Gupta
Facebook

Vinayak Dangui
Facebook

Soshant Bali
Facebook

Abishek Gopalan
Facebook

Hao Zhong
Facebook

Petr Lapukhov
Facebook

Yiting Xia
Max Planck Institute for Informatics

Ying Zhang
Facebook

ABSTRACT

This paper presents Facebook's design and operational experience of a Hose-based backbone network planning system. This initial adoption of the Hose model in network planning is driven by the capacity and demand uncertainty pressure of backbone expansion. Since the Hose model abstracts the aggregated traffic demand per site, peak traffic flows at different times can be multiplexed to save capacity and buffer traffic spikes. Our core design involves heuristic algorithms to select Hose-compliant traffic matrices and cross-layer optimization between the optical and IP networks. We evaluate the system performance in production and share insights from years of production experience. Hose-based network planning can save 17.4% capacity and drops 75% less traffic under fiber cuts. As the first study of Hose in network planning, our work has the potential to inspire follow-up research.

CCS CONCEPTS

• **Networks** → **Network management**; **Network experimentation**;

KEYWORDS

Wide-area networks, Network planning, Network modeling, Network optimization

ACM Reference Format:
Satyajeet Singh Ahuja, Varun Gupta, Vinayak Dangui, Soshant Bali, Abishek Gopalan, Hao Zhong, Petr Lapukhov, Yiting Xia, and Ying Zhang. 2021. Capacity-Efficient and Uncertainty-Resilient Backbone Network Planning with Hose . In *ACM SIGCOMM 2021 Conference (SIGCOMM '21), August 23–28, 2021, Virtual Event, USA.* ACM, New York, NY, USA, 13 pages. https://doi.org/10.1145/3452296.3472918

1 INTRODUCTION

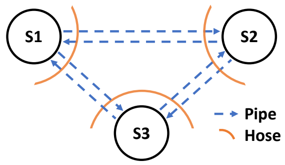
Global online service providers, such as Google, Facebook, and Amazon, build wide-area backbone networks for connecting thousands of Point-of-Presence (PoP) sites and hundreds of Data Centers (DCs) across continents. To keep up with the explosive traffic growth, tremendous amounts of money and engineering effort are constantly invested in expanding and upgrading the backbone network. Network planning is thus the key to the backbone evolvement, with the ultimate goal of devising *capacity-efficient* network build plans that are *resilient to unforeseen demand uncertainties* from service changes and traffic dynamics.

Facebook achieves this goal by innovatively adopting the Hose model in backbone planning. Traditionally, backbone planning was based on the Pipe model. As illustrated in Figure 1, the Pipe model abstracts pairwise traffic demands between network sites [16]. To provision sufficient capacity across demand variations, with the Pipe model, we must plan for the peak demand between every site pair. From the entire network's perspective, this approach aims at accommodating the "sum of peak" traffic regarding all the connected sites. The Hose model, in contrast, abstracts the aggregated ingress and egress traffic demands per site [9, 13]. It naturally sums up the traffic demands across sites, so capacity planning with the Hose model is for the "peak of sum" traffic. As the peak traffic demands across different sites are unlikely to happen simultaneously, the Hose model offers *multiplexing gain*, which saves the total capacity and leaves headroom for traffic uncertainties after deployment where individual demands across sites vary but their sum does not exceed the provisioned peak capacity.

Besides capacity saving and resilience to uncertainty, Hose-based backbone planning goes hand in hand with the industry trend of decoupling service logic from infrastructure design. In practice, services are migrated from one DC to another for various reasons, e.g., load balancing, service scaling, latency reduction, DC maintenance, etc. The network and server infrastructure should mark out the service behaviors and provide flexibility for service migration. This requirement makes accurate point-to-point traffic demand forecast between site pairs very difficult. In addition, for an actively growing backbone network like Facebook's, new DCs are built yearly, so it is almost impossible to estimate the traffic demand to/from new DCs yet to be built. Thanks to the Hose model, we only need to specify the aggregated traffic demand per site, without worrying about the other end of each traffic flow. Therefore, using Hose-based planning, the network has the potential to scale up per-node basis, as easily as storage and compute resources, in the future.

However, regardless of the advantages of Hose-based network planning, the capacity must be granted to site pairs in a point-to-point manner like in the Pipe model. Our problem with backbone planning is thus to convert the Hose per-site traffic into the Pipe pairwise traffic. The Hose model, which was originally invented for Virtual Private Network (VPN) provisioning [9] and later used for Virtual Machine (VM) placement [4] in the cloud, has never been applied to the network planning setting, so we cannot turn to the literature for readily available solutions.

Our main contribution in this paper is the solution to this new problem. The Pipe output traffic can be presented as a traffic matrix



$$peak(S1 \rightarrow S2) = 2\,Tbps\ (at\ 9am)$$
$$peak(S1 \rightarrow S3) = 3\,Tbps\ (at\ 3pm)$$
$$peak\left(\sum S1_{egress}\right) = 4\,Tbps\ (all\ day)$$
$$Pipe{:}\ sum\ of\ peak = 5\,Tbps$$
$$Hose{:}\ peak\ of\ sum = 4\,Tbps$$
$$Hose\ multiplexing\ gain = 1\,Tbps$$

Figure 1: Hose model for aggregated ingress and egress traffic demands per site vs. Pipe model for individual traffic demands between node pairs on a 3-site network. A site can be a DC or PoP. Pipe plans for "sum of peak" traffic, whereas Hose plans for "peak of sum" traffic, which offers *multiplexing gain* as individual peak traffic demands usually happen at different times.

(TM) between the site pairs. The aggregated traffic demands in Hose map to a continuous space, which contains an infinite number of TMs. It is computationally intractable to plan for all possible Pipe TMs under the Hose model. Our challenge is to generate a small subset of TMs to represent the Hose space. We propose a series of heuristic algorithms to address this challenge (§4). We first design a sampling scheme to generate candidate TMs uniformly in the Hose space. From these TMs, we find critical ones that stress the current bottleneck links, which are potential locations to deploy additional capacity. We thus propose a sweeping algorithm to quickly find bottleneck links in the network. Critical TMs are chosen through optimization, and we also define "Hose coverage" as a metric to quantify how representative these chosen TMs are.

Another contribution of this paper is to share the production network planning process, with practical considerations in Facebook's network setting. Our engineering experience includes the separation of short-term and long-term planning, the abstraction to simplify the interaction between the optical and IP networks, the resilience policy to protect against failures, and the optical-IP cross-layer capacity optimization (§5). We also evaluate the performance of our Hose-based network planning system in production (§6). We demonstrate Hose can save 17.4% capacity compared to Pipe and drops up to 75% less traffic under unplanned failures.

To the best of our knowledge, we are the first to study the Hose model in the context of network planning, and this is the first time that the end-to-end network planning procedure is introduced to academia. We wish our work to inspire a new line of research, where theoreticians can have a better formulation of our heuristic algorithms and practitioners can optimize our planning system. This work does not raise any ethical issues. We preserved user privacy and anonymity throughout this study.

2 MOTIVATION FOR HOSE

In this section, we use production traffic to demonstrate the advantages of Hose-based backbone planning on capacity saving and resilience to traffic uncertainties.

Experimental setup We collect production traffic between every site pair on the Facebook North America backbone from 11/23/2020 to 12/28/2020. To eliminate the time-of-day effect, we only look at the busy hour, when the total traffic in the backbone is the highest in the day. In the busy hour, traffic is sampled once per minute, making 60 data points. For the Pipe model, we get the 90th percentile across the 60 data points as the peak traffic demand for each site pair. For the Hose model, we add up the ingress/egress traffic per site for each data point across the source/destination sites it talks to. Among the 60 data points of aggregated traffic, we use the 90th percentile as the peak Hose traffic demand. This method gives us the "daily peak" traffic demands for the Hose and Pipe models respectively.

In production, we usually smooth traffic demands with a moving average. By Facebook's standard, we take a 21-day window to average the daily peak demands described above, and we add 3× the standard deviation of the 21-day data to the moving average as a buffer for sudden traffic spikes. This method produces the "average peak" traffic demand per Hose site and per Pipe site pair.

In the following experiments, we sum up the total traffic demand in the entire North America backbone, across sites in Hose and across site pairs in Pipe. We look at 4 numbers per day: the total daily peak demand and the total average peak demand in the backbone, under the Hose and Pipe models respectively.

Traffic reduction The key difference between Hose- and Pipe-based planning is to deploy capacity for "peak of sum" vs. "sum of peak" traffic. If using the Hose model, the *multiplexing gain* allows us to plan for less capacity, as the Pipe traffic sharing the same source/sink are unlikely to reach the peak simultaneously. Figure 2 shows the relative Hose traffic reduction, as the reduced total demand in Hose against Pipe divided by the total demand in Pipe. The "daily peak" demand of Hose (red dashed curve) is 10%-15% lower than Pipe, and the "average peak" demand (black solid curve) is 20%-25% lower. As backbone planning is based on traffic demands, we have good reasons to believe a considerable proportion of capacity can be saved just by adopting the Hose model for planning.

Tolerance to traffic dynamics The multiplexing effect also means the Hose planning result can cover more traffic variations. Figure 3 is the CDF of the total daily peak traffic demand. For confidentiality, we normalize the absolute traffic volume against the maximum demand (which is from the Pipe model). As shown in the figure, the vertical line at $x = 0.55$ maps to 90% of the days in the Hose model and 40% in Pipe. It means if we plan for 55% of the maximum total demand, under the Hose model, the daily peak demand will be satisfied for 90% of the days, while it will be satisfied for only 40% of the days in Pipe. The higher percentile in Hose indicates it can tolerate more traffic uncertainties. Since the Hose model is constrained by the aggregated traffic instead of a particular TM, it has more headroom to absorb unexpected traffic spikes.

Stable traffic demand We also measure the variance of Hose and Pipe traffic across days. To make the different traffic demands comparable, we use coefficient of variation as the metric, which is the standard deviation of the traffic demand divided by the mean. Figure 4 shows the coefficient of variation for the total daily peak traffic in the backbone. The relative traffic dispersion in Hose is much smaller than Pipe, with a shorter tail as well. As a result, the Hose model provides a more stable signal for planning and simplifies traffic forecast. With these, it is not hard to envision the network scaling up as easily as storage and compute resources, where a node can have an accurate approximation of its future growth, without worrying about the interaction with other nodes in the network.

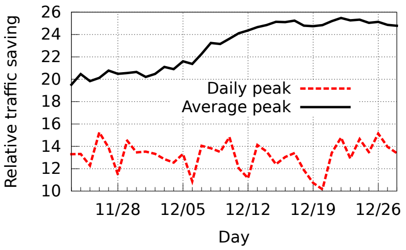


Figure 2: Hose traffic reduction.

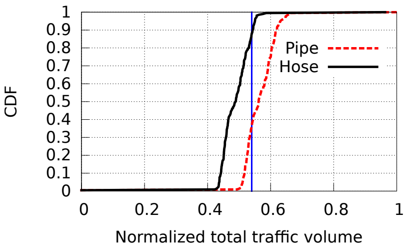


Figure 3: Total traffic distribution of Hose vs. Pipe.

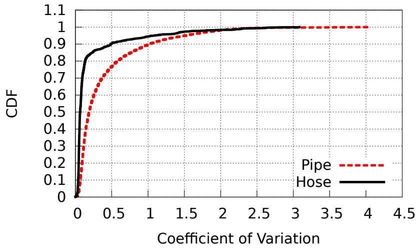


Figure 4: Coefficient of Variation with Pipe vs. Hose traffic.

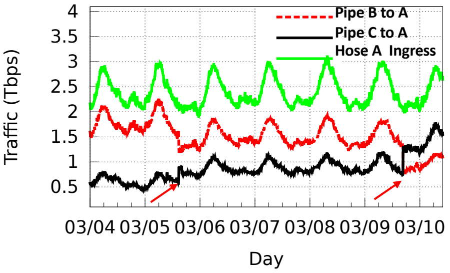


Figure 5: Service traffic from DC regions B and C to A.

Adaption to service evolvement Services evolve over time in production. Possible causes include service behavior changes, re-labeling of Quality of Service (QoS) classes, traffic shift for load balancing, new service launches, and many others. Figure 5 shows an example from the user database (UDB) service at Facebook. Due to resource and operational constraints, the UDB servers storing user data only sit in a few regions, and UDB-less regions rely on a caching service called Tao [2] to fetch data from UDB regions nearby. Figure 5 plots the amount of Tao traffic flowing from UDB regions B and C to UDB-less region A. The significant traffic change is a result of Tao service changing the primary UDB region from B to C, with a canary on a few shards on 03/05 and a complete policy change on 03/09. Both incidents created several Tbps of traffic shifts, where a Pipe model would fail. In contrast, because the total traffic amount stayed the same, the Hose ingress traffic at region A had little disruption. The traffic aggregation nature of Hose is naturally more resilient to service changes, making it a future-proof solution to network planning.

3 HOSE-BASED CAPACITY PLANNING

In this section, we give an overview of the capacity planning problem and our system design. Table 1 lists the notations throughout the paper.

Network model Our backbone network connects a number of DCs and PoPs together. It consists of IP routers over a Dense Wavelength Division Multiplexing (DWDM) optical network. The backbone routers are connected using IP links that route over multiple fiber segments. We represent this network as a two-layer graph: the IP network $G = (V, E)$, where the vertices V are backbone routers and the edges E are IP links, and the optical network $G' = (V', E')$, where the vertices V' are Optical Add-Drop Multiplexers (OADMs) and the edges E' are fiber segments.

For each IP link $e \in E$, $FS(e)$ is the set of fiber segments that e rides over, which form a path on the optical topology. The IP link e consumes a portion of spectrum on each fiber segment $l \in E'$ over which e is realized. For example, a 100Gbps IP link realized using Quadrature Phase Shift Keying (QPSK) modulation can consume 50GHz of spectrum over all fiber segments in its path. The relationship between IP capacity and optical spectrum is shown in Section 5.1.

Failure model We consider a set of fiber failures in the backbone. Every IP link $e \in E$ over the failed fibers would be down. In order to provide desired reliability to the service traffic, we pre-define a set of failures R referred to as planned failures. The production network should be planned with sufficient capacity such that all service traffic can be routed for each failure $r \in R$. Detailed resilience policy in capacity planning will be presented in Section 5.2.

Traffic forecast Capacity planning depends on the projected traffic demand in the future. Instead of modeling the organic growth of link-wise traffic like done in ISP networks, for content providers, it is common practice to forecast the future traffic demand per service based on service profiling. This is because services, as content generators, provide a more reliable source of truth for traffic demand. For inter-DC traffic, service teams calibrate server utilization, especially CPU utilization, to devise service growth plans under the server budget allocated by the company. They provide service scaling factors, which are applied to the current service traffic to form the future demands. For PoP-DC traffic, we model user growth and cache misses at PoPs to predict the amount of content retrieval between PoPs and different DCs. The demands can be aggregated in different ways, e.g., per-site-pair basis for traditional Pipe-based planning and per-site basis for Hose-based planning.

Problem statement Network capacity is the maximum throughput (in Gbps, Tbps, or Pbps) the IP network, and individual IP links, can carry. The problem of *Capacity Planning* is to compute the desired network capacity to be built in the future. Building a network involves complex steps:

(1) Procure fibers from third-party providers
(2) Build terrestrial and submarine fiber routes
(3) Pull fibers on existing ducts
(4) Install line system to light up the fibers
(5) Secure space and power at optical amplifiers and sites
(6) Procure, deliver, install hardware (optical and IP) at sites

Table 1: Notations

Symbol	Definition
$G = (V, E)$	The IP topology with backbone routers and IP links
$G' = (V', E')$	The optical topology with OADMs and fiber segments
$FS(e)$	The set of fiber segments which IP link e goes through
N	The number of sites (DCs and PoPs combined) in the backbone
M	A $N \times N$ Traffic Matrix (TM)
$m_{i,j}$	The traffic volume from site i to site j in M
$\vec{u_s}$	A $1 \times N$ all-ones vector to retrieve source nodes in M
$\vec{u'_d}$	A $N \times 1$ all-ones vector to retrieve destination nodes in M
$\vec{h_s}$	A $1 \times N$ vector bounding egress traffic of source nodes in M
$\vec{h'_d}$	A $N \times 1$ vector bounding ingress traffic of destination nodes in M
$H = \{\vec{h_s}, \vec{h'_d}\}$	Hose constraints for the egress and ingress traffic demands
α	Edge threshold in the sweeping algorithm (§ 4.2)
ϵ	Flow slack in Dominating Traffic Matrix (DTM) selection (§ 4.3)
$c \in C$	A network cut in the cut set
$D(c)$	The set of DTMs for a network cut c under flow slack ϵ
T	A set of candidate DTMs
A_M	A binary 0-1 assignment variable indicating if DTM M is selected
P	A convex polytope to represent the high-dimensional Hose space
S	A set of sample points in the Hose space P
$b \in B$	A plane in a collection of planes in the Hose space P
$x(l)$	The cost of procuring and deploying a fiber segment $l \in E'$
$y(l)$	The cost of turning up a dark fiber $l \in E'$
$z(e)$	The cost of provisioning a new wavelength to add an IP link $e \in E$
$\varphi(e)$	The spectral efficiency of an IP link $e \in E$
λ_e	The IP capacity of IP link $e \in E$
γ	Routing overhead
$r_q \in R_q$	A failure scenario in the planned failure set for QoS class q
$f_{i,j}(u, v)$	A traffic flow from source i to destination j via IP link $\{u, v\} \in E$
ϕ_l	The number of fibers to be lighted up on fiber segment $l \in E'$
ψ_l	The number of fibers to be deployed on fiber segment $l \in E'$

All these activities have high lead time, taking months or even years to deliver. Thus, capacity planning is critical to the future evolution and profitability of the network.

In the network planning problem, the objective is to dimension the network for the forecast traffic under the planned failure set R by minimizing the total cost of solution. The cost of the network is calculated based on a weighted function of equipment (fibers and other optical and IP hardware) procurement, deployment, and maintenance to realize the network plan. The specific cost model is introduced in Section 5.1.

Planning schemes At Facebook, we categorize capacity planning into two sub-problems: *short-term planning* and *long-term planning*. Short-term planning outputs the *exact IP topology*, i.e., the IP links and the capacity on each link, while long-term planning only determines the fibers and hardware to procure. This design decision is based on the fact that network building is an iterative process and long-term planning only serves as a reference most times. For example, the fiber procurement plan may change at deployment time according to availability of fiber resources on the market. Short-term planning is conducted only after fiber and hardware are secured and in place, because turning up capacity can happen at a short notice.

Planning pipeline Figure 6 illustrates the planning process. Backbone network planning starts from traffic forecast. As aforementioned, our traffic forecast is service-based and independent of the planning method, i.e., Pipe- and Hose-based planning alike. For Hose-based planning, we aggregate the service demands with respective to each backbone site to generate the ingress and egress Hose constraints. As motivated in the introduction, the key to Hose-based network planning is converting the Hose constraints into Pipe TMs. Thus, as will be shown in Section 4, the planner takes judicious steps to narrow down the infinite number of possible Pipe TMs to a small set of representative ones. Short-term and long-term planning are then applied to the reference TMs with different optimization formulations, considering various failure scenarios under the resilience policy. The optimization procedure is detailed in Section 5.

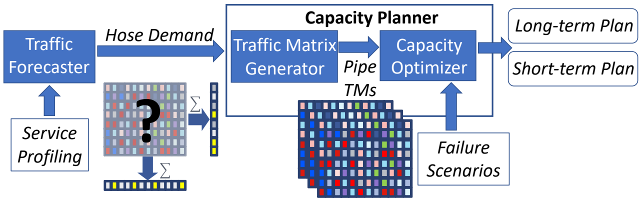


Figure 6: System Architecture

The output of planning is Plan Of Record (POR), in the format of capacity between site pairs. The POR from short-term planning is handed to the capacity engineering team for capacity turn-up, and the POR from long-term planning is given to the fiber sourcing team for fiber procurement and to the optical design and IP design teams for deployment of fibers and optical line systems. The focus of this paper is on the design of Capacity Planner.

4 TRAFFIC MATRIX GENERATION

In this section, we introduce specific steps of converting Hose constraints into reference TMs for planning, which includes heuristic algorithms, optimization, and performance metrics.

4.1 Traffic Matrix Sampling

A Traffic Matrix (TM) for a N-node network topology is a $N \times N$ matrix M, where each coefficient $m_{i,j}$ represents the traffic demand of a flow (typically in Gbps in practice) from the source node i to the destination node j. The flow traffic demand must be non-negative, and a node does not generate traffic to itself. Hence, the coefficients are in $\mathbb{R}_+$ and all diagonal coefficients are zero.

A valid TM must satisfy the following Hose constraints, where $\vec{u_s}$ and $\vec{u'_d}$ are the $1 \times N$ and $N \times 1$ all-ones column and row vectors, and the corresponding demand vectors $\vec{h_s}$ and $\vec{h'_d}$ bound the total egress and ingress traffic amount at the source and destination nodes. These constraints form a convex polytope in the $N^2 - N$ dimension space, where each non-zero coefficient in the TM is a variable. Figure 7 illustrates a highly simplified 3D example with variables $m_{1,2}$, $m_{1,3}$, and $m_{1,4}$ only. Each valid TM is a point in the polytope space, and there are an infinite number of valid TMs in this continuous space.

$$\text{Hose constraints:} \quad \begin{matrix} \vec{u_s} \cdot M \leqslant \vec{h_s} \\ M \cdot \vec{u'_d} \leqslant \vec{h'_d} \end{matrix} \tag{1}$$

To generate TMs that satisfy the Hose constraints, our first step is to sample the polytope space uniformly. Algorithm 1 shows our two-phase algorithm for generating one sample TM. We randomly

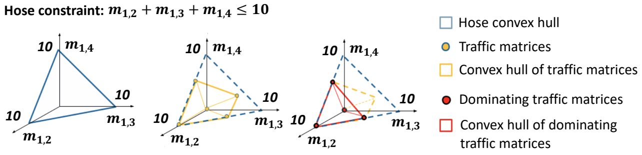

Hose constraint: $m_{1,2} + m_{1,3} + m_{1,4} \leq 10$
10
$m_{1,4}$
10
$m_{1,3}$
10
$m_{1,2}$
10
$m_{1,4}$
10
$m_{1,3}$
10
$m_{1,2}$
10
$m_{1,4}$
10
$m_{1,3}$
10
$m_{1,2}$
□ Hose convex hull
● Traffic matrices
□ Convex hull of traffic matrices
● Dominating traffic matrices
□ Convex hull of dominating traffic matrices

Figure 7: A 3D example of the Hose polytope space.

Algorithm 1 *sampleTM*( )

```
Input: network size $N$, Hose constraints $H = \{\vec{h_s}, \vec{h'_d}\}$ in Formula (1)
Output: a random $N \times N$ traffic matrix $M$ satisfying $H$
 1: $M = \mathbf{0}_{N\times N}$
 2: for every $m_{i,j}$ in $M$ in random order do
 3:     $h = Min(h_i, h'_j) \times random.uniform(0, 1)$
 4:     $m_{i,j} = h$
 5:     $h_i = h_i - h$
 6:     $h'_j = h'_j - h$
 7: end for
 8: for every $m_{i,j}$ in $M$ in random order do
 9:     $h = Min(h_i, h'_j)$
10:     $m_{i,j} = m_{i,j} + h$
11:     $h_i = h_i - h$
12:     $h'_j = h'_j - h$
13: end for
```

create a valid TM in the polytope space in Phase 1 (*lines 1-7*) and stretch it to the polytope surfaces in Phase 2 (*lines 8-13*), under the intuition that TMs on the surfaces have higher traffic demands and translate to higher capacity requirements for network planning.

In Phase 1, we initialize the TM to a zero matrix (*line 1*) and assign traffic to the TM entries one by one in a random order (*line 2*). For every entry $m_{i,j}$, the maximal allowed traffic amount is the lesser of the two Hose constraints for source i and destination j. We give it a uniformly random scaling factor between 0 and 1 (*line 3*) and assign the product to the entry (*line 4*). For bookkeeping, the consumed traffic amount is deducted from the Hose constraints (*lines 5-6*). In Phase 2, we add residual traffic to the TM to exhaust as many Hose constraints as possible. Similar to Phase 1, we iterate through the entries in a random order (*line 8*) and add the maximal allowed traffic amount to each entry (*lines 9-12*). Because we iterate through all the entries and always consume the maximal traffic, our Phase 2 guarantees to exhaust the most Hose constraints from the Phase 1 result. It also guarantees we cannot have egress and ingress hose constraints simultaneously unsatisfied (remaining constraints must be all egress or all ingress), because if that were the case, the algorithm would simply increase the associated source-destination flows until either ingress or egress constraints are exhausted.

This sampling algorithm is highly effective regardless of the simplicity. As will be shown in Figure 9a, over 97% of the Hose polytope space is covered with 10^5 sample TMs. The effectiveness comes from the high randomness: (1) we apply different permutations of the TM entries (*line 2 and line 8*) in each run to distribute the Hose traffic budget in different ways; (2) we use a scaling factor (*line 3*) to adjust the assignable traffic randomly according to the uniform distribution. Our two-phase sample-then-stretch approach is proven to be critical. In a former solution, we directly sample the polytope surfaces uniformly, but the coverage is 20%-30% lower with the same number of samples.

4.2 Bottleneck Links Sweeping

It is computationally infeasible to consider the enormous number of TM samples. Fortunately, TMs have different importance for network planning. As the goal of network planning is to add capacity to "bottleneck links" in the network, TMs with high traffic demands over the bottleneck links play a dominating role. We call such TMs *Dominating Traffic Matrices* (DTMs), and we aim to find a small number of DTMs such that designing the network explicitly for them has a high probability to satisfy the remaining TMs as well.

From the graph theory's perspective, bottleneck links are captured by the network cuts that partition the nodes into two disjoint subsets. However, the number of network cuts is exponential to the network size. A production backbone network has tens to a few hundred nodes, thus enumerating all the cuts is intractable. Even if a backbone network is not a densely connected graph, the number of possible cuts is still $O(2^{min(|V|,|E|)})$, where $|V|$ and $|E|$ are the number of nodes and edges respectively. We propose a sweeping algorithm to quickly sample the network cuts, and the sweeping process is illustrated in Figure 8.

The sweeping algorithm has a hyperparameter *edge threshold* α chosen in the $[0, 1]$ interval. The network nodes are represented by their latitude and longitude coordinates. We draw the smallest rectangle inscribing all the nodes and radar-sweep the graph centering at points on the rectangle sides. There are k equal-interval points per side and the sweeping is performed at discrete orientation angles of interval β. We typically choose $k = 1000$ and $\beta = 1^{\circ}$. The algorithm draws a reference cut line at each sweeping step, which splits the nodes into the following three mutually exclusive categories.

- Edge nodes, whose distance to the cut line over the distance of the farthest node in the network to the cut line is smaller than α.
- Above nodes, which are above the cut line but are not in the edge nodes group.
- Below nodes, which are below the cut line but are not in the edge nodes group.

Network cuts are all possible bipartite splits of the edge nodes combined with the above and below nodes respectively. In this algorithm, parameters k and β define the sampling granularity, and the edge threshold α regulates the number of cuts considered per sampling step. As α increases, we are able to generate an increasingly large number of network cuts. In particular, setting α to 1 guarantees that we enumerate all partitions of the network. The relationship between α and network cuts is shown in Figure 9b.

4.3 Selection of Dominating Traffic Matrices

The formal definition of DTM with respect to network cuts is as below. Intuitively, with the TMs sampled in Section 4.1 and network cuts generated in Section 4.2, we want to find the TM that produces the most traffic for every network cut.

Definition 4.1 (Dominating Traffic Matrix - Strict Version). The dominating traffic matrix of a network cut is the traffic matrix in all the sampled traffic matrices that has the highest traffic amount across the cut.

This definition yields as many DTMs as there are network cuts. To further reduce the number of TMs involved in our planning

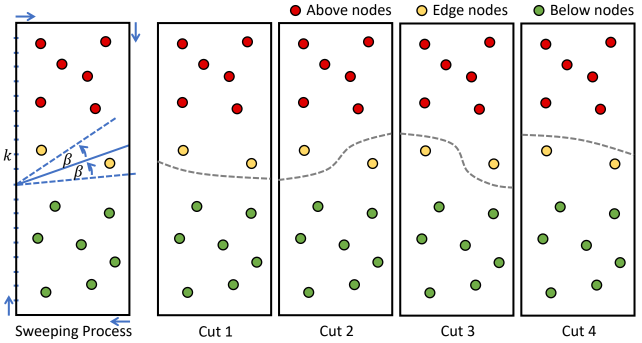


Figure 8: An example of the sweeping algorithm. The sweeping centers around k points per rectangle side and moves in β° steps. The reference cut (blue solid line) sweeping step creates 2 edge nodes (yellow dots), whose permutations form 4 cuts.

computation, we get inspiration from the *minimum set cover problem* [12]: if we slack the DTM definition from the most traffic-heavy TM per network cut to a set of relatively traffic-heavy TMs within a bound to the maximum, the sets of DTMs for different cuts are likely to overlap and the cuts may be represented by a smaller number of overlapping DTMs. We thus introduce the *flow slack* ϵ and define the slack version of DTM as below. For the rest of the paper, all DTMs refer to this slack definition.

Definition 4.2 (Dominating Traffic Matrix - Slack Version). A dominating traffic matrix of a network cut with flow slack ϵ is a traffic matrix from the sampled traffic matrices whose traffic amount across the cut is no smaller than $1-\epsilon$ of the maximum among all the sampled traffic matrices, where ϵ is a small value in $[0,1]$.

In our formulation of the minimum set cover problem, the universe is the ensemble of network cuts C. For every cut $c \in C$, we get the set of DTMs $D(c)$ under the given flow slack ϵ according to Definition 4.2. Combining them, we have a collection $T = \{M\}$ of all the candidate DTMs, where each DTM belongs to a subset of cuts in C. For example, a DTM M may be generated by multiple cuts $\{c_i, c_j, c_k\}$ at the same time. Our goal is to find the minimal number of DTMs to cover all the cuts in C.

We solve this minimum set cover problem by Integer Linear Programming (ILP). As shown below, we define a binary assignment variable A_M, which is set to 1 if a candidate DTM M is selected in the end and set to 0 otherwise. The assignment variables must guarantee each network cut is represented by at least one of its candidate DTMs, and we minimize the number of selected DTMs by minimizing the sum of the assignment variables.

$$\begin{aligned} \min \quad & \sum_{M \in T} A_M \\ \text{s.t.} \quad & \sum_{M \in D(c)} A_M \geq 1, \forall c \in C \\ & A_M \in \{0,1\}, \forall M \in T \end{aligned} \tag{2}$$

We achieve a low DTM count with the commercial ILP solver FICO Xpress [1]. As will be shown in Figure 9c, a flow slack of approximately 1% can reduce the number of DTMs by over 75%, a substantial gain in the computation needed for capacity planning. A further increase in the flow slack results in even more impressive results, though at the price of a lower Hose coverage, as we will see in the next section.

4.4 Hose Coverage

As we perform Hose-compliant capacity planning, we need to define a metric to evaluate the degree to which our generated reference TMs cover the entire Hose space. In particular, since we use a two-stage process, where we sample the Hose space using a large number of TMs and further down-sample them to reach a smaller number of DTMs, it is desirable to measure the Hose coverage for each stage of the process.

Recall that the Hose is represented by a convex polytope P in a high-dimensional vector space, a natural way to measure the coverage of a set of samples S would be by volume, namely the volume of the convex hull containing all the samples divided by the volume of the Hose space as follows. This metric is illustrated in Figure 7 in three dimensions.

$$\text{Coverage}(S,P) = \frac{\text{Volume}(\text{ConvexHull}(S))}{\text{Volume}(P)} \tag{3}$$

When applying to practical instances of network planning, however, this metric is intractable. The complexity of computing a convex hull for V points in a L-dimensional space is approximately $O(V^{\frac{L}{2}})$ [6]. In our case, $V = N^2 - N$ where N is the node count in the network, which can be a few hundred, and the sample size $V = |S|$ can be 10^5.

Instead, we define the planar coverage of the Hose space P by a set of samples S on a plane b as follows, where $\Pi(S,b)$ marks the projection of the samples in S on the plane b, and $\Pi(P,b)$ is the projection of the Hose polytope P on b.

$$\text{PlanarCoverage}(S,P,b) = \frac{\text{Area}(\Pi(S,b))}{\text{Area}(\Pi(P,b))} \tag{4}$$

For a collection of planes B, we define the coverage of the Hose space P by a set of samples S to be the mean planar coverage of P by S across all the planes in B.

$$\overline{\text{Coverage}(S,P)} = \frac{1}{n}\sum_{i=1}^{n} \text{PlanarCoverage}(S,P,b_i) \tag{5}$$

The choice of these planes is critical for picturing the high-dimensional Hose space truthfully. These planes should characterize all the variables in the Hose constraints, and the variables should contribute equally to shaping the planes. Conveniently, we construct planes with all the pairwise combinations of the variables in the Hose constraints. Recall from Formula (1) that each variable is an off-diagonal coefficient of a valid TM M, or a source-destination pair in the network. In the Figure 7 example, the chosen planes are $B = \{\text{Plane}(m_{1,2}, m_{1,3}), \text{Plane}(m_{1,2}, m_{1,4}), \text{Plane}(m_{1,3}, m_{1,4})\}$.

5 CROSS-LAYER OPTIMIZATION

Capacity planning requires cross-layer optimization of the optical network and the IP network. The optimization inputs include the DTMs, the IP topology $G = (V,E)$ with backbone routers V and IP links E, and the optical topology $G' = (V',E')$ involving the OADMs V' and fiber segments E'. The outputs are the target IP and optical topologies $G+\Delta G = (V, E+\Delta E)$ and $G'+\Delta G' = (V', E'+\Delta E')$

with the same sites but more links or greater capacity. This section presents the optimization process in detail.

5.1 Cost Model

Although planning is not a time-critical mission, given the size of our network, we want the optimization to at least finish, hopefully in hours. To simplify the optimization, we devise a cost model to abstract complications in the optical and routing systems as simple cost factors multiplied to the decision variables. The five essential cost factors are:

Fiber procurement and deployment cost This is the entire cost of purchasing and installing a new fiber before it becomes usable. If we own the fiber, it includes the equipment cost of procuring the fiber, optical amplifiers, Configurable Optical Add/Drop Multiplexers (COADMs), Wavelength Selective Switches (WSSes), IP router chassis, as well as the labor cost of cleaning the fiber, deploying the amplifiers along the fiber path and deploying COADMs, WSSes, and router chassis at the terminal sites. If we lease the fiber, it covers all the usage, operational, and maintenance cost in the leasing contract. This cost varies fiber to fiber depending on the vendor, fiber length, fiber type (terrestrial, submarine, or aerial), etc., and we model it based on these features. We denote this cost as $x(l)$ for fiber segment l on the optical topology G'.

Fiber turn-up cost This is the cost of turning up a dark fiber that is already installed. It includes the cost of purchasing extra equipment such as transponders and line cards and the manual effort of configuring devices. We estimate this cost based on historical data. It is denoted as $y(l)$ for fiber segment l on G'.

Capacity addition cost This is the cost of provisioning a new wavelength on a turned-up fiber. It adds one unit of bandwidth capacity, i.e., 100Gbps, on the IP layer. This cost involves the labor work of wavelength provisioning and router port configuration. It is a flat cost, denoted as $z(e)$ for IP link e on the IP topology G.

Spectral efficiency This factor captures the proportion of optical spectrum a unit of IP capacity consumes over all fiber segments on its path, which depends on the modulation required to get error-free transmission on the circuit. We denote the spectral efficiency of an IP link e as $\varphi(e)$ and delegate the sophisticated optical link engineering calculations to an optical link simulator similar to [21]. The following spectral conservation constraint regulates the spectral consumption per fiber segment $l \in E'$. Assume l has ϕ_l lighted-up fibers, each having a maximum allowable spectrum $MaxSpec(l)$. For an IP link $e \in E$, the required spectrum is the IP capacity λ_e multiplied by its spectral efficiency $\varphi(e)$. Thus, the total spectrum consumed over fiber segment l must be greater than or at least equal to the sum of spectrum required by each IP link e riding over this fiber segment, specified by the IP-optical mapping function $FS(e)$. To account for the loss of usable spectrum due to the spectrum continuity constraint [3], we reserve a percentage of $MaxSpec(l)$ as a planning buffer while turning up fibers. This abstraction of wavelength contention saves the effort of accurate wavelength allocation and works well in practice.

$$SpecConserv(G, G'):$$
$$\sum_{e \in E,\, l \in FS(e),} \varphi(e) \times \lambda_e \leq MaxSpec(l) \times \phi_l, \forall l \in E' \tag{6}$$

Routing overhead This is the loss of bandwidth capacity due to imperfection of routing algorithms. We formulate capacity planning as a multi-commodity flow problem [11] on the IP layer. In practice, backbone routers only allow for a small number of parallel paths per flow, such as in Equal-Cost Multi-Path (ECMP) and K-shortest path routing, which makes the problem NP-hard. To solve it in polynomial time, we switch to fractional flows, i.e., every flow being infinitely splittable, and we capture the difference from the actual routing algorithm by routing overhead. For a particular routing algorithm, the routing overhead γ is a $[1, +\infty)$ factor multiplied to the original traffic demand to give headroom for routing inefficiency.

5.2 Resilience Policy

Our services are categorized into several QoS classes for different performance guarantees. Different QoS classes have different resilience policies. Higher QoS classes (usually denoted by smaller class numbers) can tolerate more failures, through more robust routing algorithms and greater protection capacity in backup paths. Based on the resilience policy, each QoS class has a pre-defined set of failure scenarios to protect against. A failure scenario presents the physical-layer fiber cuts and the loss of IP links on these fibers.

With Hose-based capacity planning, we need to fully satisfy the traffic demand of each QoS class under the protected failures. As Equation (7) shows below, for QoS class q, we have a set of post-failure residual IP topologies G_q, whose elements are formed by removing the failed IP links of a particular failure scenario r_q in the scenario set R_q.

$$G_q = \bigcup_{r_q \in R_q} (G_0 - r_q) \tag{7}$$

As described in Section 3, we forecast traffic for individual service types. Aggregating across services, we have a Hose model H_q per QoS class $q \in \{QoS\}$. We design resilience policies in such a way that traffic from one QoS class is protected against failure scenarios from its own class and all other classes lower than it. Hence, the residual topology G_q must carry traffic of its own class and all higher classes. Per Section 5.1, each QoS class may use a different routing scheme, thus having a different routing overhead. Like shown in the equation below, the reference DTMs of a QoS class q is derived from the TM generation in Section 4 over all the protected traffic, as the union of the Hose constraints in classes 1 to q, with the routing overhead applied.

$$T_q = DTM\left(\bigcup_{i=1}^{q} \gamma(i) \times H_i\right) \tag{8}$$

For each QoS class q, given the DTMs T_q and post-failure IP topologies G_q, the traffic flows in each reference TM $M \in T_q$ must satisfy the conservation constraints on every topology $G \in G_q$, as shown below. That is, for every flow in a TM M, the source and sink of the flow have the required traffic amount, all intermediate nodes of the flow have zero traffic in sum, and the flows over an IP link cannot exceed the bandwidth capacity λ. Here we simply assume all flows are infinitely splittable, because the difference from the actual routing algorithms is accounted for by the routing overhead.

$$
\begin{aligned}
&\mathit{FlowConserv}(M, G) \text{ for } M \in T_q, G \in G_q\text{:}\\
&\quad \sum_{\{i,u\}\in E} f_{i,j}(i,u) - \sum_{\{i,u\}\in E} f_{i,j}(u,i) = m_{i,j}\\
&\quad \sum_{\{j,u\}\in E} f_{i,j}(u,j) - \sum_{\{j,u\}\in E} f_{i,j}(j,u) = m_{i,j}\\
&\quad \sum_{\substack{\{u,v\}\in E,\\ u\neq i, v\neq j}} f_{i,j}(u,v) - \sum_{\substack{\{u,v\}\in E,\\ u\neq i, v\neq j}} f_{i,j}(v,u) = 0\\
&\quad \sum_{\{u,v\}\in E} f_{i,j}(u,v) \leq \lambda_{u,v} \qquad\qquad \forall m_{i,j} \in M
\end{aligned}
\tag{9}
$$

5.3 Short-Term Planning

Short-term network planning is for the next 6 months to 2 years. In this period, we rely on the existing optical infrastructure. Thus, we assume the IP topology stays the same, yet the capacity of IP links can be increased. The physical-layer topology formed by active fiber segments can be expanded under the limit of deployed (maybe inactive) fiber resources. Our goal is to minimize cost while admitting the future traffic derived from Hose-based traffic forecast.

The ILP formulation is as follows. The optimization takes in the current IP topology G and the expandable optical topology $G' + \Delta G'$, where $\Delta G'$ is the expansion budget offered by the dark fibers. ϕ_l is the number of fibers on fiber segment $l \in E' + \Delta E'$ that will be lighted in the end, and λ_e is the target capacity on IP link $e \in E$. Multiplying them with the respective cost as described in Section 5.1, i.e., per-fiber turn-up cost $y(l)$ and per-unit-bandwidth capacity addition cost $z(e)$, we get the optimization objective of minimizing the total cost of building the final network.

$$
\begin{aligned}
\min \quad & \sum_{l\in E'+\Delta E'} y(l) \times \phi_l + \sum_{e\in E} z(e) \times \lambda_e\\
\text{s.t.} \quad & \mathit{SpecConserv}(G, G' + \Delta G')\\
& \bigcup_{M\in T_q, G\in G_q} \mathit{FlowConserv}(M, G), \forall q \in \mathit{QoS}\\
& \lambda_e \geq \Lambda_e, \forall e \in E\\
& \phi_l \geq \Phi_l, \forall l \in E' + \Delta E'
\end{aligned}
\tag{10}
$$

This objective is intrinsically equivalent to minimizing the additional cost of network expansion, because the sunk cost of building the existing network has been paid for, but it simplifies the constraints. For example, the spectral conservation constraint described in Section 5.1 is regarding the total IP capacity and total fiber counts. The flow conservation constraint in Section 5.2 should also be satisfied. Note that we need to consider this constraint for every QoS class. Besides, we have additional constraints that λ_e and ϕ_l must be greater than or equal to the current values Λ_e and Φ_l in the existing network, based on the fact that a network keeps growing: we do not reduce IP capacity or disable optical fibers once a network has been built.

5.4 Long-Term Planning

Long-term network planning targets at 2 to 5 years in the future. The purpose of long-term planning is to estimate the worst-case hardware requirements and make sure sufficient equipment is procured ahead of time. An important difference from short-term planning is long-term planning considers installation of new fibers. The large scale of our backbone network makes it infeasible to perform global search for all possible fiber installation locations. A practical solution is to narrow down to a small number of candidate locations based on fiber availability on the market and our operational experience. We sketch an optical topology $G' + \Delta G'$, with the candidate fibers in $\Delta G'$, and we map these fibers to possible IP links to form the IP topology $G + \Delta G$, where the potential IP links are in ΔG with zero initial capacity.

In this way, we convert the long-term planning problem to a similar formulation as the short-term planning problem. As shown below, the optimization objective is still minimizing the total cost, yet with one more term for the fiber procurement and deployment cost. On the candidate optical topology $\Delta G'$, ψ_l is the number of fibers to deploy on the fiber segment l and $x(l)$ is the per-fiber procurement and deployment cost defined in Section 5.1. The fiber turn-up cost and capacity addition cost are similar to short-term planning, but need to be considered on topologies $G' + \Delta G'$ and $G + \Delta G$ respectively with candidate fibers and IP links.

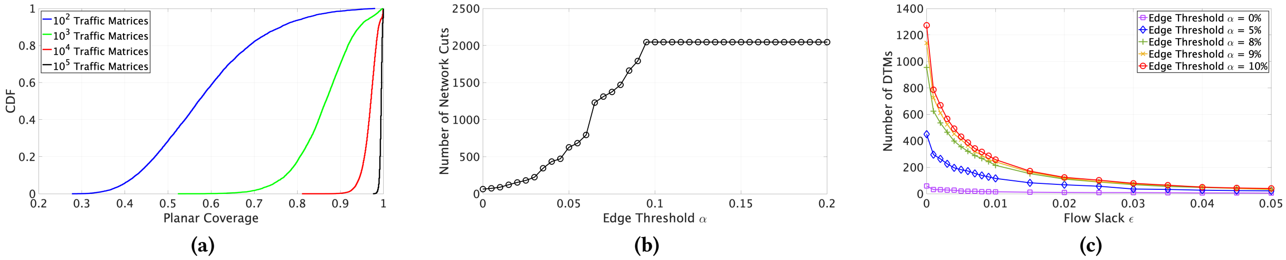
$$
\begin{aligned}
\min \quad & \sum_{l\in \Delta E'} x(l) \times \psi_l + \sum_{l\in E'+\Delta E'} y(l) \times \phi_l + \sum_{e\in E+\Delta E} z(e) \times \lambda_e\\
\text{s.t.} \quad & \mathit{SpecConserv}(G + \Delta G, G' + \Delta G')\\
& \bigcup_{M\in T_q, G\in G_q+\Delta G_q} \mathit{FlowConserv}(M, G), \forall q \in \mathit{QoS}\\
& \lambda_e \geq \Lambda_e, \forall e \in E + \Delta E\\
& \phi_l \geq \Phi_l, \forall l \in E' + \Delta E'\\
& \psi_l \geq 0, \forall l \in \Delta E'
\end{aligned}
\tag{11}
$$

Likewise, the spectral conservation constraint and flow conservation constraint also apply to the potential topologies $G' + \Delta G'$ and $G + \Delta G$. Although our approach results in a large number of possible IP links over the new fibers, the spectral conservation constraint guarantees to select a subset whose capacity can be fully accommodated by the fibers. Similar to short-term planning, the variables λ_e, ϕ_l, and ψ_l must increase relative to the base values, namely existing capacity numbers in the current network and zero for the candidate topologies. Since the fiber procurement and deployment cost is orders of magnitude higher than the fiber turn-up cost and capacity addition cost, our formulation naturally favors exhausting existing fiber resources first. In case the optimization fails to produce feasible solutions, we enlarge the pool of candidate fibers and rerun the optimization.

6 EVALUATION

Our Hose-based capacity planning system has been running in production for several years. Its core component is an optimization engine implemented on top of the Xpress solver [1] with a max-flow-based route simulator. It is a production-grade software with substantial engineering efforts put into scaling up the optimization. In this section, we first evaluate the Hose conformance of the TM generation process in Section 4 to give guidelines for parameter tuning in our system, then we compare the end-to-end planning results with Pipe-based planning to show the performance advantages.

All experiments are on Facebook's latest North America production topology, which contains hundreds of nodes and thousands of IP links over hundreds of optical fibers. We plan for 500 failure scenarios based on historical data, including 300 single-fiber failures and 200 multi-fiber failures. We predict future traffic with



(a)

(b)

(c)

Figure 9: (a) Distribution of planar Hose coverage by different numbers of sampled TMs, (b) Network cuts generated under different edge threshold α, and (c) The number of DTMs as a function of flow slack ϵ, for various edge threshold α values.

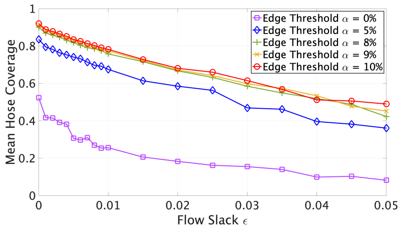


Figure 10: Average Hose coverage of DTMs as a function of the flow slack ϵ, for various edge threshold α values.

our production traffic forecast system, and our experiments strictly follow the two-step planning procedure in production: deciding the hardware infrastructure with long-term planning and feeding the result into short-term planning for the final IP network build plan. Traffic forecast and capacity planning for the Pipe model are based on our legacy systems before Hose was adopted.

6.1 Hose Conformance

Hose coverage of TM sampling The effectiveness of our TM sampling algorithm is shown in Figure 9a. Here, we present the CDF distribution of planar coverage, as defined in Section 4.4, for different sample sizes. With 10^5 TM samples, among all the projection planes for the Hose polytope, even the worst plane reaches over 97% coverage, and the mean coverage is over 99%. This result indicates that the Hose space can be represented by 10^5 sample TMs with negligible loss of accuracy.

Comparing different curves, intuitively, more TM samples result in higher Hose coverage. Yet, the increase of coverage slows down as the number of samples grows. For example, the mean coverage of 10^4 samples is 10% higher than 10^3 samples, while the increase from 10^4 to 10^5 samples is only 3%. This trend shows a rewarding tradeoff: we can reduce a large number of sample TMs at minimal degradation of Hose coverage. However, recall that our TM sampling algorithm (Algorithm 1) has $O(N^2)$ complexity with regard to the network size N, sampling 10^5 TMs takes only 200 seconds in practice. In our production, we choose 10^5 samples for highly accurate planning results.

Effect of edge threshold on cut generation Figure 9b looks into the performance of the sweeping algorithm in Section 4.2. It plots the number of generated network cuts with the variance of the edge threshold parameter α. Recall from the algorithm illustration in Figure 8 that α determines the number of edge nodes that are permuted to form different cuts, thus a larger α results in more network cuts and $\alpha = 1$ guarantees to find all cuts in the network. In practice, however, we do not need to set $\alpha = 1$ to get all the cuts. According to Figure 9b, the number of cuts reaches the maximum when $\alpha \geq 0.095$. The curve has a sharp increase for α between 0.065 and 0.07, indicating the algorithm can be sensitive with α. Based on these observations, we conclude a sufficiently large α should be chosen, otherwise a significant number of cuts may be ignored.

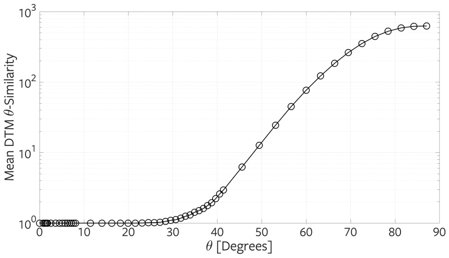
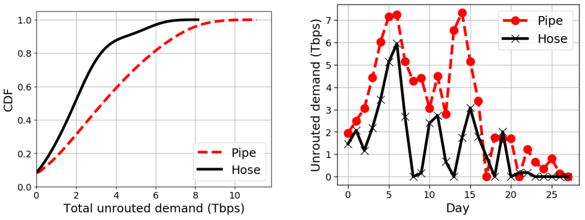


Figure 11: Mean number of DTMs θ-similar to each other with an increasing angle of θ

Effect of flow slack on DTM selection Figure 9c quantifies the relationship between the number of DTMs and the flow slack factor ϵ as of the DTM selection process in Section 4.3. According to Definition 4.2, a sample TM can be a candidate DTM if its traffic across a network cut is at least $1 - \epsilon$ of the maximum traffic across the cut. So, a bigger ϵ will cause more TMs to be qualified as DTMs, among which a smaller subset can represent all the network cuts. Figure 9c is consistent with this expectation: the minimum DTM count to cover all network cuts reduces with the increase of ϵ, sharply in the beginning and slowing down as ϵ grows. A smaller number of DTMs means less computation for planning optimization, yet the Hose coverage may be compromised. We discuss the details in Figure 10.

It also shows the effect of edge threshold α on the number of DTMs. Interestingly, comparing to Figure 9b, the effective α value can be further reduced with DTM selection in place. Specifically, the top curves where α is 8%, 9%, and 10% show little difference in terms of the number of DTMs, although $\alpha = 8\%$ finds 25% fewer network cuts than $\alpha = 10\%$ in Figure 9b. This result proves the robustness of our DTM selection process: with a reasonable α, even



(a)

(b)

Figure 12: Traffic drop on Hose and Pipe network plans: (a) CDF of daily drop, (b) drop per day.

if some network cuts are not explicitly considered, the resulted difference in the number of DTMs is small.

Hose coverage of DTMs Figure 10 combines the above factors and shows their joint effect on DTM selection. The curves have similar trends as those in Figure 9c. However, for α values 8%, 9%, and 10%, their Hose coverage almost overlap completely. Thus, we claim the edge threshold $\alpha = 8\%$ is sufficient for our network, as the slightly lower number of DTMs can cover the Hose space equally well. Compared to Figure 9c, Hose coverage shows a more smooth, near-linear reduction with the increasing flow slack ϵ, which confirms the design purpose of our DTM selection process: a small set of well-chosen DTMs can reach high Hose coverage. We set $\alpha = 8\%$ and $\epsilon = 0.1\%$ in production and reach a relatively high Hose coverage of 83%.

DTM Similarity From another angle to examine the coverage, we also analyze the similarity of DTMs. A diverse set of DTMs implies tolerance to traffic uncertainty. We define similarity between two DTMs M_1 and M_2 as follows:

$$\text{Similarity}(M_1, M_2) = \frac{< M_1 \cdot M_2 >}{\|M_1\|_2 \, \|M_2\|_2} \tag{12}$$

where $\|.\|_2$ denotes the L2-norm of a matrix and $< \cdot >$ denotes the dot product of the vectors obtained from unrolling the matrices. The similarity can be expressed as the cosine of the angle of alignment between the two matrices w.r.t. the origin. For example, $\text{Similarity}(M_1, M_2) = 1$ if M_2 is a multiple of M_1 by a strictly positive scalar. We then define the two matrices M_1 and M_2 to be θ-similar iff $\text{Similarity}(M_1, M_2) \geq \cos\theta$.

We evaluate the similarity of the DTMs used in production, where $\alpha = 8\%$ and $\epsilon = 0.1\%$. For each DTM, we compute the number of DTMs (including itself) that are θ-similar to it. We then average the numbers across all DTMs to get the mean DTM θ-similarity. Figure 11 shows this metric with the increase of θ. When DTMs are all isolated, the mean number of DTMs similar to each other should be 1, i.e., a DTM is only similar to itself. As θ increases, DTMs further away are θ-similar, and the mean DTM similarity would increase. We see here that the mean DTM similarity remains close to 1, even for values of θ in excess of 20°, indicating that the DTMs are each well-isolated in the Hose space of TMs, and that applying additional clustering would not yield many benefits.

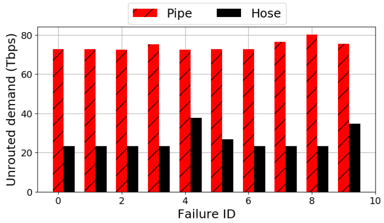


Figure 13: Traffic drop under random fiber failures.

6.2 Performance Comparison with Pipe

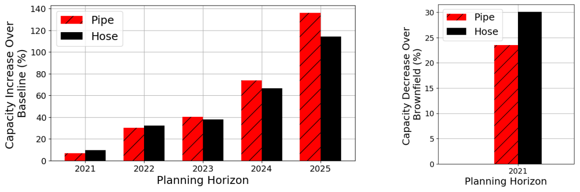
Planning result vs. actual traffic We evaluate the planning accuracy by seeing whether the planned capacity can satisfy the actual traffic. To do so, we take the June 2020 network as our baseline topology and perform demand forecast of the next 6 months with both Hose and Pipe models to generate the capacity plans. Note that these plans are not the production topology in December 2020, but rather what the network hypothetically asked for 6 months ago in history. We evaluate how good these plans are by replaying 28 days of actual traffic in December 2020 on them. The difference between the actual traffic and the forecast traffic is the main reason for either under-provisioning causing dropped demand or over-provisioning causing wasted capacity.

Traffic drop is especially harmful to service performance. Figure 12 compares the dropped traffic volume on the Hose and Pipe plans under the steady state, i.e., no failures in the network. In subfigure (a), we observe from the CDF distribution that the daily dropped demand in the Hose model is much lower than Pipe, and for 80% of the days, the difference in dropped demand is almost 50%. In the day-to-day view of subfigure (b), for almost all days, the dropped demand for Pipe is higher than Hose, and the difference can be as high as several Tbps on some days like 12/08 and 12/13. Both results confirm our initial hypothesis in Section 2 that Hose-based planning is more resilient to traffic dynamics and can provide better overall performance.

Resilience to unplanned failures We further compare the traffic drop with Hose and Pipe plans under unplanned failures in Figure 13. It uses the same setting as Figure 12 with 10 randomly selected fiber cuts. We observe that Hose consistently drops less traffic than Pipe in all scenarios by 50%-75%. Compared to steady state in Figure 13, the benefit of Hose dropping less traffic is even more profound.

Yearly capacity growth Figure 14a shows Hose and Pipe's yearly capacity growth as a percentage of the baseline capacity in the next 5 years. The projected traffic demand from our production traffic forecaster roughly doubles every two years. Hose-based capacity planning is more capacity-efficient in the long run. First, the relative capacity gain of Hose is greater year by year. By 2025, it can save 17.4% capacity compared to Pipe. Second, while both Pipe and Hose capacity scale faster than traffic growth (more capacity is needed to account for failure scenarios), the Hose capacity increases at a lower rate. The capacity saving of the Hose model comes from the multiplexing gain of traffic aggregation, as discussed in Section 2.

The advantage is not obvious in the near future because the Hose model has been in use for only a few years. Our current



(a)

(b)

Figure 14: (a) Yearly capacity growth of Hose and Pipe, (b) 2021 capacity decrease with clean-slate planning.

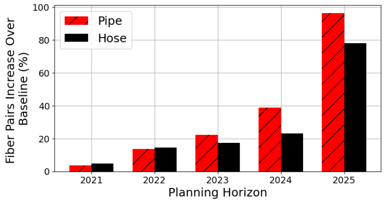


Figure 15: Cost benefit of Hose measured by fiber consumption.

topology is mostly built with Pipe-based planning, and it takes time to become Hose-compliant. In Figure 14b, we remove this factor by planning the network from scratch, and we show the capacity decrease against the 2021 Pipe result in Figure 14a. In this case, Hose can save almost 7% more capacity than Pipe. These observations suggest evolving a network with the Hose model can reach a more optimized network topology than evolving with the Pipe model.

Cost saving While we cannot share the proprietary cost values, we approximate the cost benefit of Hose using the fiber pair consumption. Figure 15 shows the additional percentage of fiber usage normalized by the baseline. We observe a similar trend as the capacity growth. The cost advantage manifests as the years of deployment increase, with as high as 20% saving in four to five years.

Optimization time vs. accuracy Table 2 further investigates the Figure 14b results with varying Hose coverage. We see even a relatively low coverage of 40% achieves a large capacity saving of 8.62%. At a high coverage of 83%, the overall computation time is an affordable 1063 minutes, or 17.7 hours. Because the DTMs are consumed by the optimization procedure iteratively in batches, the DTMs in later batches may already be satisfied by earlier batches. Thus, the computation time per DTM is only a few minutes, and further reduces given more DTMs, thanks to the batching effect. This result highlights that our solution is scalable and insensitive to the DTM selection when the coverage is sufficient. Figure 16 compares the planned capacity per IP link for different Hose coverage values against the 83% coverage baseline. The planning difference is remarkable, though shrinking, as the Hose coverage improves. Considering the good time scalability of our system, we suggest choosing a high Hose coverage in practice, so as to avoid under-provisioning from overly high capacity reduction such as achieved by coverage values of 58% and 67%. The capacity saving has little change when the coverage value is above 83%.

Hose coverage	# DTMs	Reduced capacity %	Time in mins	Time per DTM
40%	21	8.62	48	2.28
52%	64	8.28	312	4.87
58%	89	10.52	342	3.84
67%	154	9.31	412	2.67
83%	628	8.45	1063	1.69

Table 2: Capacity saving with different Hose coverage

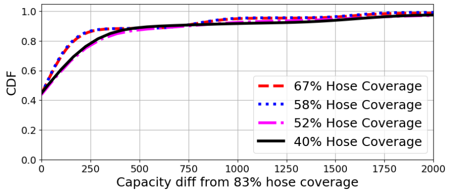


Figure 16: Capacity saving of Hose over Pipe: per-link capacity difference relative to the 83% coverage plan.

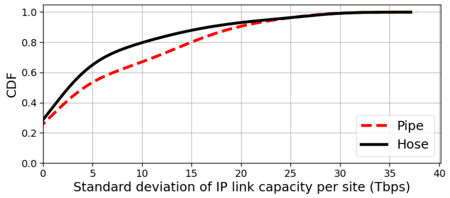


Figure 17: CDF of the capacity variance of IP links per site.

Capacity Distribution Figure 17 shows the standard deviation of capacity across all the IP links at each site for Year 1 planning (2021 result in Figure 14). Capacity is distributed more uniformly in Hose. For the Hose model, almost 70% of sites have capacity variance less than 5Tbps, while the number is only 50% for Pipe. At 80%, Pipe has a variance 1.5× larger than Hose. The tail of variance for Pipe is also larger than Hose. More uniform capacity distribution is desirable for resilience against unplanned failures and future scaling, because more TMs can fit into even link capacities at a site. Hose-based planning adds capacity more uniformly across links thanks to the variant TMs it has considered.

7 OPERATIONAL EXPERIENCE

We have learned important operational lessons throughout years of running Hose-based network planning in production. This section reveals unexpected use cases, system adjustments, and directions for future improvements.

7.1 Disaster Recovery Buffer

The concept of "disaster readiness" has been built into every aspect of Facebook's infrastructure [22]. Disaster refers to any catastrophic failure that takes a long time to recover, such as hurricane, major fire event in a DC, etc. Facebook conducts disaster recovery (DR) exercises to test its capability under actual disasters. These DR tests migrate requests originally sent to potentially failing DCs to other healthier DCs. This process explores the inter-service dependencies and dynamic resource constraints (such as compute and storage resources) to identify the mitigation plan for each service in real time. Each candidate mitigation plan will create a drastic shift to the original TM. For a network planned with Pipe, it requires careful evaluation of every TM (one for every candidate migration plan) to certify if the current production backbone can accommodate this changed TM. By moving to Hose-based network planning, a planner is able to provide an upper bound on the total ingress and egress traffic supported per DC. By looking at the current traffic utilization, one could quantify additional traffic that can be added to the DC without overloading the region, i.e., a planner is able to provide deterministic DR buffers that can be used by operational teams performing the DR exercise.

7.2 Partial Hose

Our Hose model is based on the general assumption that a service would send traffic to any destination region. However, we find a service may only need to communicate with a small subset of the regions, as the service placement is limited to these regions. For example, we have a data warehouse service that utilizes a specialized server type, which is only available in 4 regions. The data warehouse traffic accounts for 75% of the total inter-region traffic between these 4 regions. Taking service placement into consideration can help us estimate DTMs more accurately. Thus, in this case, we can create a smaller Hose, consisting of only these 4 regions, and a larger Hose consisting of the remaining traffic to all destinations. This *partial Hose* model gives us additional information of the application communication patterns. However, considering the large number of services at Facebook, we only use partial Hose under two conditions: (1) if the traffic volume of the service is significantly large; (2) if the service placement is inherently limited by the hardware resource such that it is impossible to move the service to other regions easily.

7.3 A/B Testing

Testing network plans using demand forecast and modeling assumptions for production network is non-trivial. The actual performance only becomes clear several months or even years after the plan is deployed. In practice, we rely on extensive A/B testing and manual verification by experts across teams, typically from network planning, sourcing, and deployment teams, to verify our designs. We set up A/B testing for different network build plans. For example, given two sets of input demands, or two different policies, two versions of PORs will be generated. We compare key metrics quantitatively, such as IP topology, optical fiber count, cost, flow availability, latency, failures unsatisfied, etc. The experts then check these multiple designs for any anomalies. Right now, our testing strategy is largely based on engineering tribe knowledge. We encourage more research in this area to enable scientific A/B testing for network build plans.

7.4 Stability of Parameter Setting

In production deployment, we find the choice of parameters, e.g., Hose coverage, to be stable over time. The fundamental reason is the relative stability of traffic demand variations. The backbone traffic is dominated by machine-to-machine traffic between DCs, which fundamentally reflects the service placement. In production, the service placement is relatively stable to accommodate various infrastructure constraints pertaining to server availability, fault tolerance requirements, and disaster recovery planning. Thus, we observed that *complete* demand shifts are rare but moderate shifts of 30-50% traffic between different regions are still common under different failures. This leads to our engineering choice of 83% Hose coverage, as demonstrated earlier in Section 6.

8 RELATED WORK

Hose model in Virtual Private Network (VPN) The seminal work by Duffield *et al.* [9] proposes the Hose model for resource management in VPNs. It allocates bandwidth to satisfy Hose-conformant worst-case traffic distribution. Several follow-up work have been developed to improve the dynamic bandwidth resizing [7, 10, 20]. Their problem formulation is fundamentally different from ours as they allocate *existing* bandwidth resources to best guarantee the Hose requirement, whereas our work designs the underlying network to satisfy all possible traffic splits under a Hose. Our work is more closely related to [15] which designs a tree topology to satisfy Hose, while our solution works with general graphs.

Hose model in cloud resource sharing Hose is also used to model demands in DCs [4] and the cloud environment [4, 8, 14, 17–19]. These work use the Hose model for per-VM traffic demand and use a big virtual switch to abstract the network fabric. For instance, Oktopus [4] proposes a VM placement algorithm based on the Hose constraints of any two sets of VMs. The demand between the two VM sets is determined by the sum of all VMs' Hose demands in each set. This model essentially adds up all the worst-case TMs and results in significant over-provisioning. Our approach is more efficient because we use an operationally effective slack factor (Section 4.3) to choose hard-enough TMs, but not the worst-case TMs, and the resulting multiplexing gain has been demonstrated in production (Section 6.2).

Network planning Scenario-based planning copes with traffic uncertainty by using forecast results for a few given network scenarios, and each scenario emphasizes on a set of TMs [23]. Our selection of TMs is more general, not limited by any pre-defined scenarios. Zhang *et al.* proposes to find critical TMs by clustering for general network analysis applications [24]. However, their work is not tailored for network planning. We are interested in applying their algorithm to network planning and comparing the efficacy against our DTM selection algorithm. Little has been revealed about production network planning except for a brief introduction in [5]. To the best of our knowledge, we are the first to describe real-world network planning in detail.

9 CONCLUSION

Network planning plays an important role in long-term network evolvement and service growth. In this paper, we demonstrate the effectiveness of using the Hose model for network planning by leveraging its multiplexing gain to simultaneously save capacity and absorb traffic uncertainty. We share the experience of planning a production backbone over several years. Our work sheds light on the potential of Hose in a new problem domain, network planning, in the hope of stimulating more research in this area.

REFERENCES

[1] [n. d.]. FICO Xpress Optimization. ([n. d.]). https://www.fico.com/en/products/fico-xpress-optimization

[2] [n. d.]. TAO: The power of the graph. ([n. d.]). https://engineering.fb.com/2013/06/25/core-data/tao-the-power-of-the-graph/

[3] R. Andersen, F. Chung, A. Sen, and G. Xue. 2004. On Disjoint Path Pairs with Wavelength Continuity Constraint in WDM Networks. *IEEE INFOCOM* (2004).

[4] Hitesh Ballani, Paolo Costa, Thomas Karagiannis, and Ant Rowstron. 2011. Towards Predictable Datacenter Networks. In *Proceedings of the ACM SIGCOMM 2011 Conference*. Association for Computing Machinery, New York, NY, USA, 12.

[5] Ajay Kumar Bangla, Alireza Ghaffarkhah, Ben Preskill, Bikash Koley, Christoph Albrecht, Emilie Danna, Joe Jiang, and Xiaoxue Zhao. 2015. Capacity Planning for the Google Backbone Network. In *ISMP 2015 (International Symposium on Mathematical Programming)*.

[6] C Bradford Barber, David P Dobkin, and Hannu Huhdanpaa. 1996. The Quickhull Algorithm for Convex Hulls. *ACM Transactions on Mathematical Software (TOMS)* 22, 4 (1996), 469–483.

[7] Haesun Byun and Meejeong Lee. 2007. Extensions to P2MP RSVP-TE for VPN-specific State Provisioning with Fair Resource Sharing. *Comput. Commun.* 30, 18 (Dec. 2007), 3736–3745.

[8] Mosharaf Chowdhury, Yuan Zhong, and Ion Stoica. 2014. Efficient Coflow Scheduling with Varys. 44, 4 (2014).

[9] N. G. Duffield, P. Goyal, A. Greenberg, P. Mishra, K. K. Ramakrishnan, and J. E. V. der Merwe. 1999. A Flexible Model for Resource Management in Virtual Private Networks. *ACM Sigcomm, San Diego, California, USA* (1999).

[10] Friedrich Eisenbrand and Edda Happ. 2006. Provisioning a Virtual Private Network Under the Presence of Non-communicating Groups. In *Proceedings of the 6th Italian Conference on Algorithms and Complexity (CIAC'06)*. 105–114.

[11] Shimon Even, Alon Itai, and Adi Shamir. 1975. On The Complexity of Time Table and Multi-commodity Flow Problems. In *16th Annual Symposium on Foundations of Computer Science (sfcs 1975)*. IEEE, 184–193.

[12] Uriel Feige. 1998. A Threshold of ln n for Approximating Set Cover. *Journal of the ACM (JACM)* 45, 4 (1998), 634–652.

[13] J. A. Fingerhut, S. Suri, and J. S. Turner. 1997. Designing Least-cost Nonblocking Broadband Networks. *Journal of Algorithms* 24, 2 (Aug. 1997), 287–309.

[14] Albert Greenberg, James R. Hamilton, Navendu Jain, Srikanth Kandula, Changhoon Kim, Parantap Lahiri, David A. Maltz, Parveen Patel, and Sudipta Sengupta. 2009. VL2: A Scalable and Flexible Data Center Network. In *Proceedings of the ACM SIGCOMM 2009 Conference on Data Communication*. Association for Computing Machinery, New York, NY, USA, 12.

[15] Anupam Gupta, Jon Kleinberg, Amit Kumar, Rajeev Rastogi, and Bulent Yener. 2001. Provisioning a Virtual Private Network: A Network Design Problem for Multicommodity Flow. In *Proceedings of the Thirty-third Annual ACM Symposium on Theory of Computing (STOC '01)*. 389–398.

[16] M Rashidul Islam and M Hanif Chaudhry. 1998. Modeling of Constituent Transport in Unsteady Flows in Pipe Networks. *Journal of Hydraulic Engineering* 124, 11 (1998), 1115–1124.

[17] Simon Kassing, Asaf Valadarsky, Gal Shahaf, Michael Schapira, and Ankit Singla. 2017. Beyond Fat-Trees without Antennae, Mirrors, and Disco-Balls. In *Proceedings of the Conference of the ACM Special Interest Group on Data Communication*. Association for Computing Machinery, New York, NY, USA.

[18] Lucian Popa, Gautam Kumar, Mosharaf Chowdhury, Arvind Krishnamurthy, Sylvia Ratnasamy, and Ion Stoica. 2012. FairCloud: Sharing the Network in Cloud Computing. *SIGCOMM Comput. Commun. Rev.* 42, 4 (Aug. 2012).

[19] Henrique Rodrigues, Jose Renato Santos, Yoshio Turner, Paolo Soares, and Dorgival Guedes. 2011. Gatekeeper: Supporting Bandwidth Guarantees for Multi-Tenant Datacenter Networks. In *Proceedings of the 3rd Conference on I/O Virtualization*. USENIX Association, USA.

[20] Thomas Rothvoß and Laura Sanità. 2009. On the Complexity of the Asymmetric VPN Problem. In *Proceedings of the 12th International Workshop and 13th International Workshop on Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques (APPROX '09 / RANDOM '09)*. Springer-Verlag, Berlin, Heidelberg, 326–338. https://doi.org/10.1007/978-3-642-03685-9_25

[21] Daniel Semrau and Polina Bayvel. 2018. The Gaussian Noise Model in the Presence of Inter-channel Stimulated Raman Scattering. *Journal of Lightwave Technology* 36, 14 (2018), 3046–3055.

[22] Kaushik Veeraraghavan, Justin Meza, Scott Michelson, Sankaralingam Panneerselvam, Alex Gyori, David Chou, Sonia Margulis, Daniel Obenshain, Shruti Padmanabha, Ashish Shah, Yee Jiun Song, and Tianyin Xu. 2018. Maelstrom: Mitigating Datacenter-level Disasters by Draining Interdependent Traffic Safely and Efficiently. In *13th USENIX Symposium on Operating Systems Design and Implementation (OSDI 18)*. USENIX Association, Carlsbad, CA, 373–389.

[23] Shu Zhang and Ulrich Killat. 2011. Multiple-layer Network Planning with Scenario-based Traffic Forecast. In *Proceedings of the 17th International Conference on Energy-aware Communications*. 77–88.

[24] Y. Zhang and Z. Ge. 2005. Finding Critical Traffic Matrices. In *2005 International Conference on Dependable Systems and Networks (DSN'05)*. 188–197.