Google

Capacity planning for the Google backbone network

Christoph Albrecht, Ajay Bangla, Emilie Danna, Alireza Ghaffarkhah, Joe Jiang, Bikash Koley, Ben Preskill, Xiaoxue Zhao

July 13, 2015

ISMP

Multiple large backbone networks

B2: Internet facing backbone
70+ locations in 33 countries

B4: Global software-defined
inter-datacenter backbone

20,000+ circuits in operation
40,000+ submarine fiber-pair miles

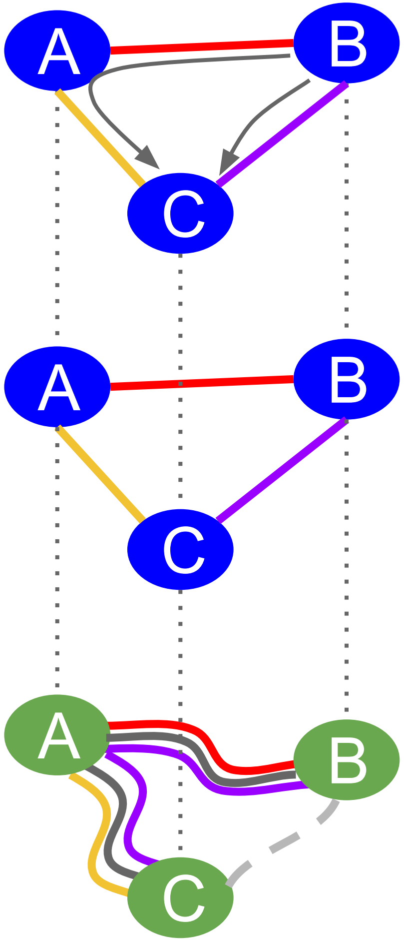
Google

Multiple layers

Flows routed on logical links

Logical Topology (L3): router adjacencies

Physical Topology (L1): fiber paths

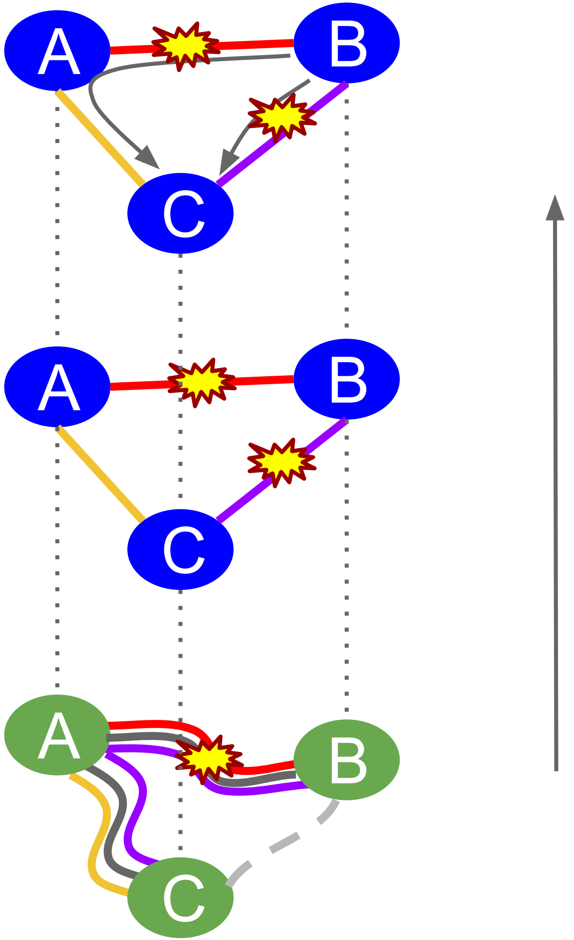


Multiple layers

Flows routed on logical links

Logical Topology (L3): router adjacencies

Physical Topology (L1): fiber paths



Failures propagate from layer to layer

Multiple time horizons

O(seconds)	O(months)	O(years)
• Failure events • Fast protection/restoration • Routing changes • Definitive failure repair: O (hours or days)	• Demand variation • Capacity changes • Risk assessment	• Long-term demand forecast • Topology optimization and simulation • What-if business case analysis

Multiple objectives

Strategic objectives: **minimize cost, ensure scalability**

Service level objectives (SLO): **latency, availability**

- ➔ Failures are modeled probabilistically
- ➔ The objective is defined as points on the probability distribution
- ➔ Latency example: the 95th percentile of latency from A to B is at most 17 ms
- ➔ Availability example: 10 Gbps of bandwidth is available from A to B at least 99.9% of the time

Multiple practical constraints

Example: Routing of flows on the logical graph

➔ A flow can take a limited number of paths

➔ Routing is sometimes not deterministic

➔ There is a time delay to modify routing after a failure happens

Deterministic optimization

Problem

What is the cheapest network that can route flows during a given set of failure scenarios?

➔ **L3-only version:** physical topology and logical/physical mapping are fixed, decide logical capacity

➔ **Cross-layer version:** decide physical and logical topology, mapping between the two, and logical capacity

Mixed integer program: building block

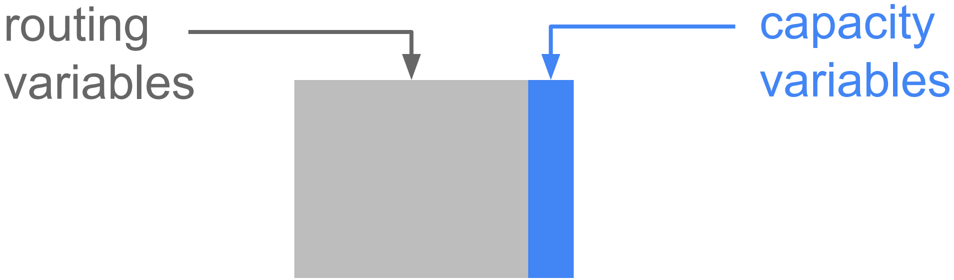
Each flow is satisfied during each failure scenario in the given set.

For each failure scenario:

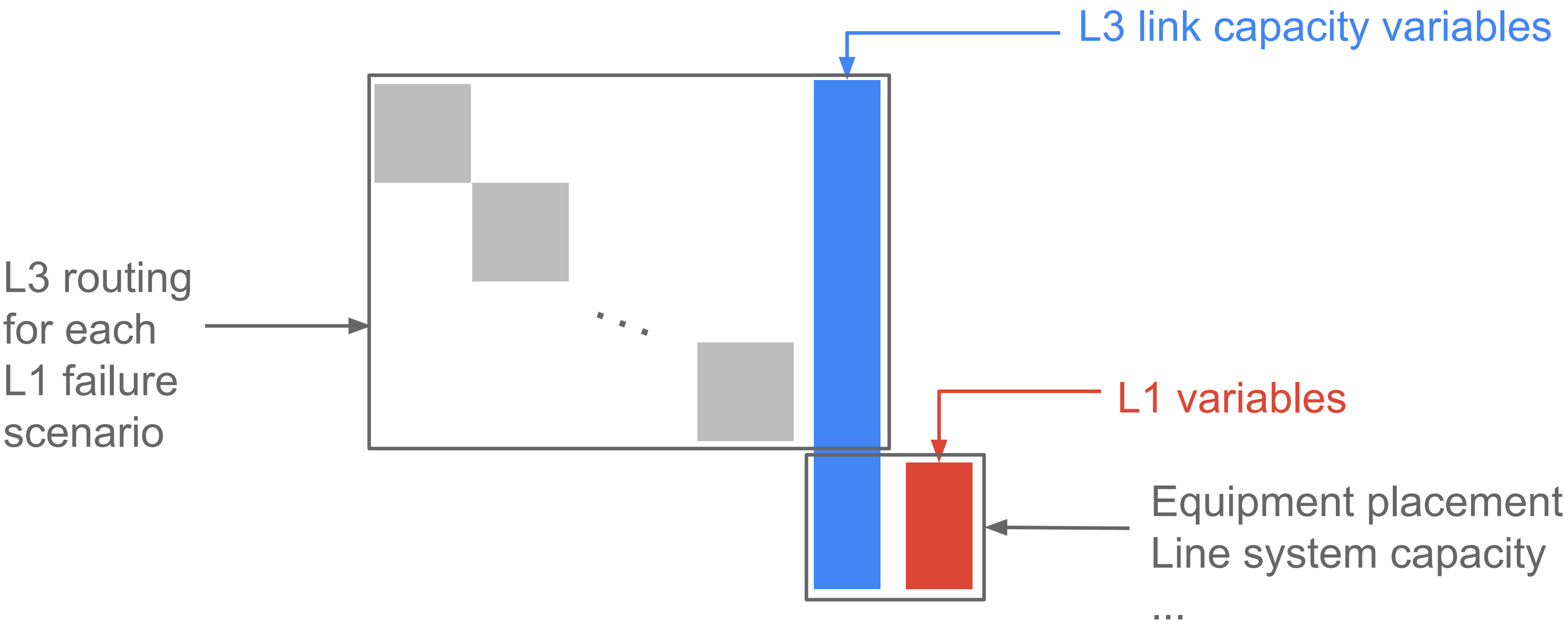
Variables: how the flow is routed

Constraints: for each link, utilization is less than capacity under failure

Matrix represented as:

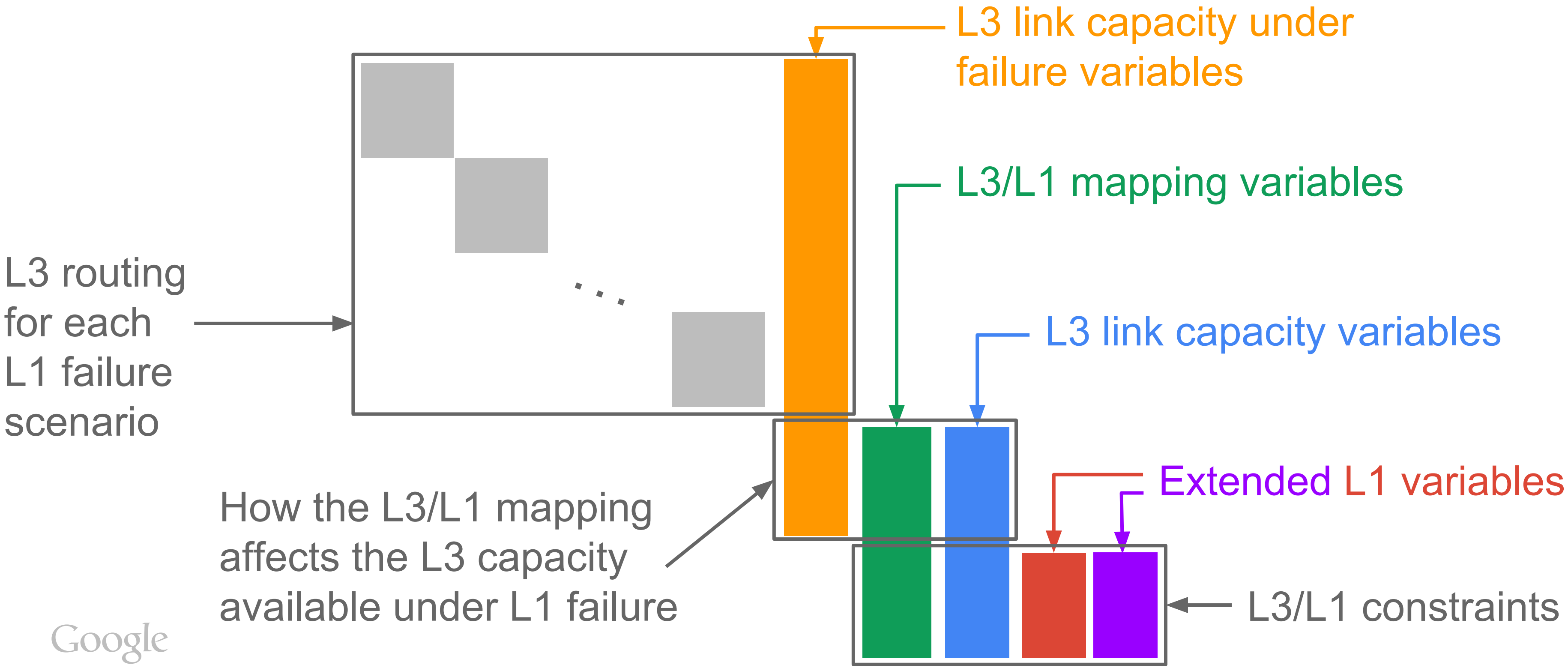


Mixed integer program: L3-only



Google

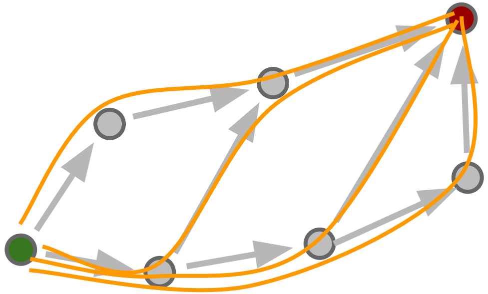
Mixed integer program: cross-layer



Google

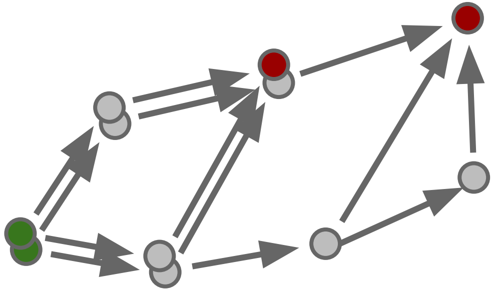
Routing for each failure scenario

Path formulation



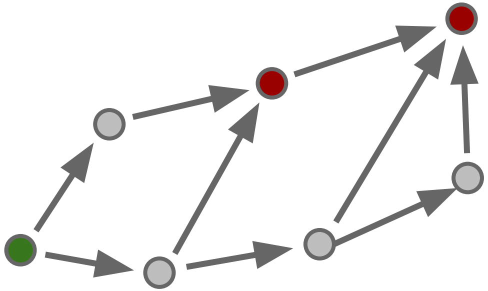
For each src-dst flow:
. Multiple paths are generated from src to dst
. One variable per path for the amount of traffic along the path.

Edge formulation



For each src-dst flow:
. One variable for the amount of traffic for each link and each direction
. At each node: flow conservation constraint

Edge formulation for single source to multiple destinations flow

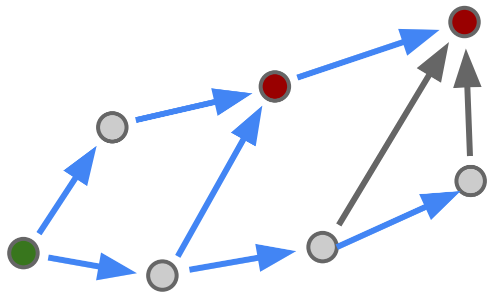


. All flows of the same source are combined into one flow with multiple destinations.
. For each src-multiple dst flow: (link, direction) variables and flow conservation constraints

Google

Latency constraints

Strict version:
Edge formulation
for single source to multiple destinations
on the shortest path tree



Challenges:

. How to make the constraint **less strict**?
. How to make it **probabilistic**?

Results

Potential cost reduction:

Cross-layer optimization can reduce cost **2x** more than L3-only optimization

Stochastic simulation

Google

Problem

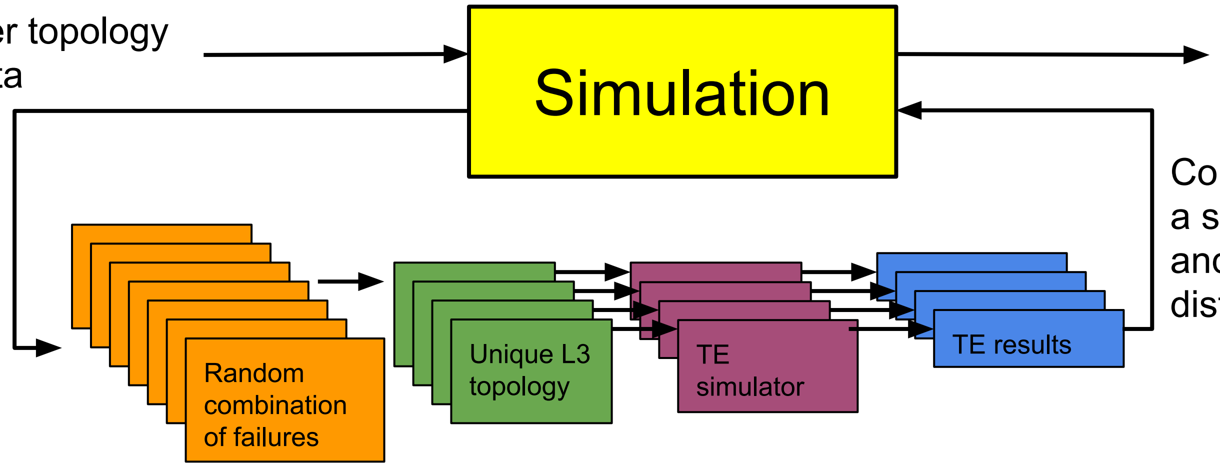
Does a given network meet availability and latency SLOs?

- ➔ Current network: risk assessment
- ➔ Hypothetical future networks: what-if analysis

Monte Carlo simulation

Input:
. Demand
. Cross-layer topology
. Failure data

Output:
Whether each flow meets SLO



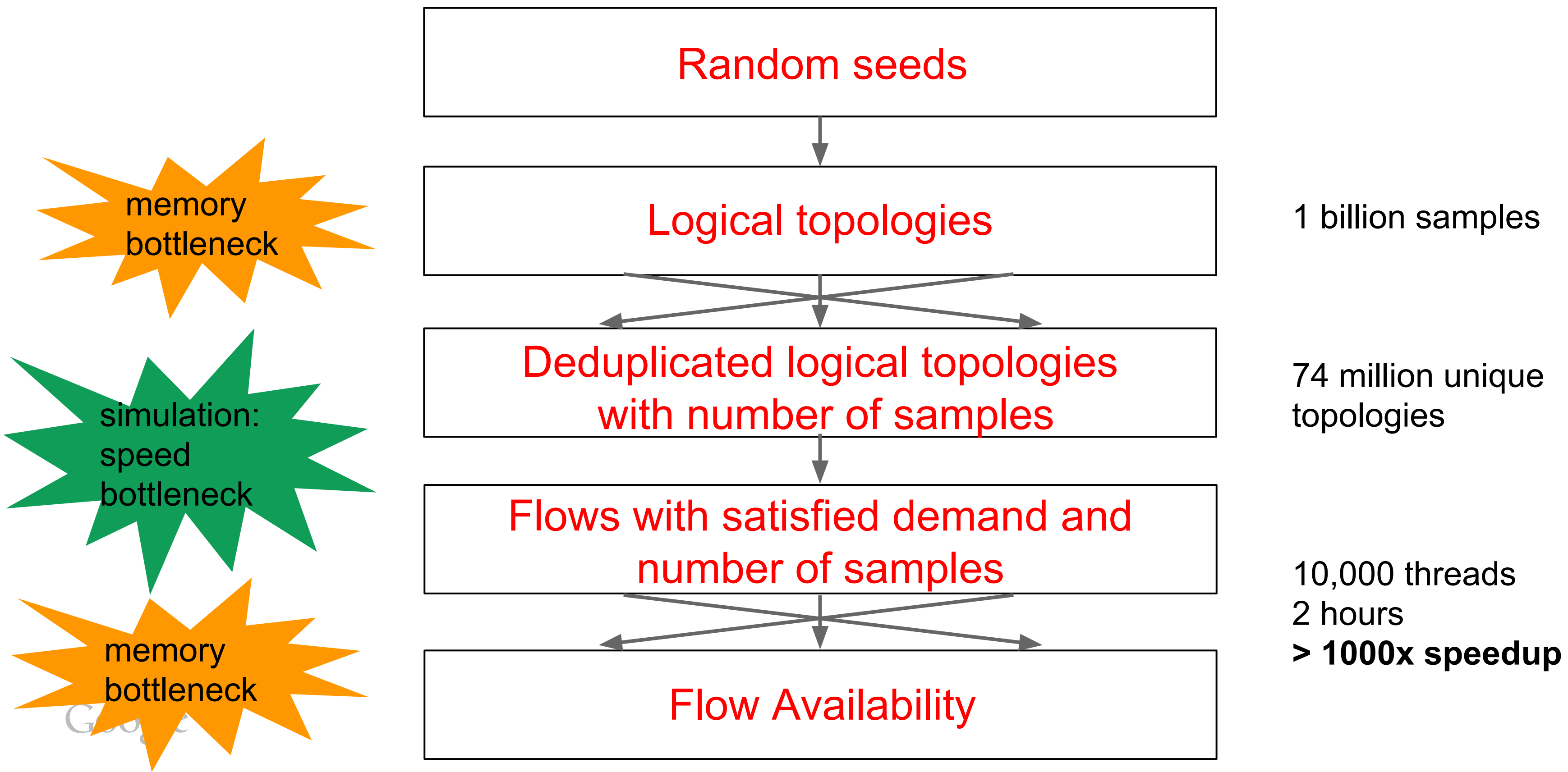
Combine results into a satisfied demand and latency distribution

Draw many samples from the failure distribution

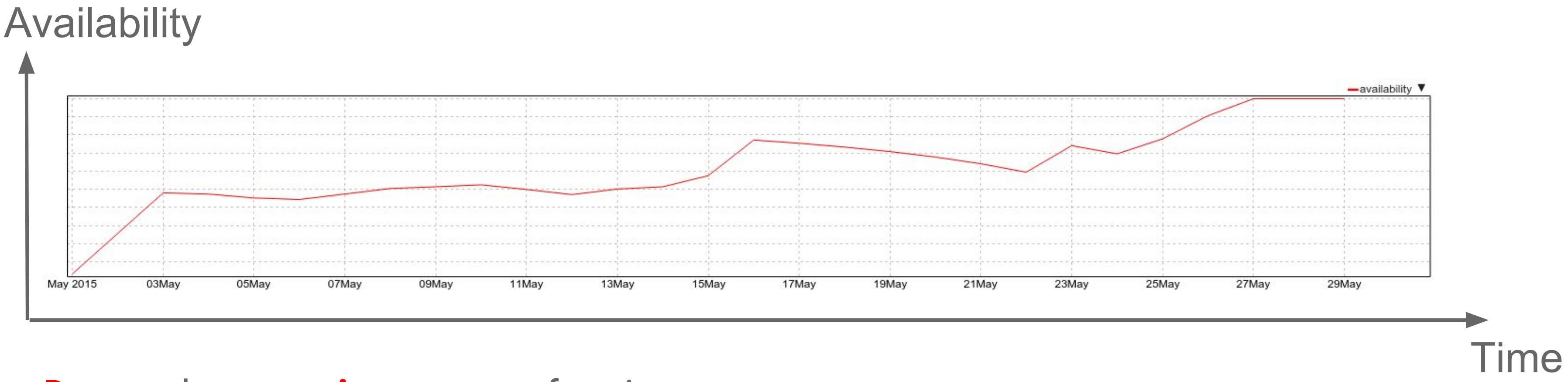
Derive corresponding L3 topologies with link capacity and deduplicate

For each L3 topology:
Evaluate the satisfied demand and latency with the traffic engineering (TE) simulator

Parallel implementation



Results



Data and **automation** are transforming

➔ our decision making

➔ the definition of our business: measurable service quality and guarantees

Stochastic optimization

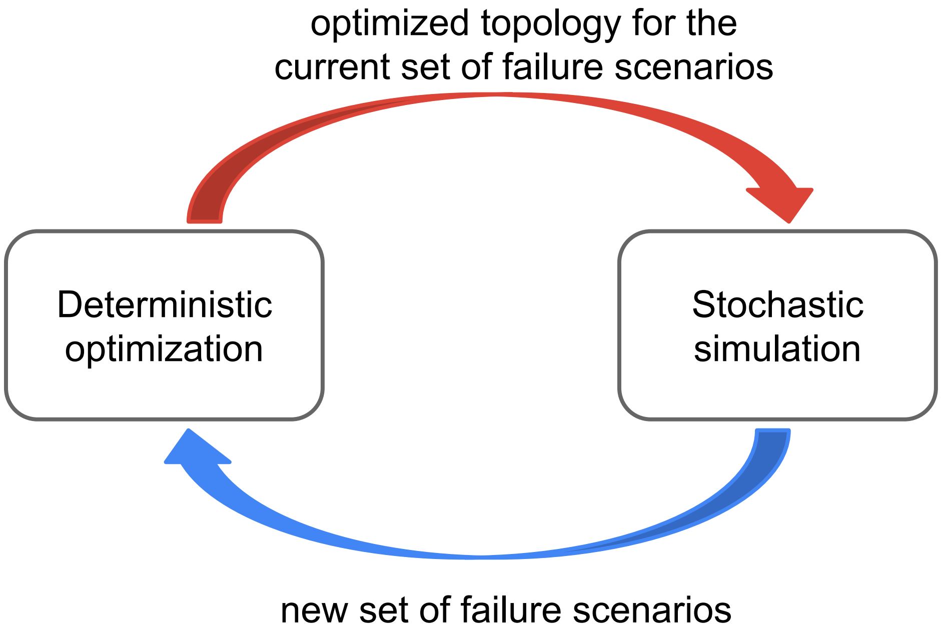
Google

Problem

What is the cheapest network that can meet SLO?

- ➔ Probabilistic modeling of failures
- ➔ SLO = chance constraints
  - ◆ Probability (latency from A to B <= 17 ms) >= 0.95
  - ◆ Probability (satisfied demand from A to B >= 10 Gbps) >= 0.999

Simulation / Optimization loop with scenario-based approach



Greedy approach to meet SLO by optimizing with the smallest number of failure scenarios

Add failure scenarios with
. highest probability
. highest number or volume of flows that miss SLO and are not satisfied during that failure scenario

Challenges

Tradeoff between accuracy, optimality, scalability, complexity and speed

Examples

→ **Accuracy:** routing convergence
→ **Optimality:** better stochastic optimization
→ **Scalability:** more failure scenarios
→ **Complexity:** explanation of solutions to our users
→ **Speed:** repair the topology on the fly (transport SDN)

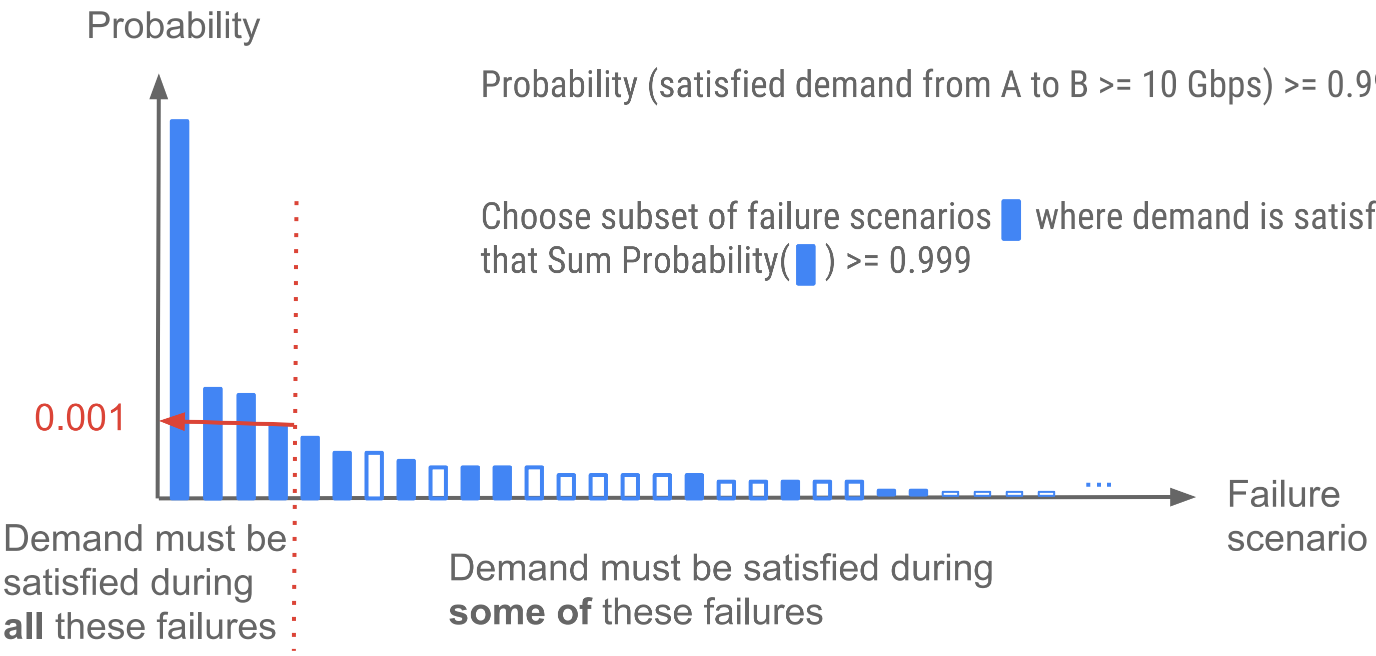
Thank you

Google

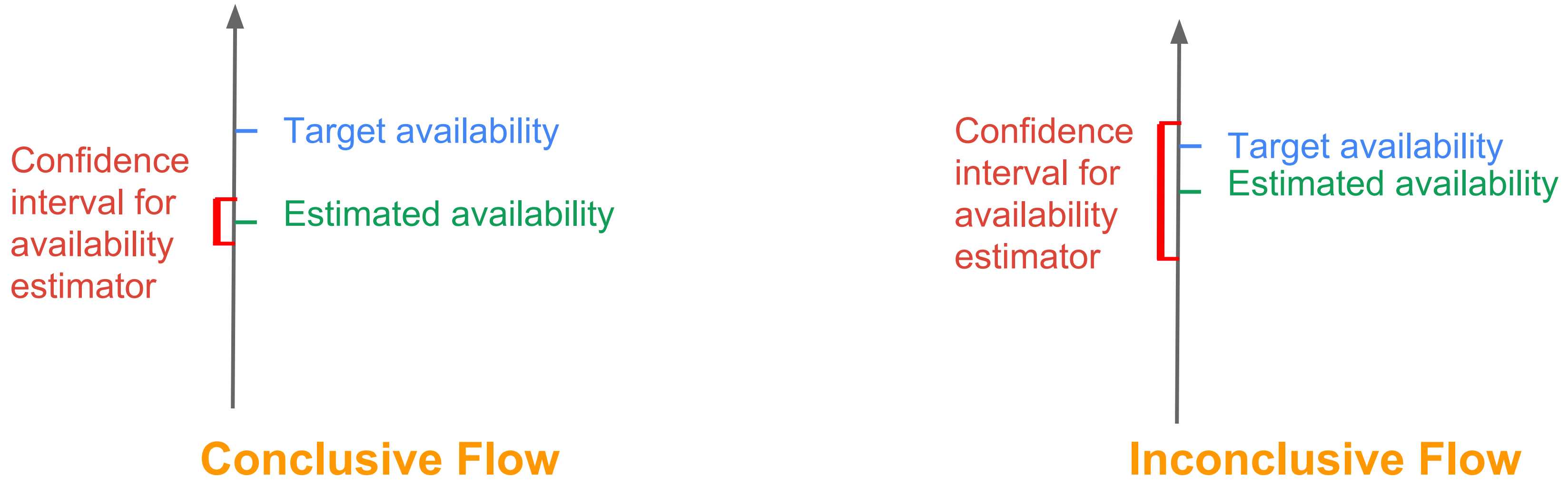
Scenario-based approach

Probability (satisfied demand from A to B >= 10 Gbps) >= 0.999

Choose subset of failure scenarios where demand is satisfied such that Sum Probability( ) >= 0.999



Accuracy



The availability calculation is a statistical estimation

Flow is conclusive if confidence interval lies entirely on one side of its target availability

This is used to determine the necessary number of samples