

TOPOS INSTITUTE COLLOQUIUM · 15 OCTOBER 2026

A Fast *Quantitative* Analyzer for NetKAT

Alexandra Silva · Cornell University

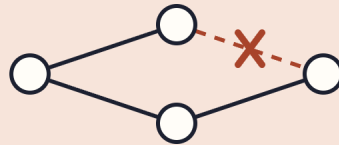
Joint work with Emmanuel Suárez Acevedo, Tiago Ferreira, Kevin Batz, Oliver Bøving, Thomas Lu, Qiancheng Fu, Mark Moeller and Nate Foster

Weighted NetKAT (arXiv 2604.13987) · A Fast Quantitative Analyzer for NetKAT (arXiv 2607.14420)

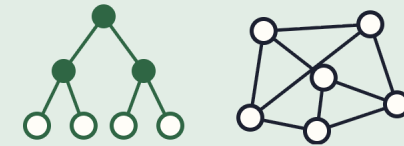
Quantitative questions about networks



Bandwidth vs latency



Resilience

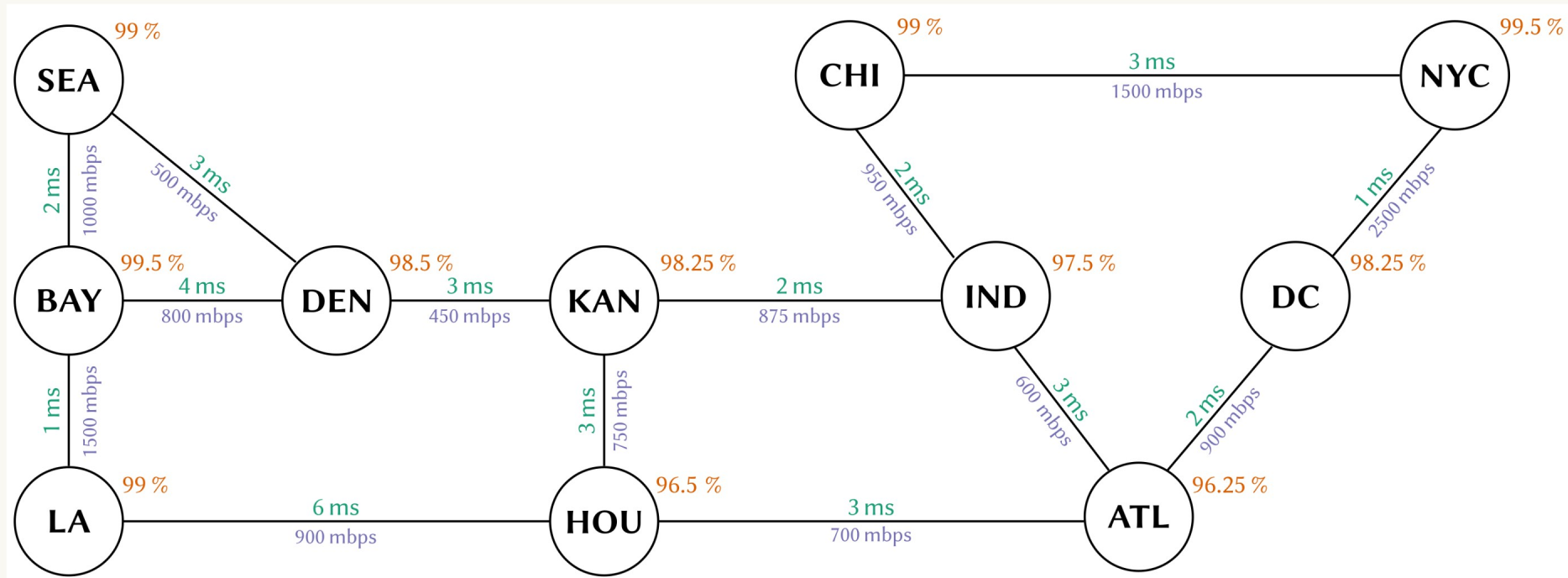


Topology design

Engineers often need to reason about *quantities*. Most verification tools answer *yes or no*.

In this talk: language design (syntax, semantics, algorithms) underlying a tool for *fast* quantitative verification.

The Abilene backbone



What is the max bandwidth between SEA and NYC?

Can LA deliver packets to CHI within 10 ms?

Is all traffic delivered with at least 90% reliability?

Goal

1

A language

to model networks together with their quantitative information: latency, bandwidth, reliability, cost.

2

Algorithms

to answer quantitative questions about these models automatically, with a witness path when one exists.

3

Scalability

make these algorithms scale to realistic networks, using clever symbolic data structures.

network + quantities



weighted NetKAT program



decision procedure



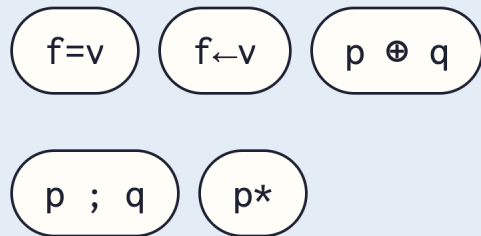
answer + witness

Background: NetKAT

sw	pt	src	dst	vlan	
1	1	10.0.0.1	10.0.0.7	42	...

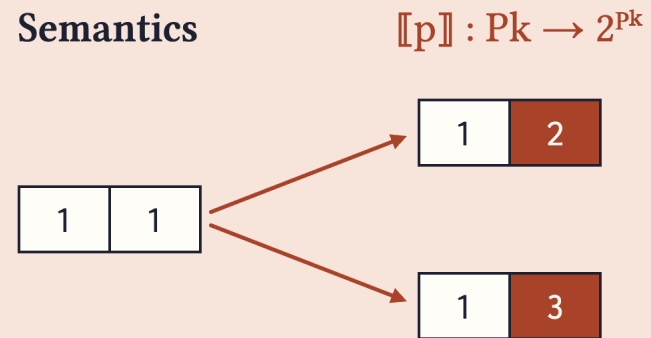
Packets are records of fields.

Policies



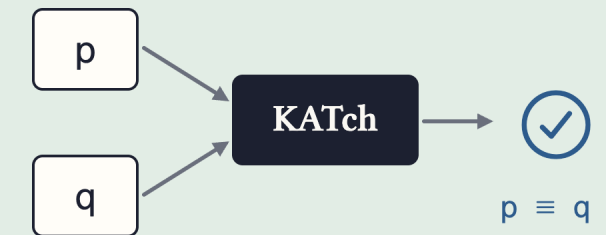
Tests, assignments, union, sequencing, star:
Kleene algebra with tests.

Semantics



$pt \leftarrow 2 \oplus pt \leftarrow 3$: one packet in, a *set* of packets out.

Payoff



Sound, complete axioms. KATch decides equivalence fast, or gives a counterexample.

From Boolean to weighted semantics

$$\llbracket p \rrbracket : P_k \rightarrow P_k \rightarrow \mathbb{B}$$

$$\llbracket p \rrbracket : P_k \rightarrow P_k \rightarrow \mathbf{S}$$

The semiring establishes *what* is measured.

Semirings and network properties

Semiring	$(S, +, \cdot, 0, 1)$	Network property
Boolean	$(\{0,1\}, \vee, \wedge, 0, 1)$	Reachability (NetKAT)
Probability	$([0,\infty], +, \times, 0, 1)$	Delivery probability
Tropical	$(\mathbb{N} \cup \{\infty\}, \min, +, \infty, 0)$	Best-case latency, cost, confidentiality
Arctic	$(\mathbb{N} \cup \{\pm\infty\}, \max, +, -\infty, 0)$	Worst-case latency, information leakage
Bottleneck	$(\mathbb{N} \cup \{\pm\infty\}, \max, \min, -\infty, \infty)$	Bandwidth of the widest path
Viterbi	$([0,1], \max, \times, 0, 1)$	Most reliable path
Probabilistic-union	$([0,1] \cup \{-\infty\}, \max, \wp, -\infty, 0)$	Worst-case failure rate

$r_1 \wp r_2 = r_1 + r_2 - r_1 \cdot r_2$: the probability that at least one of two independent failures happens.

Syntax of weighted NetKAT

$t ::=$ $\text{false} \mid \text{true} \mid t \vee t \mid t \wedge t \mid \neg t$ | $f = v$
Boolean algebra · tests **packet algebra**

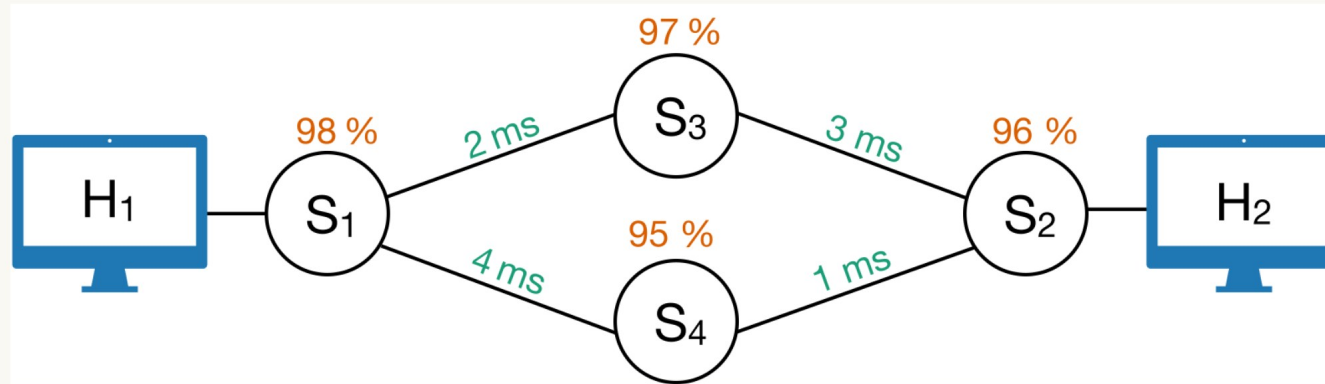
$p ::=$ t | $f \leftarrow v$ | $p ; q \mid p \oplus q \mid p^*$ | $w \odot p$ ← new: semiring weights
packet algebra **Kleene algebra**

$\oplus$ adds the weights of two branches.

$;$ multiplies weights along a path.

$w \odot p$ scales p by the weight w .

Weights on a network topology



LATENCY · TROPICAL

```
p1 := if dst = H2 then  
    (2ms ⊙ sw←S3) ⊕ (4ms ⊙ sw←S4)  
else ...
```

RELIABILITY · VITERBI

```
p1 := 98% ⊙ ( if dst = H2 then  
    sw←S3 ⊕ sw←S4  
else ... )
```

Denotational semantics

$\llbracket p \rrbracket : P_k \rightarrow P_k \rightarrow \mathbf{S}$ for a semiring $(S, +, \cdot, 0, 1)$: each policy denotes an S -weighted relation on packets.

$$\llbracket w \odot p \rrbracket(\alpha)(\beta) = w \cdot \llbracket p \rrbracket(\alpha)(\beta)$$

$$\llbracket p ; q \rrbracket(\alpha)(\beta) = \oplus_{\gamma} \llbracket p \rrbracket(\alpha)(\gamma) \cdot \llbracket q \rrbracket(\gamma)(\beta)$$

$$\llbracket p \oplus q \rrbracket(\alpha)(\beta) = \llbracket p \rrbracket(\alpha)(\beta) + \llbracket q \rrbracket(\alpha)(\beta)$$

$$\llbracket p^* \rrbracket = 1 + \llbracket p \rrbracket + \llbracket p;p \rrbracket + \llbracket p;p;p \rrbracket + \cdots$$

Sequencing is matrix multiplication over S . Star is an *infinite* sum, so S must be ω -continuous.

One program, two semirings

$$p = 2 \odot (f=1) \oplus (4 \odot (f \leftarrow 2) \oplus 10 \odot (f \leftarrow 2))$$

$f=1 \rightarrow f=1$

2

Only the test branch fires.

$f=1 \rightarrow f=2$, arctic

$\max(4, 10) = 10$

Worst case: choice maximises.

$f=1 \rightarrow f=2$, tropical

$\min(4, 10) = 4$

Best case: choice minimises.

Thompson construction for weighted NetKAT automata

A WNKA $\mathcal{A} = (Q, \iota, \delta, \lambda)$ reads packets in linked pairs $\alpha\beta$, where $\mathcal{W}_S(X) \triangleq X \rightarrow S$ are the S -weightings of X .

Policy	States Q	Initial $\iota \in \mathcal{W}_S(Q)$	Transitions $\delta_{\alpha\beta} : Q \rightarrow \mathcal{W}_S(Q)$	Outputs $\lambda_{\alpha\beta} \in \mathcal{W}_S(Q)$
$\pi?$	$\{\heartsuit\}$	$\eta(\heartsuit)$	0	$\lambda(\heartsuit) = [\alpha = \beta = \pi]$
$\pi!$	$\{\diamondsuit\}$	$\eta(\diamondsuit)$	0	$\lambda(\diamondsuit) = [\beta = \pi]$
dup	$\{\heartsuit, \clubsuit\}$	$\eta(\heartsuit)$	$\delta(\heartsuit) = \eta(\clubsuit)$ if $\alpha = \beta$	$\lambda(\clubsuit) = [\alpha = \beta]$
$r \odot p_1$	Q_1	$r \cdot \iota_1$	δ_1	λ_1
$p_1 \oplus p_2$	$Q_1 + Q_2$	$[\iota_1, \iota_2]$	$\text{diag}(\delta_1, \delta_2)$	$[\lambda_1, \lambda_2]$
$p_1 ; p_2$	$Q_1 + Q_2$	$[\iota_1, 0]$	$\delta_1, \Sigma \gamma \lambda_1 \iota_2 \delta_2, \delta_2$	$[\Sigma \gamma \lambda_1 \iota_2 \lambda_2, \lambda_2]$
p_1^*	$Q_1 + \{\heartsuit\}$	$[0, 1]$	$\delta_1 + \lambda_1 \lambda \heartsuit \iota_1 \delta_1, \lambda \heartsuit \iota_1 \lambda_1$	$[\lambda_1 \lambda \heartsuit, \lambda \heartsuit], \lambda \heartsuit = (\iota_1 \boxtimes \lambda_1)^*$

Theorem (soundness of the Thompson construction). For every policy p , $\llbracket \mathcal{A}_p \rrbracket = G(p)$, the language model of p .

Corollary. $\llbracket p \rrbracket(\pi_0)(\pi_n \cdots \pi_1) = \llbracket \mathcal{A}_p \rrbracket(\pi_0 \pi_1 \text{ dup} \cdots \text{dup} \pi_n)$.

Automata and decision procedures

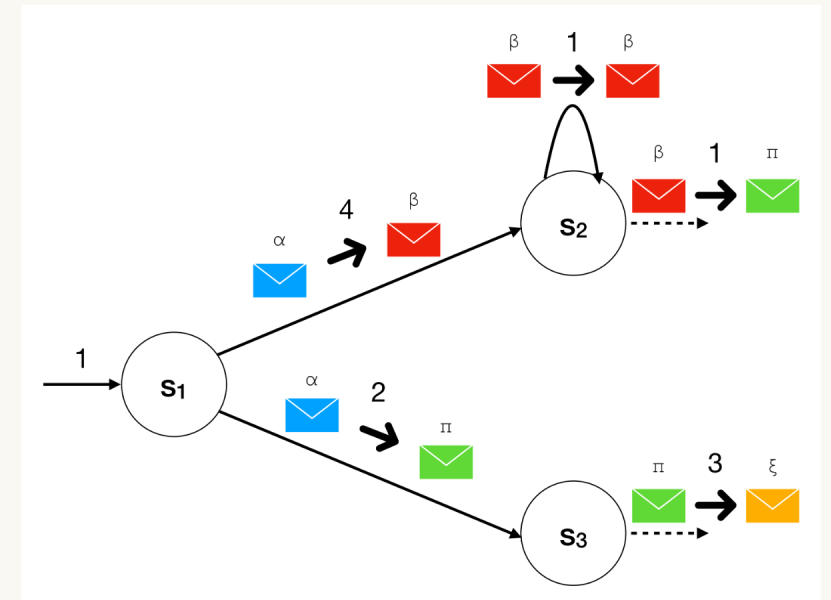
policy $p \rightarrow \text{compile} \rightarrow \text{automaton } A_p$, with $\llbracket p \rrbracket = \llbracket A_p \rrbracket$

r-safety: does all traffic have weight at most r ?

90% reliability? Abilene is 0.1-safe (probabilistic-union).

r-reachability: is there a path with weight at least r ?

LA to CHI in 10 ms? sw=LA ; abilene ; sw=CHI is 10-reachable (tropical).



Correct and general, but explicit automata do not scale!

Weighted symbolic packet programs

Weighted symbolic packet programs: decision diagrams with weights at the leaves.

Scalability of the explicit semantics

$$|P_k| = \prod |\text{dom}(f)|$$

The packet space is exponential in the number of fields, and $\llbracket p \rrbracket$ is a $P_k \times P_k$ matrix.

Boolean: solved

KATch uses symbolic packet programs, a decision-diagram view of NetKAT.

Weighted: this work

Keep the symbolic structure, put semiring weights at the leaves.

Running example: a retry protocol

```
sw=1 ; (  
st=rdy ; (  $\frac{1}{4} \odot$  sw $\leftarrow$ 2  $\oplus$   $\frac{1}{2} \odot$  st $\leftarrow$ wait  $\oplus$   $\frac{1}{4} \odot$  drop )  
 $\oplus$  st=wait ; (  $\frac{1}{2} \odot$  skip  $\oplus$   $\frac{1}{4} \odot$  st $\leftarrow$ rdy  $\oplus$   $\frac{1}{4} \odot$  drop ) )
```

Ready

Transmit $\frac{1}{4}$ · hold $\frac{1}{2}$ · drop $\frac{1}{4}$

Waiting

Keep waiting $\frac{1}{2}$ · ready $\frac{1}{4}$ · drop $\frac{1}{4}$

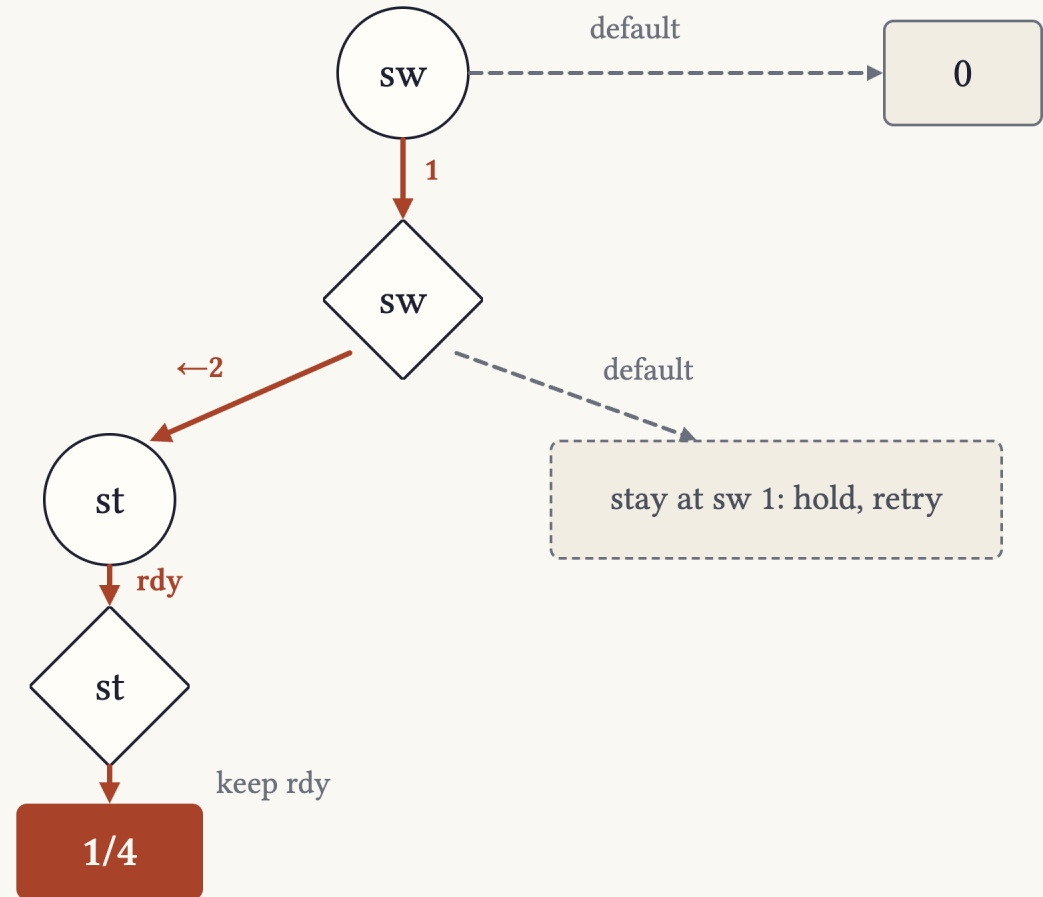
Weighted symbolic packet programs

Circle: branch on the input value of a field.

Diamond: choose the output value of that field.

Dashed: the default edge. All other values, or leave the field unchanged.

Leaf: the weight of the packet pair.



Reading weights from a wSPP

Input	Output	Weight
sw 1, ready	sw 2, ready	1/4
sw 1, ready	sw 1, wait	1/2
sw 1, wait	sw 1, ready	1/4
sw 1, wait	sw 2, ready	0

One slot is easy. *Until the packet is delivered* needs iteration: p^* .

Properties of wSPPs

Every weighted NetKAT operation has a direct construction on wSPPs:

t $f \leftarrow v$ $w \odot p$ $p \oplus q$ $p ; q$ p^* ← the difficult case: next

Compositional

$\text{wSPP}(p)$ is computed from the wSPPs of its parts, never from the $P_k \times P_k$ matrix.

Symbolic

Default edges stand for all remaining values, so size follows the policy, not $|P_k|$.

Parametric and verified

One construction for every semiring; correctness formalised in Lean.

Unbounded iteration

Unbounded iteration turns into infinite sums of weights.

Semantics of iteration

$$w^* = 1 + w + w^2 + \dots$$

$$w^* = 1 / (1 - w) \text{ for } w < 1$$

In the probability semiring, star on a single weight is the geometric series.

Eventually delivered

from ready: 1/3

from wait: 1/6

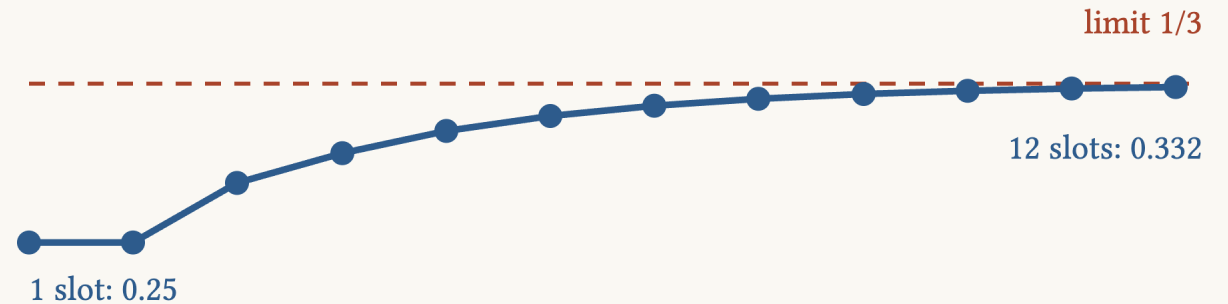
From wait: $\sum_j (\frac{1}{2})^j \cdot \frac{1}{4} \cdot \frac{1}{3} = 1/6$

Finite unfolding does not terminate

Booleans: KATch unfolds p^* until nothing changes.

The lattice is finite, so it stops.

Weights: each unfolding adds a little more. The partial sums only reach $1/3$ in the limit.



delivery probability from ready, after $n = 1 \dots 12$ slots

Requirements on the semiring

1

ω -continuous

Countable sums exist, so p^* means something.

2

Computable semiring star

Finite representations, decidable equality, and a computable $a \mapsto a^\oplus$.

3

An embedding

$\iota(a^\oplus) = \iota(a)^*$, so the fixpoint and denesting laws carry over.

Symbolic state elimination

Walk the wSPP. At each node, pull out the part that loops on itself.

Its star is a **single weight**, handed to the computable star.

Denest the rest and recurse. No unfolding, no approximation.

LAWS USED

$$p^* = 1 + p ; p^*$$

$$(a + b)^* = a^* (b a^*)^*$$

$$(1/3)^* = 3/2$$

Multi-objective analysis

Pareto frontiers, and the paths that achieve them, as one more semiring.

Latency and bandwidth

Tropical (min , +)

Latency. Choice keeps the faster branch; sequencing adds delays.

Bottleneck (max , min)

Bandwidth. Choice keeps the wider branch; sequencing throttles to the thinnest link.

Each answers its own question. Together, we need *every* non-dominated pair.

Pareto semirings

Carrier: finite antichains of vectors, such as (latency, bandwidth).

Star: iterate $G_0 = 1$, $G_{n+1} = 1 + F \cdot G_n$ until it stabilises.

Finiteness: stars of finitely generated sets stay finitely generated (Thm 5.3).

$$F + G = \max(F \cup G)$$

$$F \cdot G = \max \{ f \cdot g \}$$

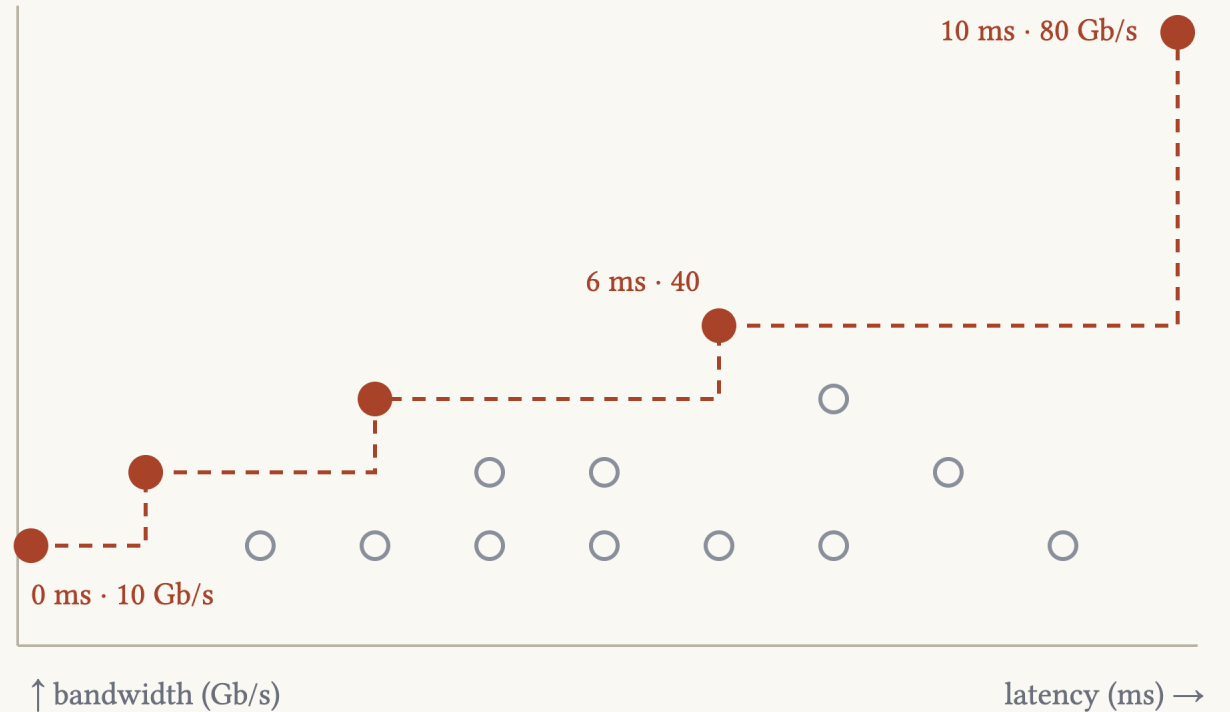
$\max$ keeps only the non-dominated vectors; $\cdot$ is componentwise.

Example: a bypass chain

Four ageing spans; span i carries $10 \cdot i$ Gb/s.

A detour around span i costs i ms over an 80 Gb/s fabric.

Gold traffic, switch 0 to 4: **5 Pareto-optimal** routes, 11 dominated.



Trace-carrying weights

A weight is a pair: (**cost**, **trace**). Weighting doubles as logging.

Trace order: a longer trace is dominated by any subsequence of it. The shortest witness wins.

Incomparable traces are both kept: one cost can have two witnesses.

Bypass chain, gold

(6 ms, 40 Gb/s)

adm · d₁ d₂ d₃

Lease the detours around spans 1 to 3.

Absorption (Theorem 5.5)

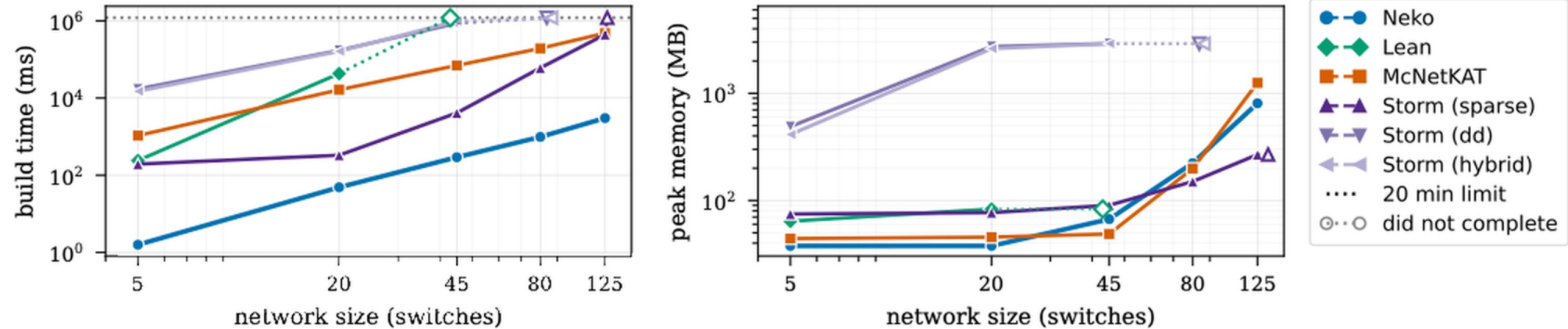
$$L \cdot L \leq 1 + L \implies L^* = 1 + L$$

A second trip round a loop costs more and only adds events, so it is dominated.

Star keeps the empty trace and one traversal. It is finite, *and* keeps the witness.

A bounded star ($L^* = 1$) would be finite too, but would erase every witness.

Implementation and performance



Probabilistic reachability: build time (left, log scale) and peak memory (right) against network size

2–3 orders of magnitude faster

Neko is fastest at every size: ≈ 1.5 ms vs ≈ 1 s (McNetKAT) at 5 switches; ≈ 3 s vs ≈ 8 min at 125.

Others time out

The Lean reference implementation and Storm (dd, hybrid) hit the 20-minute limit.

Comparable memory

Peak memory in line with McNetKAT and Storm (sparse), under 1 GB at 125 switches.

Fat-tree vs. Jellyfish: summary

−2 ms

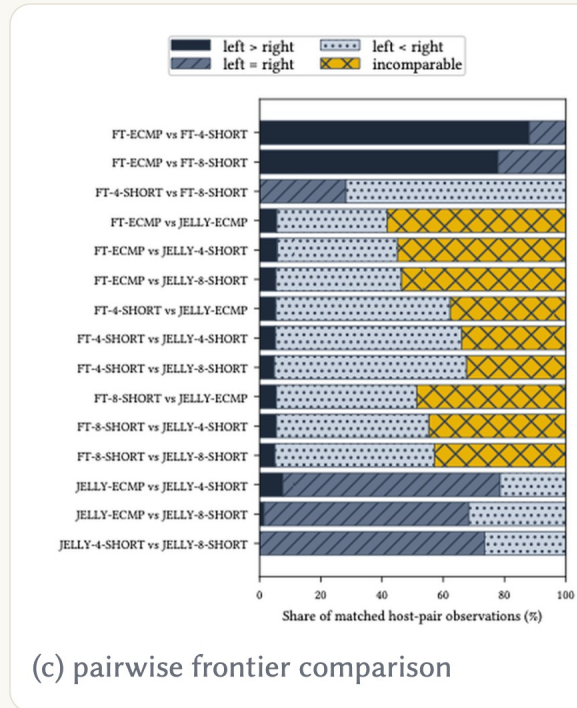
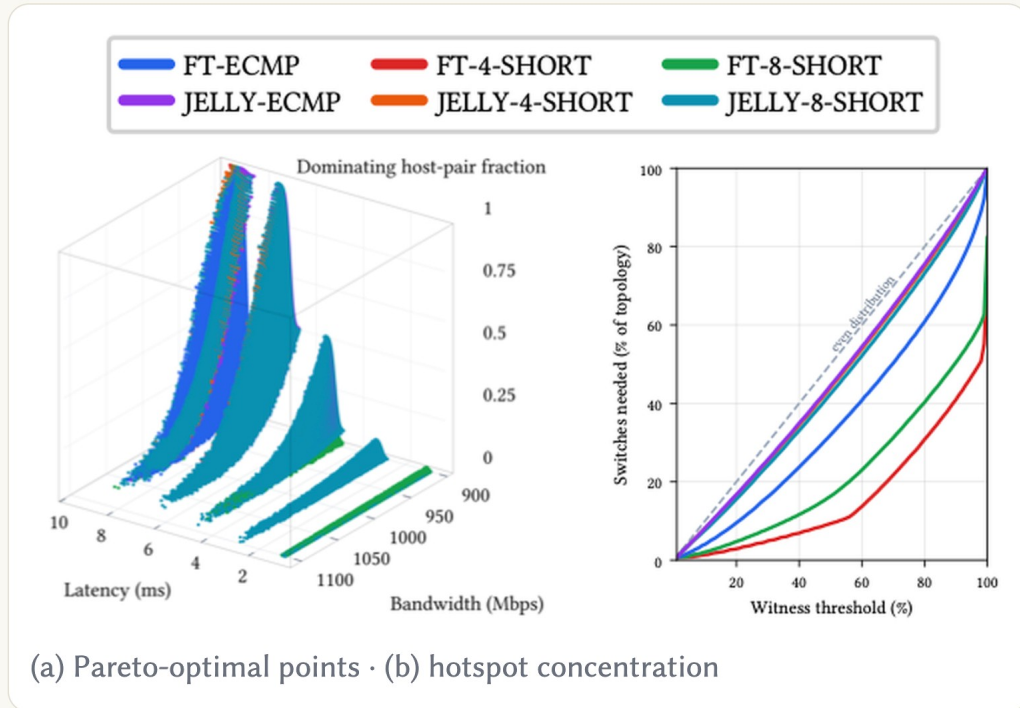
median latency: 6.85–6.92 ms on Jellyfish vs 8.81–8.88 ms on Fat-tree

≈ 930

Mbps median bandwidth on both: 920–932 Mbps across all six configurations

Topology + routing	Frontier size	Latency median	Latency mean	Bandwidth median	Bandwidth mean
Fat-tree · ECMP	2.47	8.81	8.56	932	936.5
Fat-tree · 4-SHORT	1.44	8.88	8.45	920	927.3
Fat-tree · 8-SHORT	1.88	8.86	8.52	925	930.9
Jellyfish · ECMP	1.38	6.85	6.27	927	933.2
Jellyfish · 4-SHORT	1.61	6.89	6.48	928	934.4
Jellyfish · 8-SHORT	1.82	6.92	6.72	932	936.7

Fat-tree vs. Jellyfish: Pareto frontiers



(a) Jellyfish frontiers sit at lower latency, in the same bandwidth band.

(b) Shortest-path routing on Fat-tree funnels optimal paths through few switches: **hotspots**.

(c) Fat-tree vs Jellyfish: Jellyfish wins for 40–60% of host pairs; most others are **incomparable**.

Conclusions

1

One language, semantics parametric in a semiring.

2

Symbolic diagrams plus an algebraic star make it fast.

3

Trade-offs and witnesses are just another semiring.

QUESTIONS FOR THIS AUDIENCE

A categorical home for trace-carrying Pareto semirings? Which constructions on semirings preserve a computable star?

Thank you

Weighted NetKAT · arXiv 2604.13987

A Fast Quantitative Analyzer for NetKAT · arXiv 2607.14420

Alexandra Silva · Cornell University