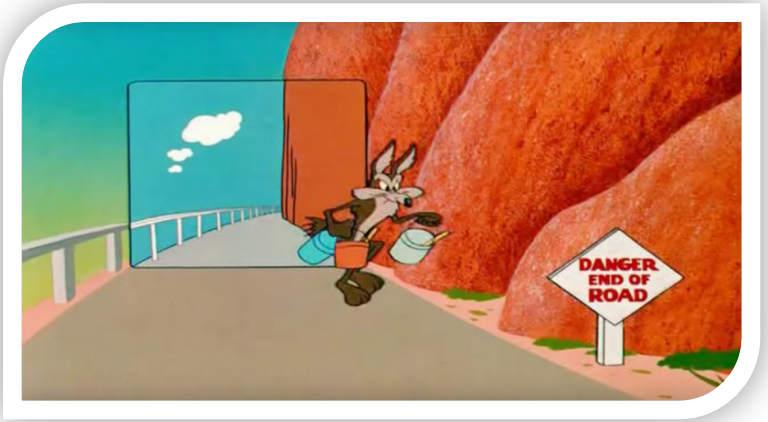
Lying Your Way to Better Traffic Engineering

Marco Chiesa

Université catholique de Louvain
Belgium



Joint work with
Gábor Rétvári and **Michael Schapira**

MŰEGYETEM 1782

האוניברסיטה העברית בירושלים
THE HEBREW UNIVERSITY OF JERUSALEM

Software-Defined-Networking (SDN)

SDN holds great promise

- for enhancing network performance
- for better manageability of the network

Software-Defined-Networking (SDN)

SDN holds great promise

- for enhancing network performance
- for better manageability of the network

Yet, transitioning to SDN is challenging

- requires new infrastructure 

Towards programmable legacy networks

Fibbing: **SDN-like** control over IP **legacy networks**

[SIGCOMM2015]

Benefits

- centralized, SDN-like control
- backwards compatible

Towards programmable legacy networks

Fibbing: **SDN-like** control over IP **legacy networks**

[SIGCOMM2015]

Benefits

- centralized, SDN-like control
- backwards compatible

Inherent constraints

- **destination-based** (IP) routing
- **limited** network measurement capabilities

Our goal: Enhancing traditional TE

Traffic Engineering (TE)

- tuning routing protocol parameters to optimize traffic flow
- traditional TE a notoriously **cumbersome** and **inefficcient**

Can we leverage SDN-like control to better TE?

Our goal: Enhancing traditional TE

Traffic Engineering (TE)

- tuning routing protocol parameters to optimize traffic flow
- traditional TE a notoriously **cumbersome** and **inefficcient**

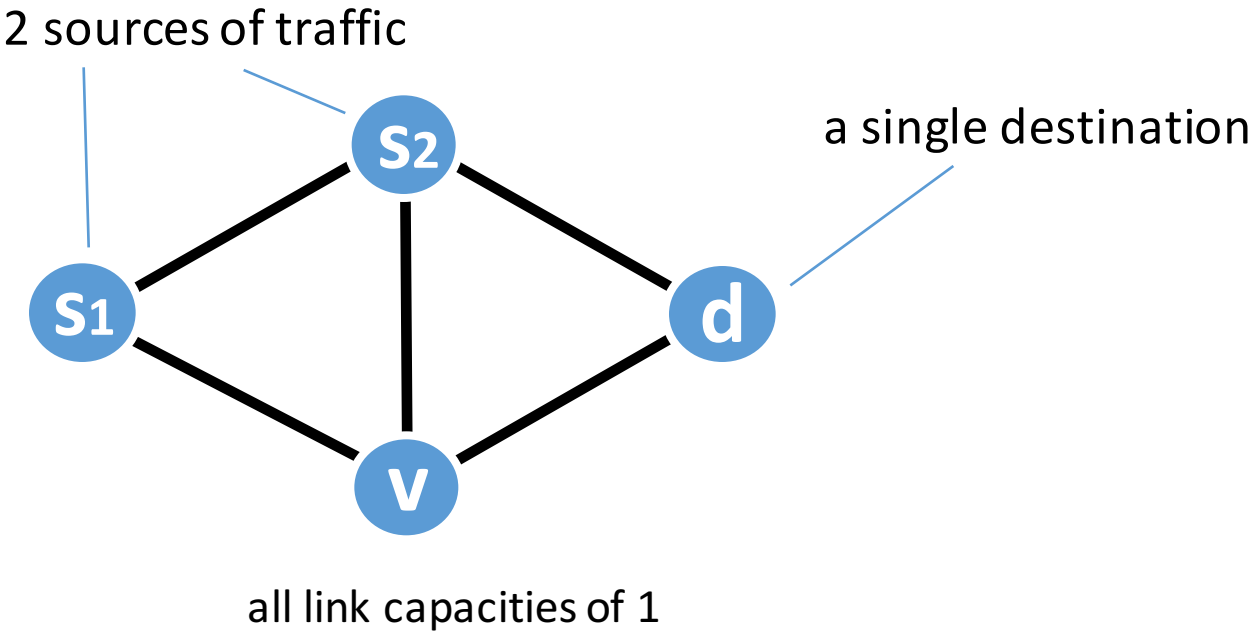
Can we leverage SDN-like control to better TE? **Yes!**

COYOTE

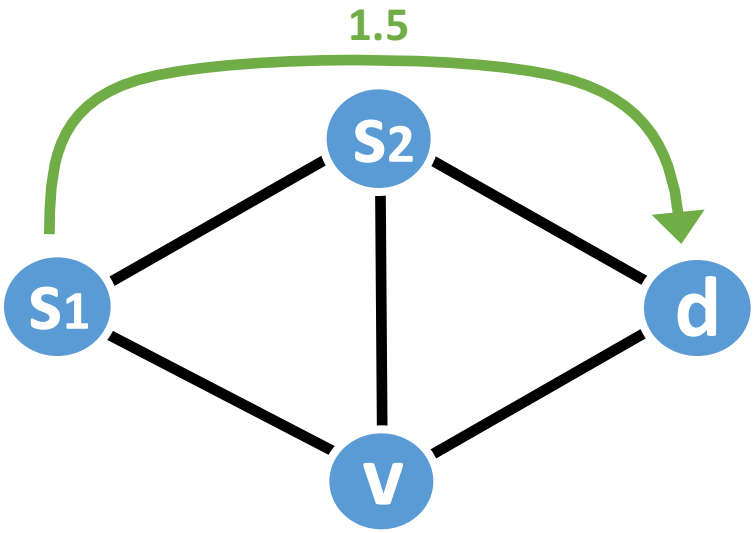
Compatible **O**blivious **Y**et **O**ptimized **T**raffic **E**ngineering

- a novel approach to TE in **legacy** networks
- **assumes limited/no knowledge** about prevailing traffic demands
- **significantly improves** network performance

Motivating example



Motivating example: two possible traffic scenarios

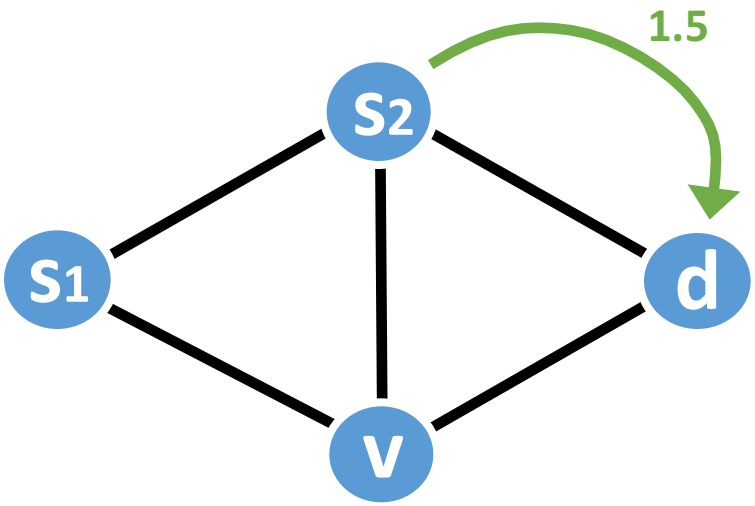


all link capacities of 1

Only two possible demand matrices:

1. **only $s_1 \rightarrow d = 1.5$**
2. only $s_2 \rightarrow d = 1.5$

Motivating example: two possible traffic scenarios

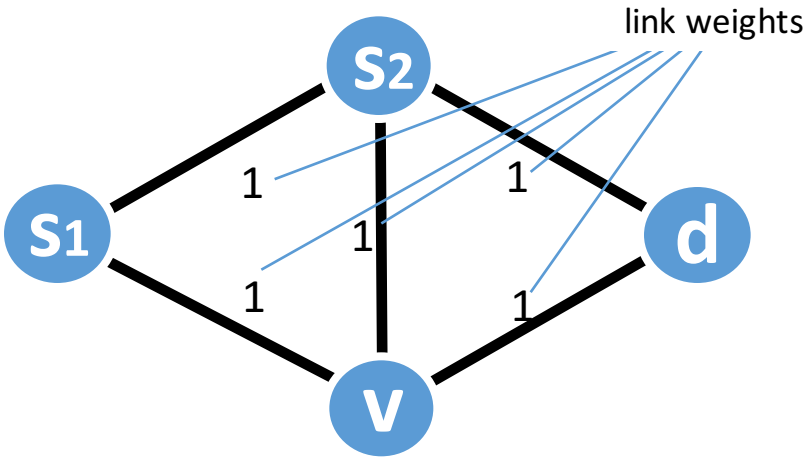


all link capacities of 1

Only two possible demand matrices:

1. only $s_1 \rightarrow d = 1.5$
2. **only $s_2 \rightarrow d = 1.5$**

Motivating example
Traditional TE



all link capacities of 1

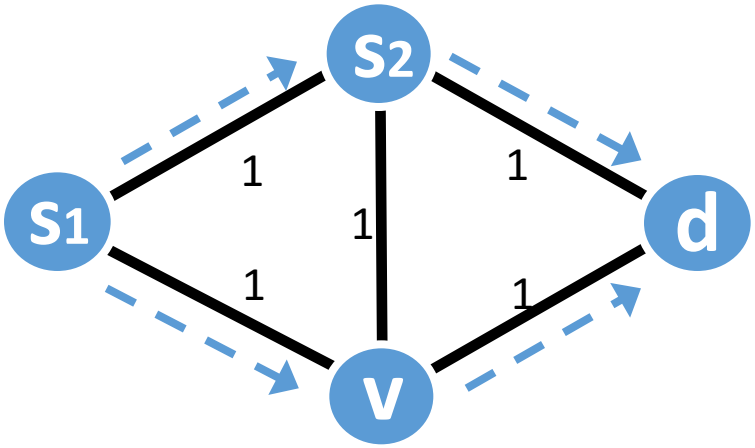
Only two possible demand matrices:

1. only $s_1 \rightarrow d = 1.5$
2. only $s_2 \rightarrow d = 1.5$

Traditional TE (with OSPF/ECMP):

- operator sets link weights

Motivating example
Traditional TE



all link capacities of 1

shortest path DAG

Only two possible demand matrices:

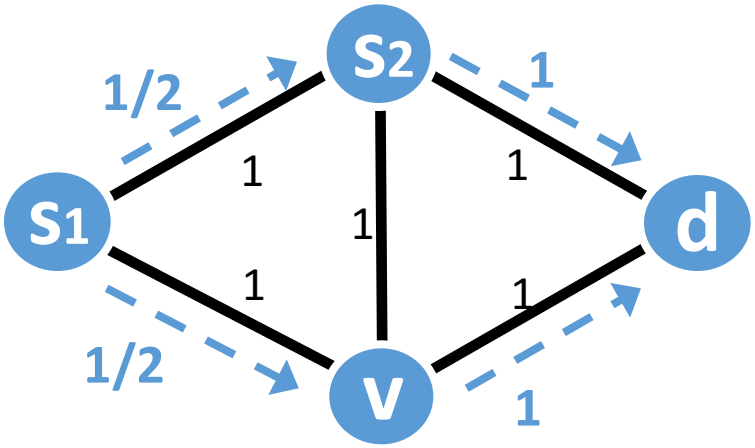
1. only $s_1 \rightarrow d = 1.5$
2. only $s_2 \rightarrow d = 1.5$

Traditional TE (with OSPF/ECMP):

- operator sets link weights
- per-destination routing
- **shortest paths DAGs**

directed acyclic graph

Motivating example
Traditional TE



all link capacities of 1

shortest path DAG

1/2 splitting ratio

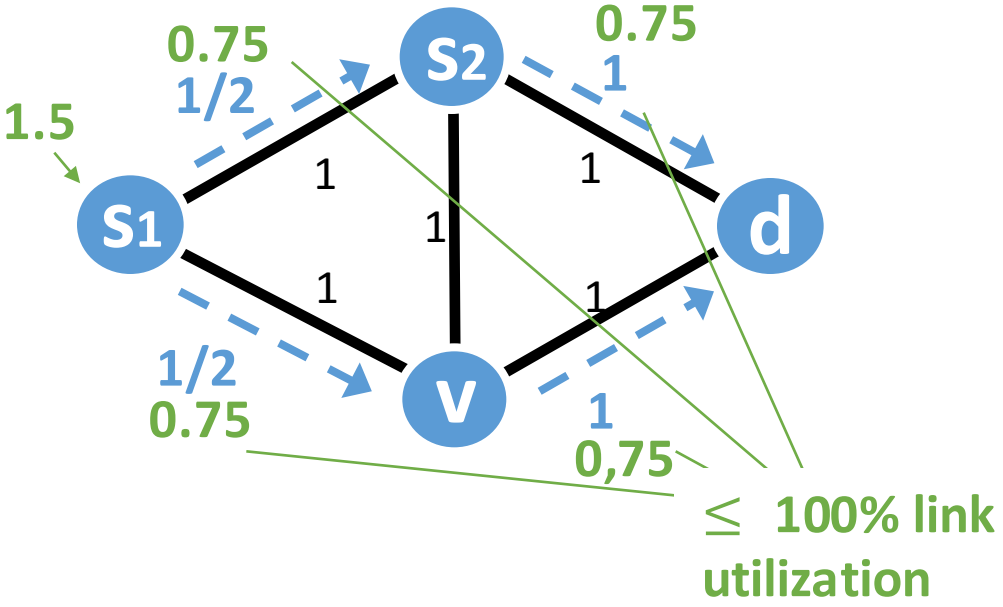
Only two possible demand matrices:

1. only $s_1 \rightarrow d = 1.5$
2. only $s_2 \rightarrow d = 1.5$

Traditional TE (with OSPF/ECMP):

- operator sets link weights
- per-destination routing
- shortest paths DAGs
- **equal-split**

Motivating example
Traditional TE



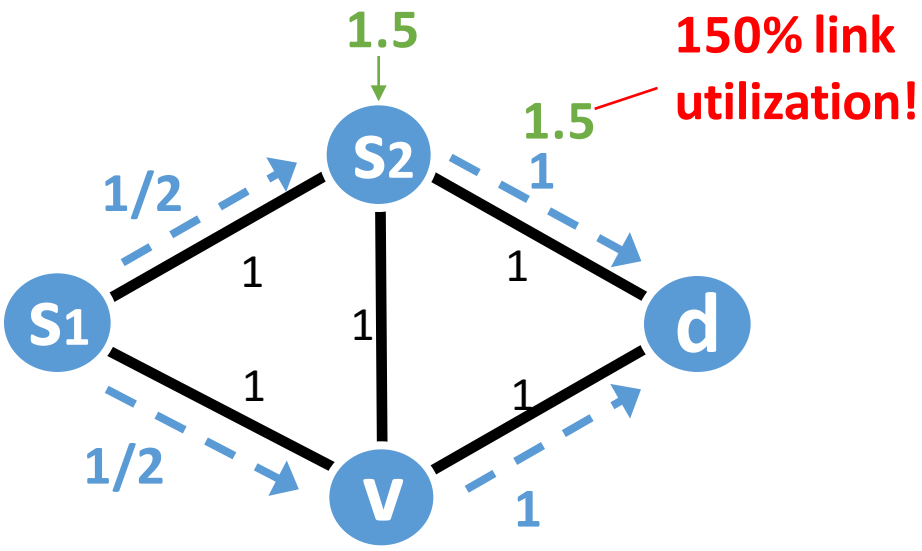
Only two possible demand matrices:

1. **only $s_1 \rightarrow d = 1.5$**
2. only $s_2 \rightarrow d = 1.5$

Traditional TE (with OSPF/ECMP):

- operator sets link weights
- per-destination routing
- shortest paths DAGs
- equal-split

Motivating example
Traditional TE



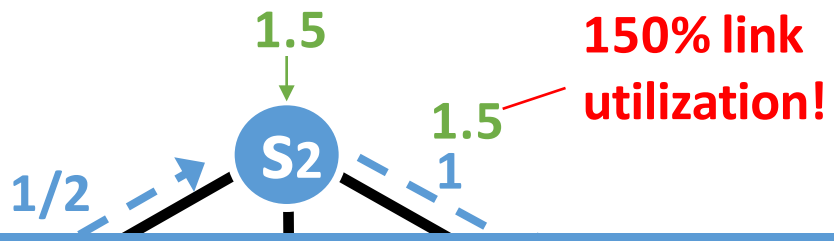
Only two possible demand matrices:

1. only $s_1 \rightarrow d = 1.5$
2. only $s_2 \rightarrow d = 1.5$

Traditional TE (with OSPF/ECMP):

- operator sets link weights
- per-destination routing
- shortest paths DAGs
- equal-split

Motivating example
Traditional TE



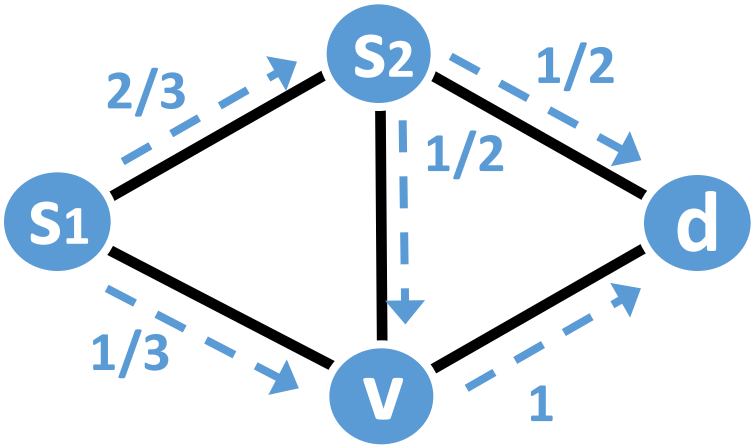
Only two possible demand matrices:

1. only $s_1 \rightarrow d = 1.5$
2. only $s_2 \rightarrow d = 1.5$

No link-weight assignement can attain
$\leq$ 100% link utilization!
(for both demand matrices)

- per-destination routing
- shortest paths DAGs
- equal-split

Motivating example
Better legacy-compatible TE



Operator **leverages SDN-like control**

to configure:

- **arbitrary** **per-destination** DAGs
- **arbitrary** splitting ratios

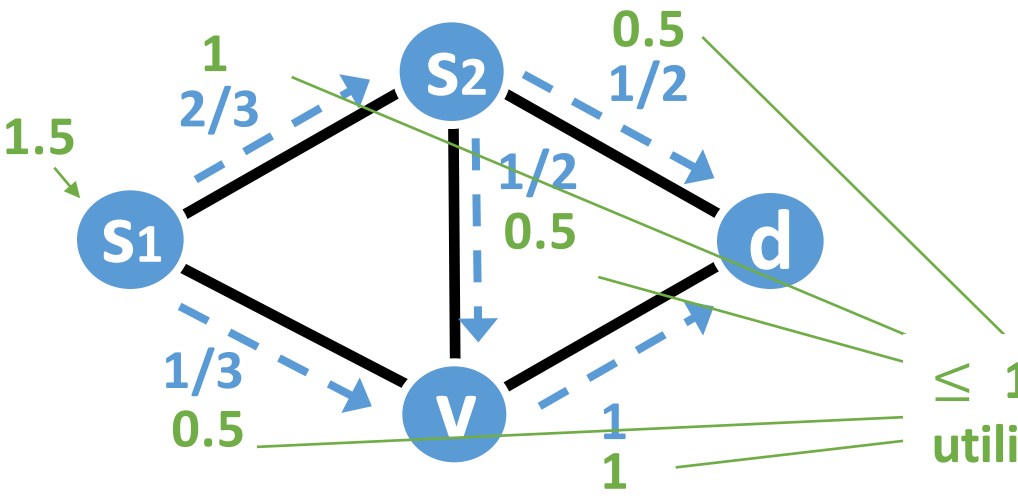
Only two possible demand matrices:

1. only $s_1 \rightarrow d = 1.5$
2. **only $s_2 \rightarrow d = 1.5$**

Traditional TE (with OSPF/ECMP):

- operator sets link weights
- per-destination routing
- shortest paths DAGs
- equal-split

Motivating example
Better legacy-compatible TE



Only two possible demand matrices:

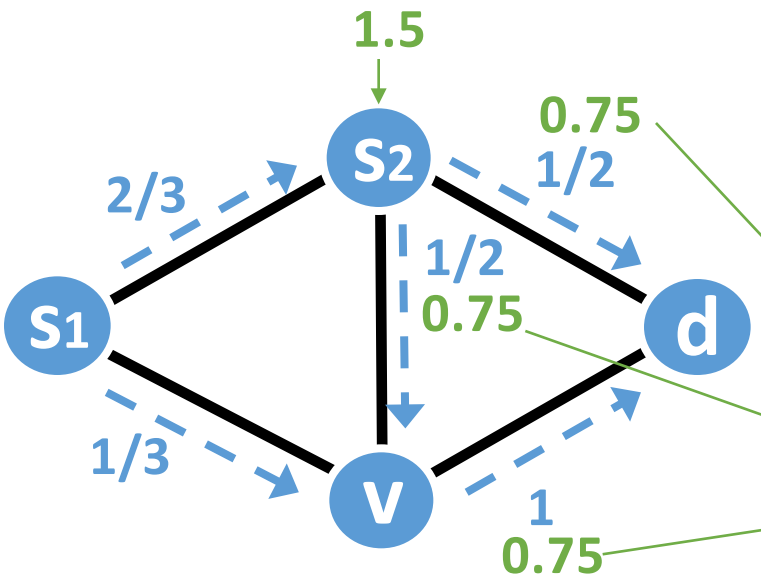
1. **only $s_1 \rightarrow d = 1.5$**
2. only $s_2 \rightarrow d = 1.5$

Operator **leverages SDN-like control**

to configure:

- **arbitrary** **per-destination** DAGs
- **arbitrary** splitting ratios

Motivating example
Better legacy-compatible TE



Only two possible demand matrices:

1. only $s_1 \rightarrow d = 1.5$
2. **only $s_2 \rightarrow d = 1.5$**

$\leq$ 100% link utilization

Operator **leverages SDN-like control**

to configure**:**

- **arbitrary** **per-destination** DAGs
- **arbitrary** splitting ratios

Uncharted algorithmic space!

Algorithmic challenge:

Input: network topology +

set of possible demand matrices

Uncharted algorithmic space!

Algorithmic challenge:

Input: network topology +

set of possible demand matrices

Output: per-destination DAGs +

in-DAG splitting ratios that

optimize network performance (min. link utilization)

Uncharted algorithmic space!

Algorithmic challenge:

Input: network topology +

set of possible demand matrices

Output: per-destination DAGs +

in-DAG splitting ratios that

optimize network performance (min. link utilization)

Relates to rich body of research on "oblivious routing" [Applegate02][Racke08]**,**

but…

Source-destination-based routing	Destination-based routing
Efficient (polytime)	?

Uncharted algorithmic space!

Algorithmic challenge:

Input: network topology +

set of possible demand matrices

Output: per-destination DAGs +

in-DAG splitting ratios that

optimize network performance (min. link utilization)

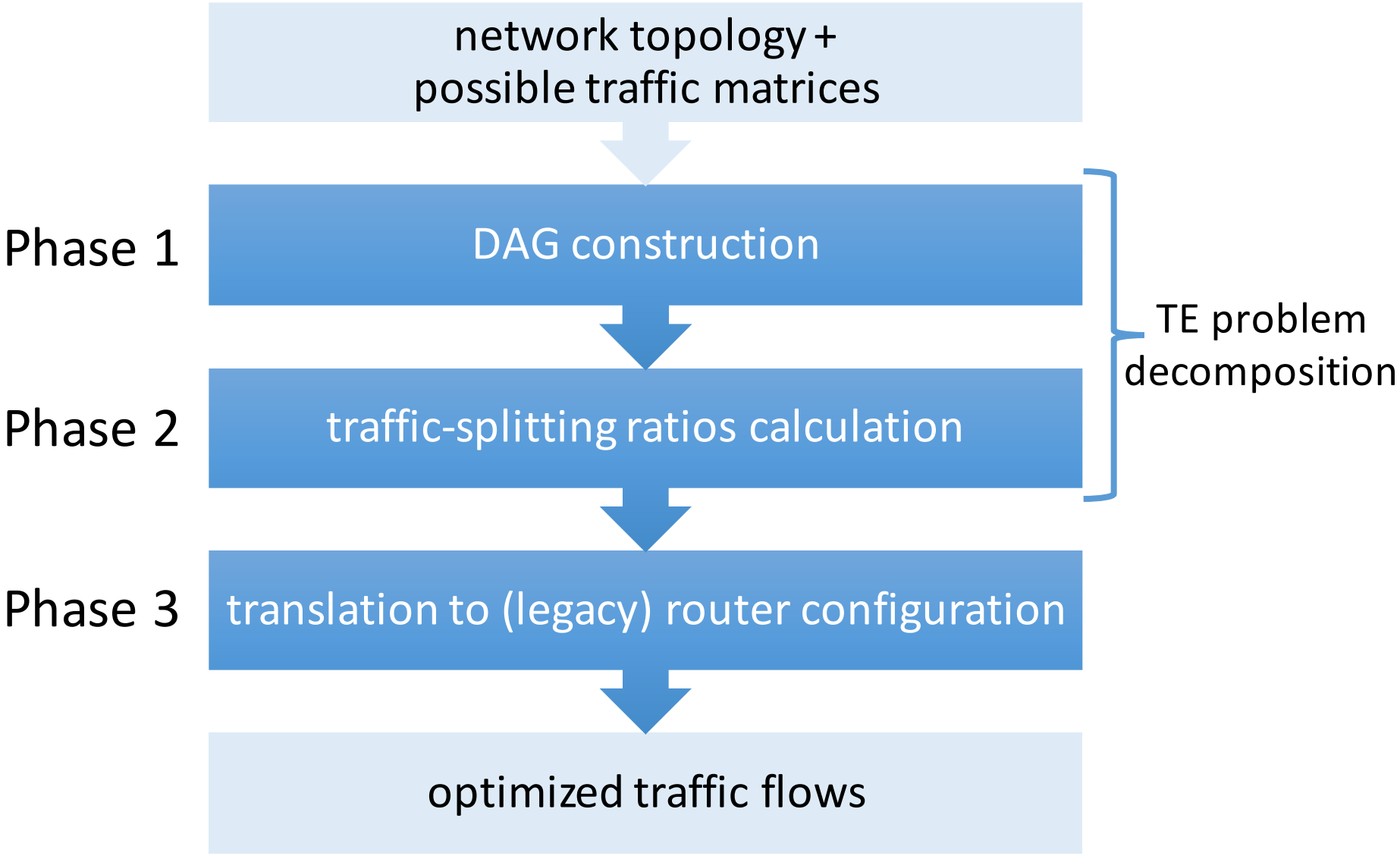
Relates to rich body of research on "oblivious routing" [Applegate02][Racke08],

but…

Source-destination-based routing	Destination-based routing
Efficient (polytime)	**Intractable (NP-hard)**

even for just two demand matrices, two sources, and a single destination!

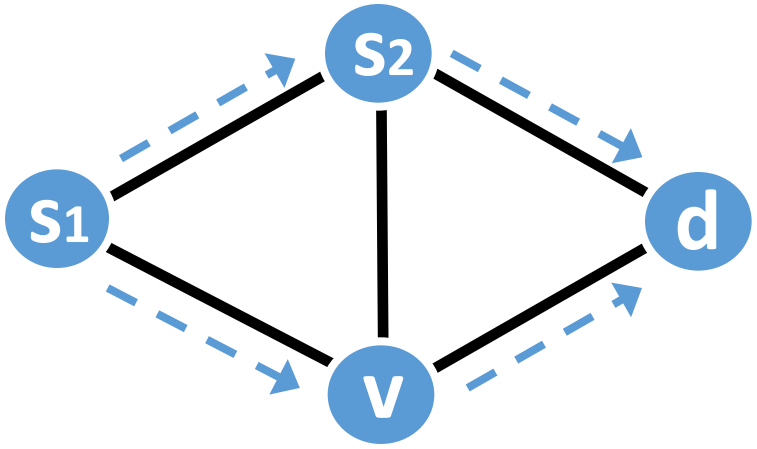
COYOTE architecture



COYOTE: step-by-step

Phase 1. DAG construction

- **Step 1.** Generate shortest-path DAG (e.g., via local search over link weights [Amit2006])

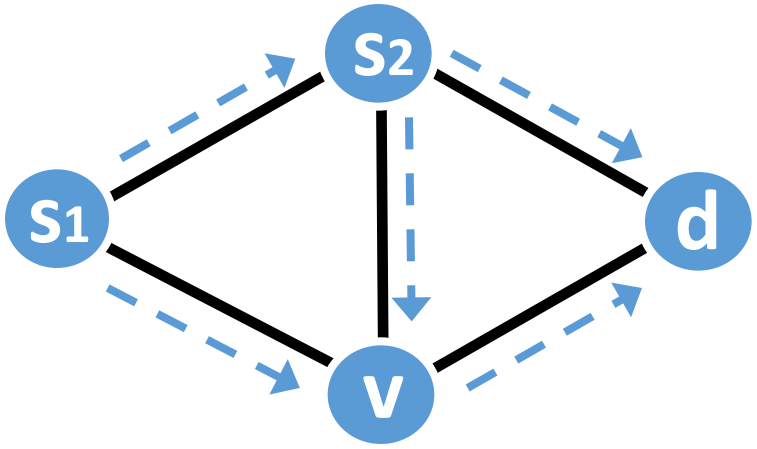


shortest path DAG

COYOTE: step-by-step

Phase 1. DAG construction

- **Step 1.** Generate shortest-path DAG (e.g., via local search over link weights [Amit2006])
- **Step 2.** DAG augmentation



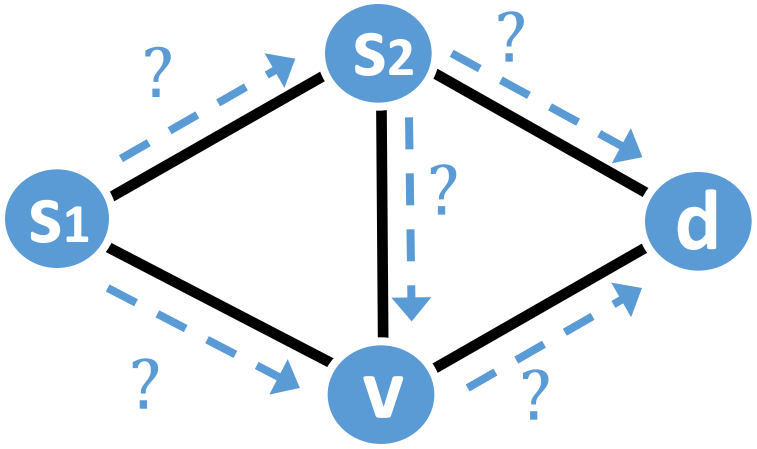
shortest path DAG

COYOTE: step-by-step

Phase 1. DAG construction

Phase 2. Traffic splitting ratio calculation

Crucial: How to split traffic with traffic uncertainty?



shortest path DAG

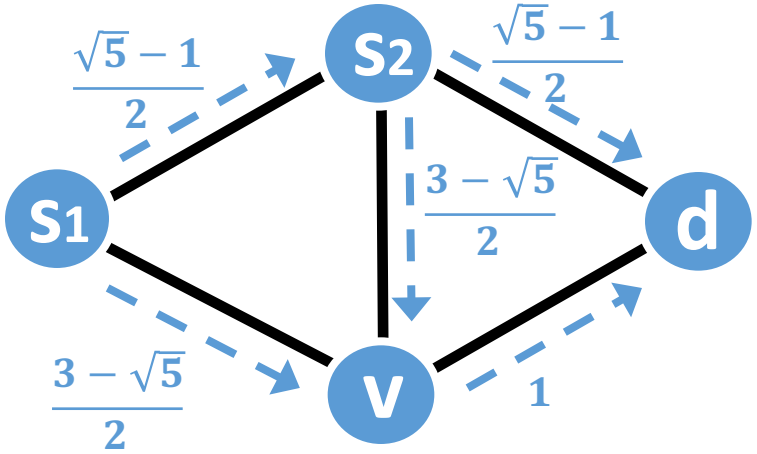
COYOTE: step-by-step

Phase 1. DAG construction

Phase 2. Traffic splitting ratio calculation

- **Step 1.** Compute optimal in-DAG traffic-splitting ratios
  - ➢ robust to traffic uncertainty
  - ➢ leverages **dualization theory** and **mixed linear-geometric programming**

Crucial: How to split traffic with traffic uncertainty?



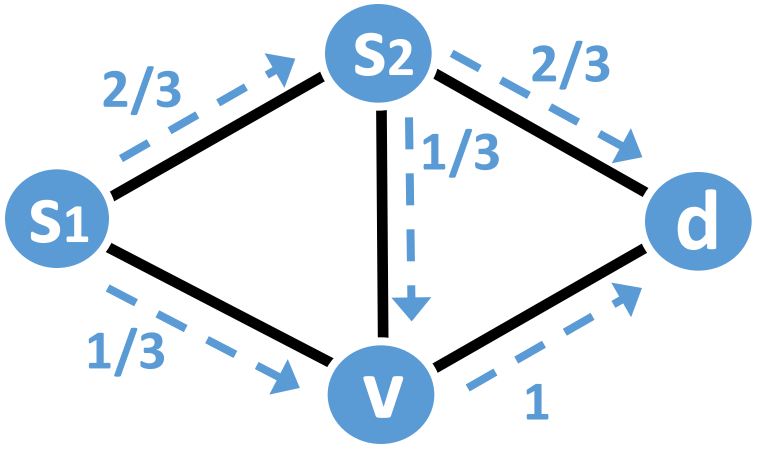
shortest path DAG

COYOTE: step-by-step

Phase 1. DAG construction

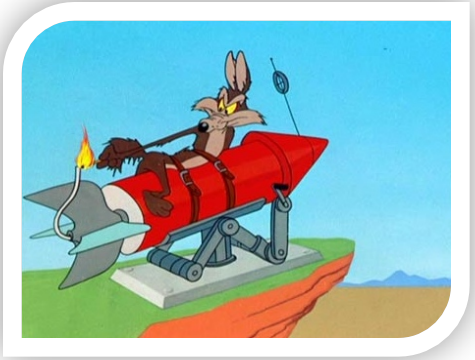
Phase 2. Traffic splitting ratio calculation

- **Step 1.** Compute optimal in-DAG traffic-splitting ratios
  - ➢ robust to traffic uncertainty
  - ➢ leverages dualization theory and mixed linear-geometric programming
- **Step 2.** Approximate splitting ratios [Nemeth2013]



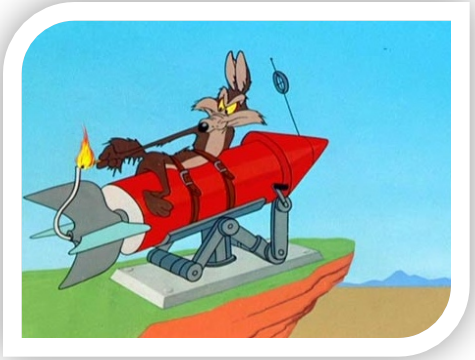
shortest path DAG

Evaluation



- 16 Internet backbone Rocketfuel topologies
- traffic uncertainty:
  - base TM (gravity/bimodal) + “uncertainty margins”

Evaluation

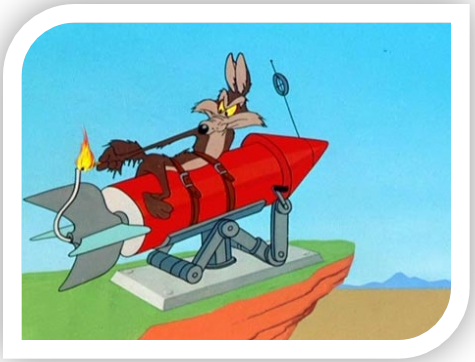


- 16 Internet backbone Rocketfuel topologies
- traffic uncertainty:
  - base TM (gravity/bimodal) + “uncertainty margins”

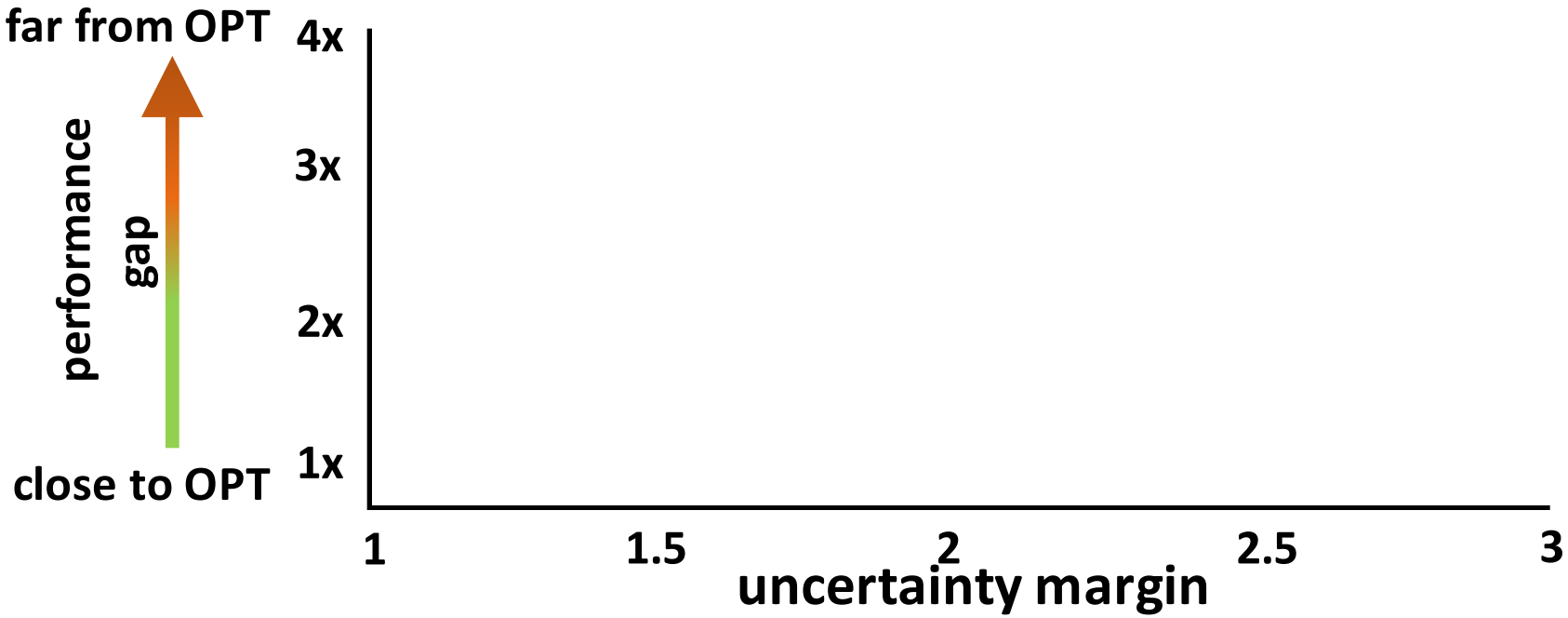
1 **1.5** **2** **2.5** **3**

no uncertainty **uncertainty margin** **high uncertainty**

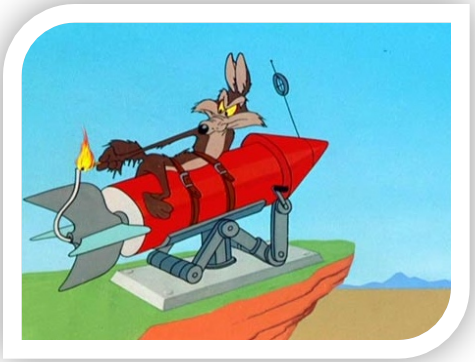
Evaluation



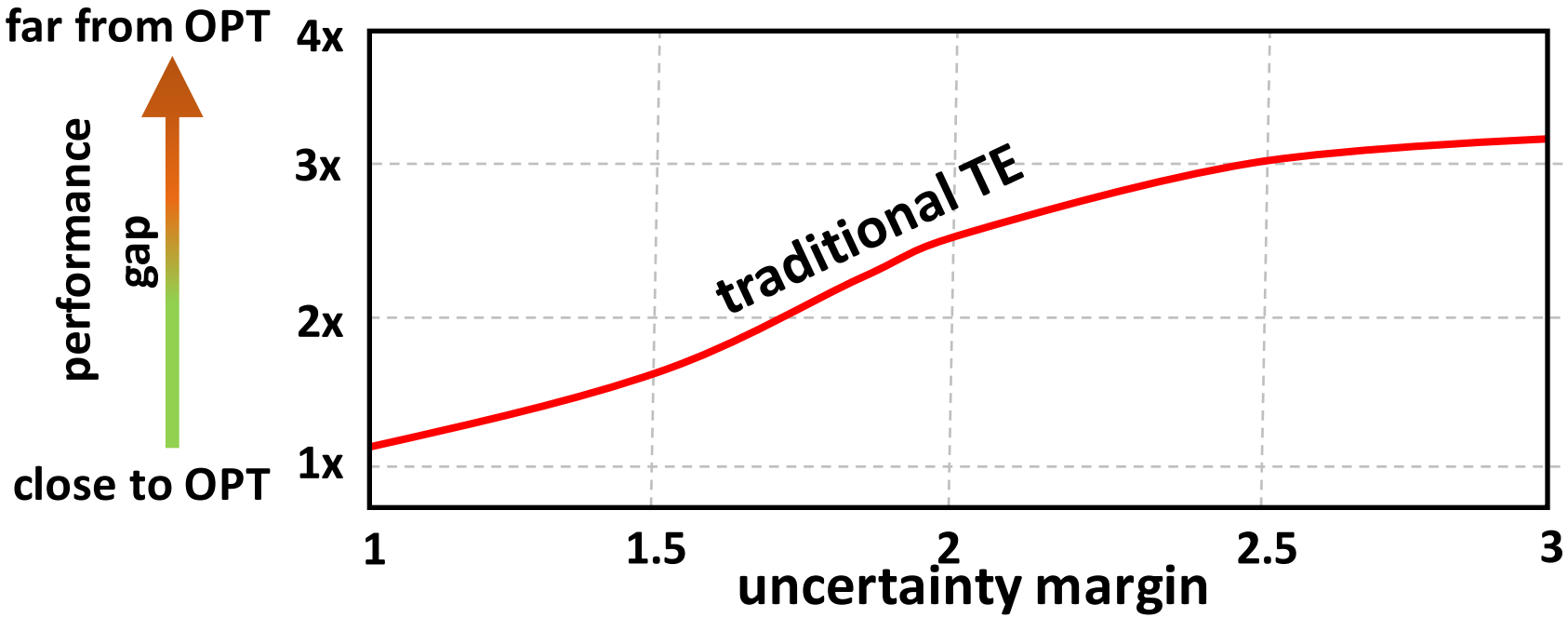
- 16 Internet backbone Rocketfuel topologies
- traffic uncertainty:
  - base TM (gravity/bimodal) + “uncertainty margins”



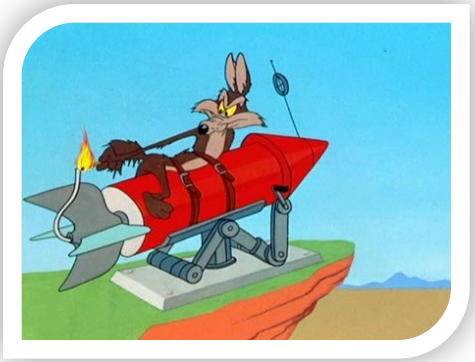
Evaluation



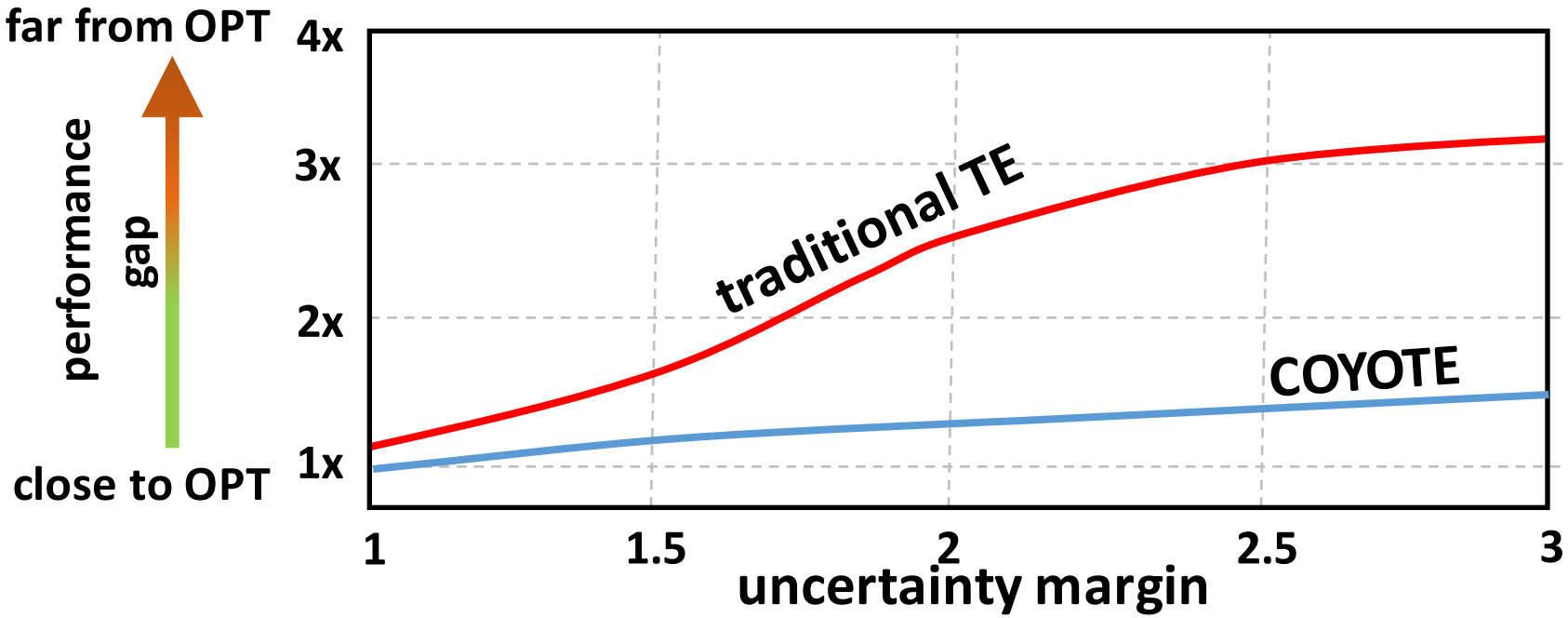
- 16 Internet backbone Rocketfuel topologies
- traffic uncertainty:
  - base TM (gravity/bimodal) + “uncertainty margins”



Evaluation



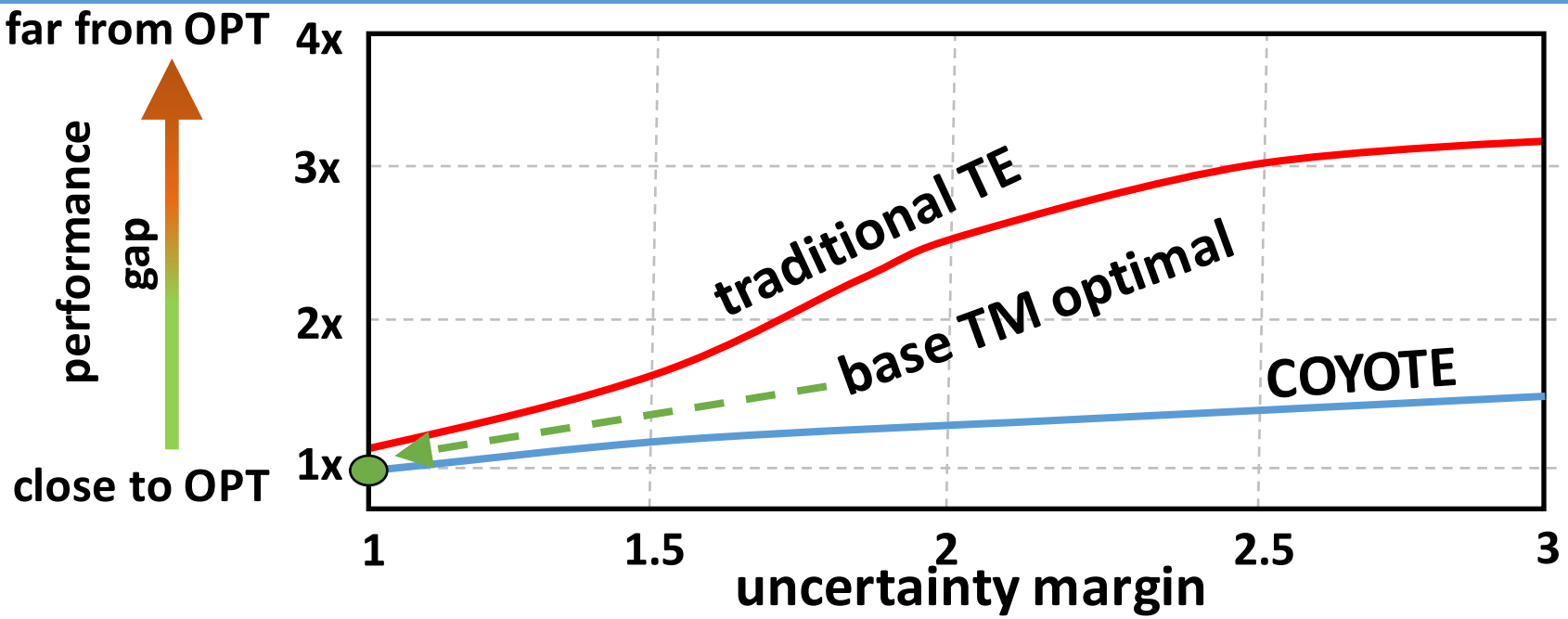
- 16 Internet backbone Rocketfuel topologies
- traffic uncertainty:
  - base TM (gravity/bimodal) + “uncertainty margins”



Evaluation

- 16 Internet backbone Rocketfuel topologies
- traffic uncertainty:

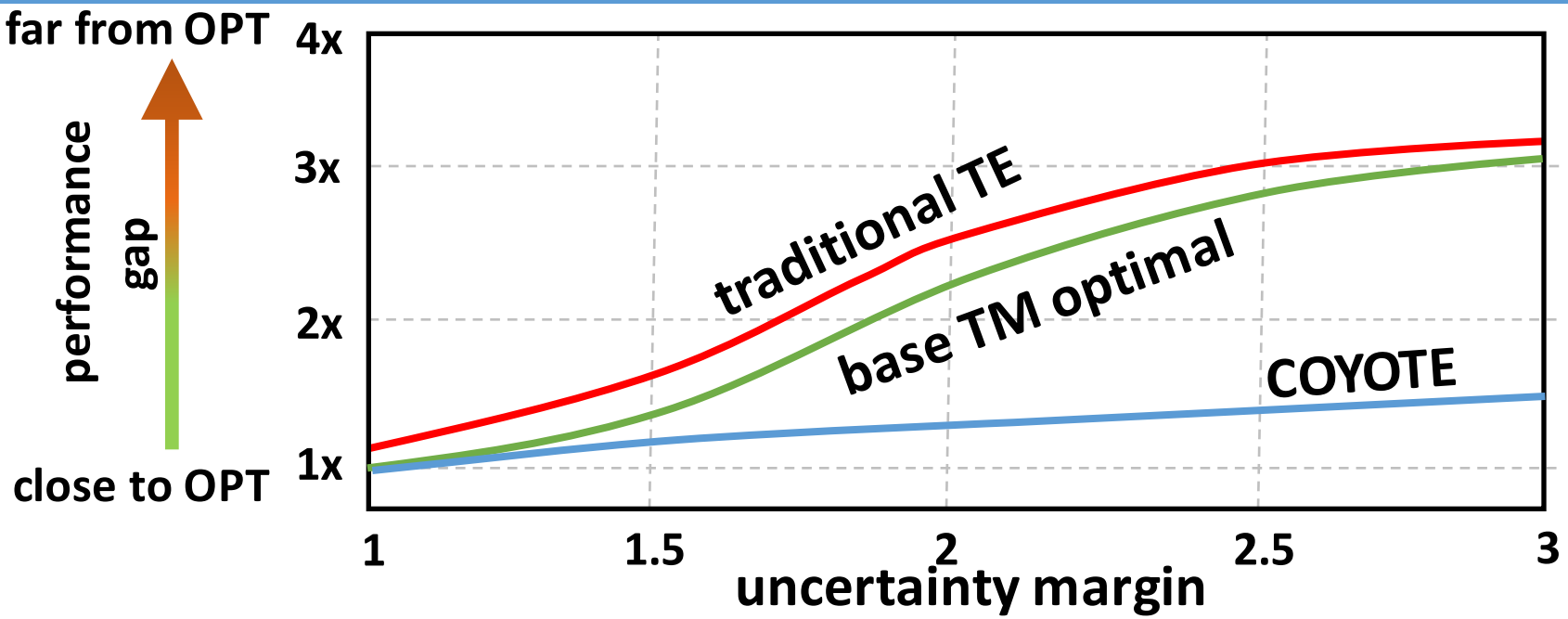
Is arbitrary-splitting alone the game-changer?



Evaluation

- 16 Internet backbone Rocketfuel topologies
- traffic uncertainty:

Is arbitrary-splitting alone the game-changer? No!



See paper for

- results for path-stretch
- results for approximating splitting ratios
- **experiments with prototype implementation**

Conclusions

- **novel** approach to TE in legacy networks
  - leverage SDN-like control
- **novel** algorithmic framework
  - NP-hard!
  - dualization theory + geometric programming approach
- significant **improvements** in performance upon traditional TE
- **important** application of the SDN approach to legacy networks

COYOTE

- **novel** approach to TE in legacy networks
- **novel** algorithmic framework
- significant **improvements** in performance upon traditional TE
- **important** application of the SDN approach to legacy networks

That's all folks!

Marco Chiesa

Université catholique de Louvain
Belgium

Joint work with
Gábor Rétvári and **Michael Schapira**