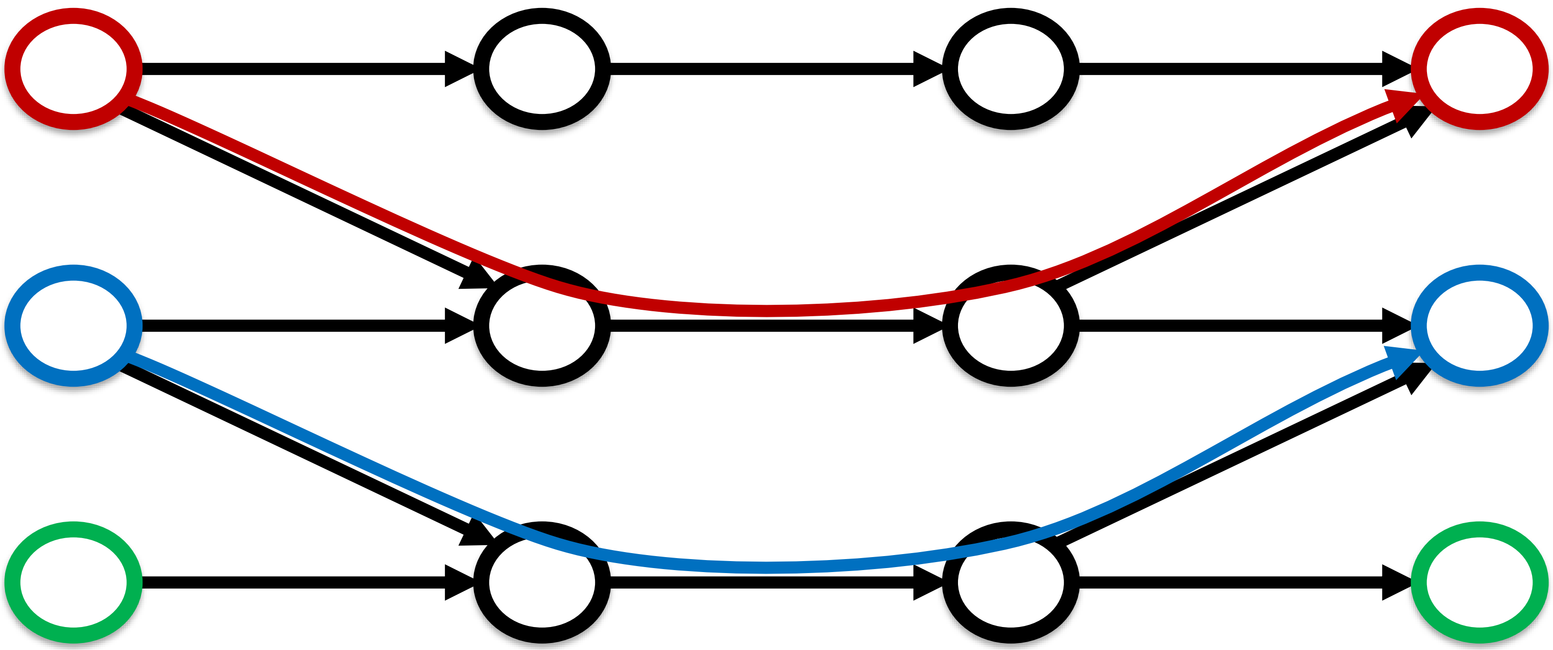
Don't disturb my Flows: Consistent Migration of Flows in SDNs

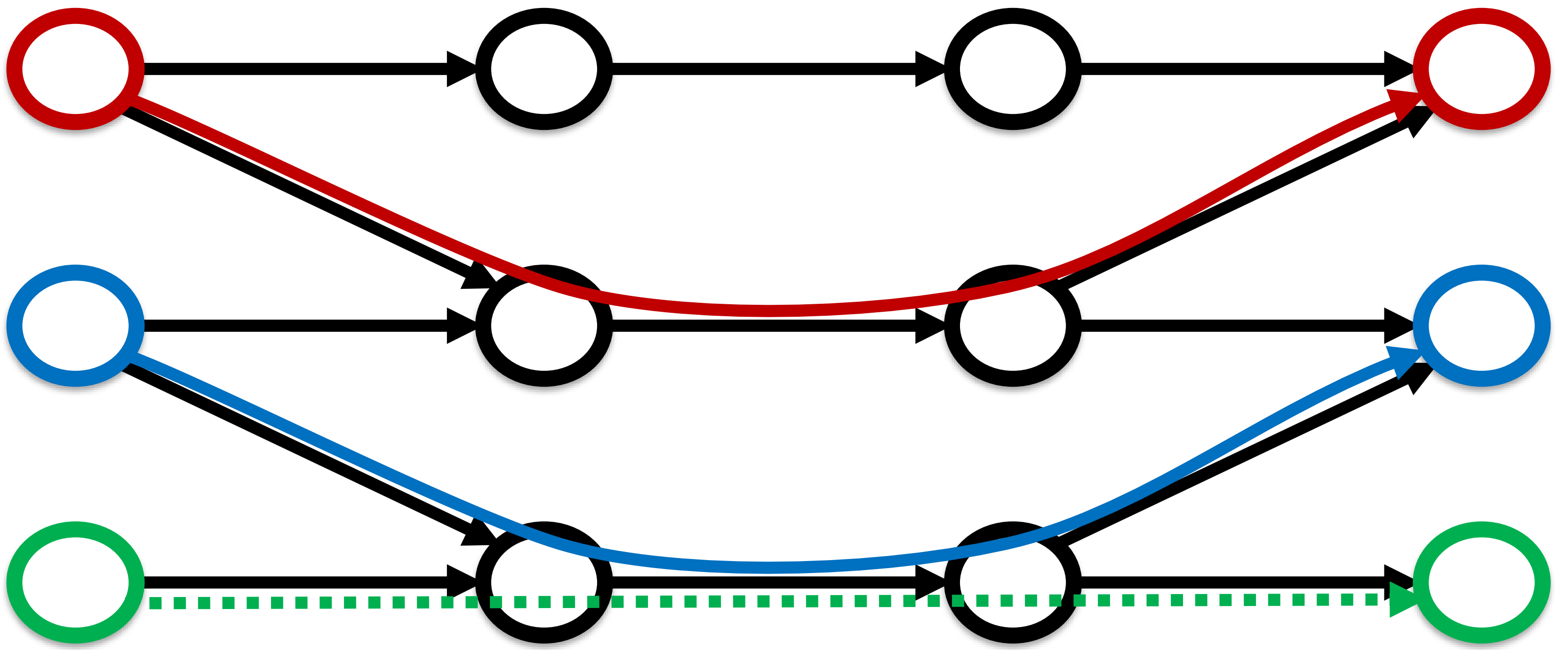
Klaus-Tycho Förster

Joint work with Sebastian Brandt, Laurent Vanbever, and Roger Wattenhofer

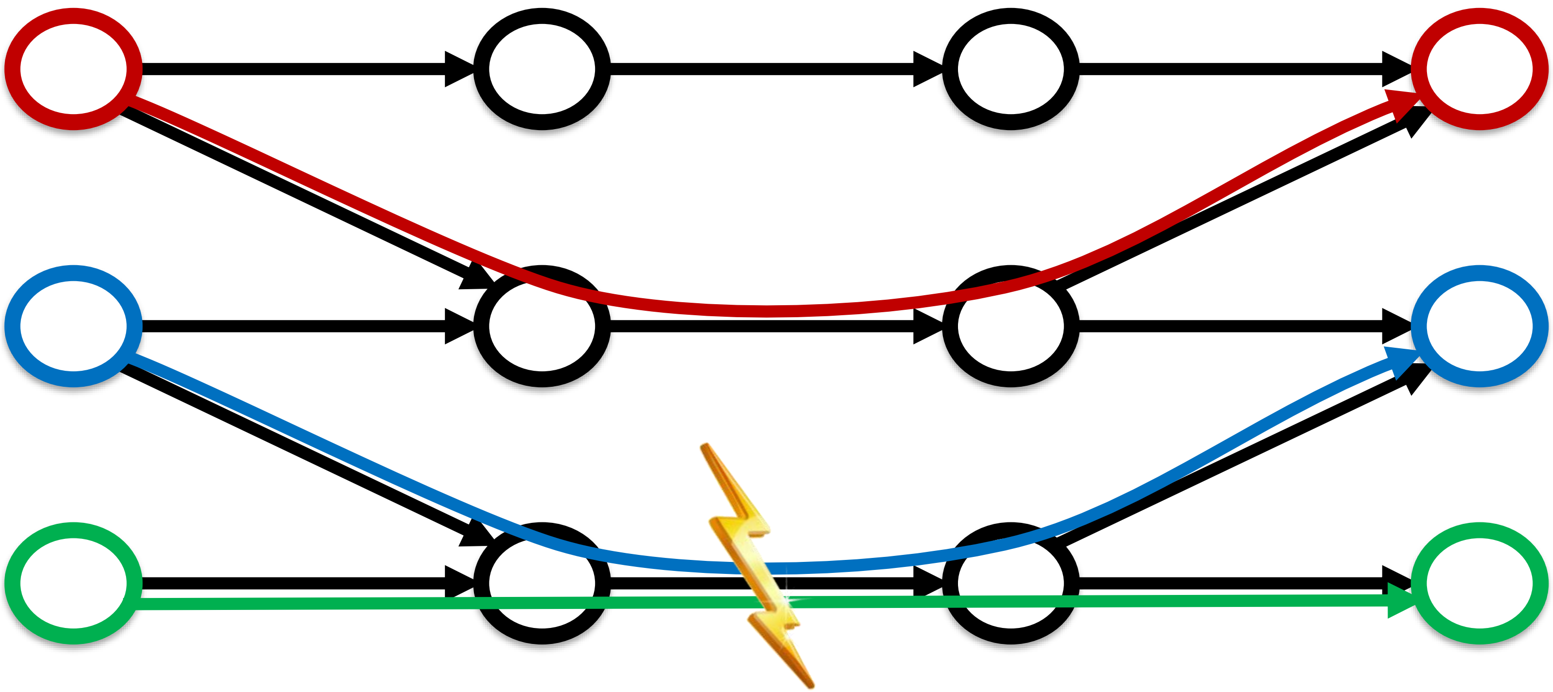
A Small Sample Network



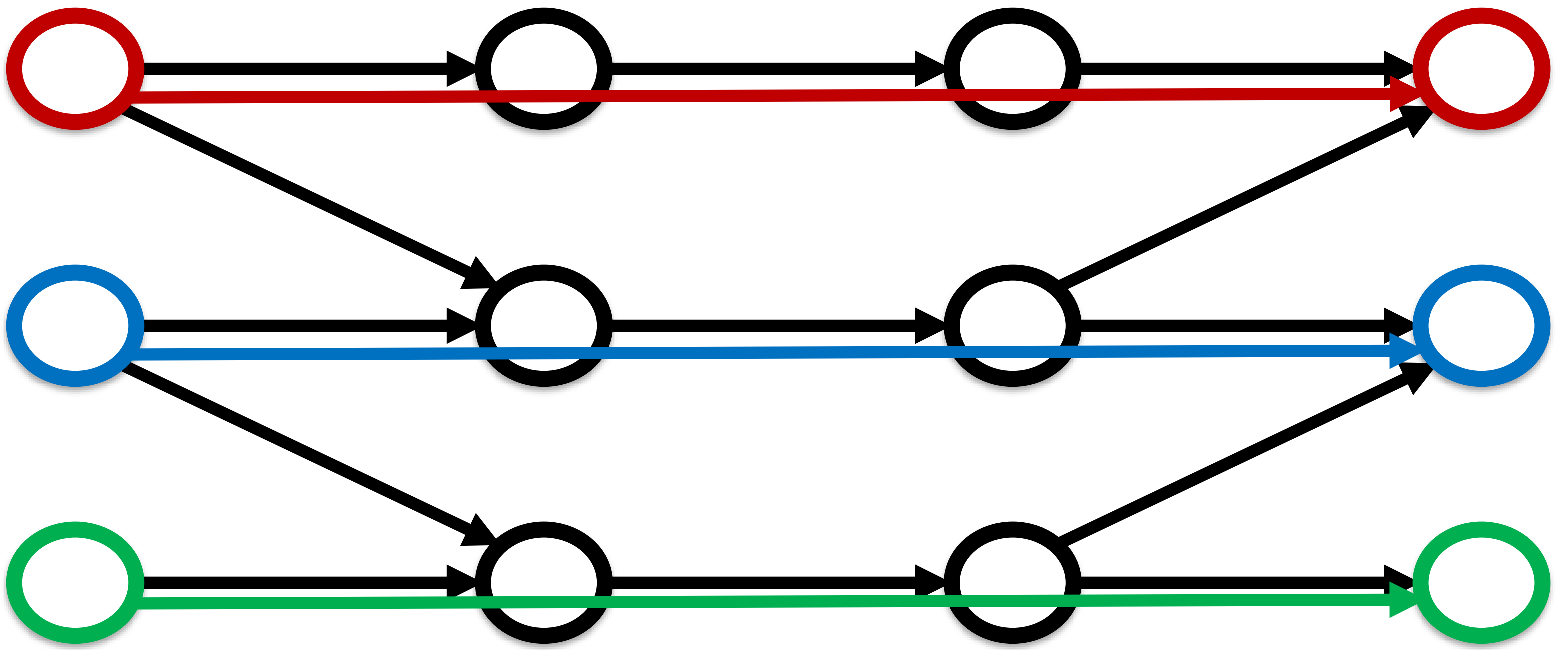
Green wants to send as well



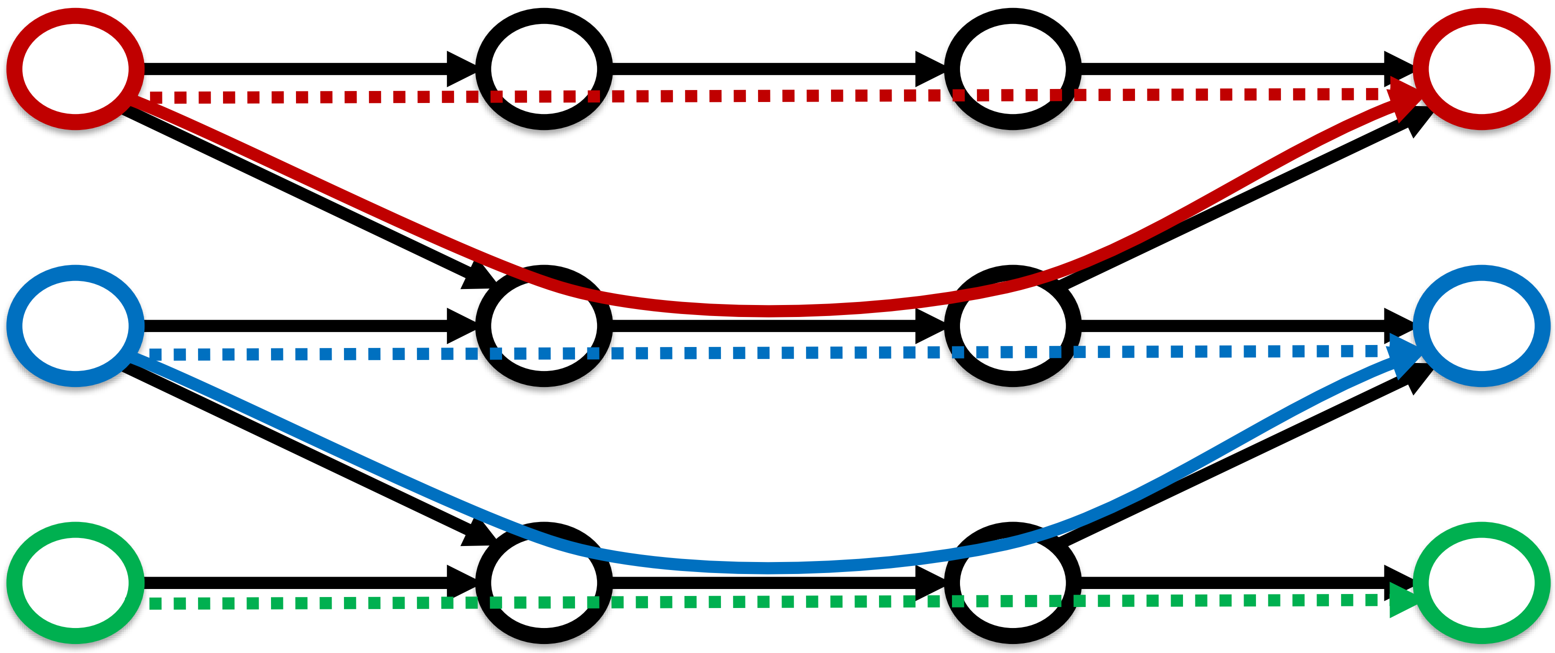
Congestion!



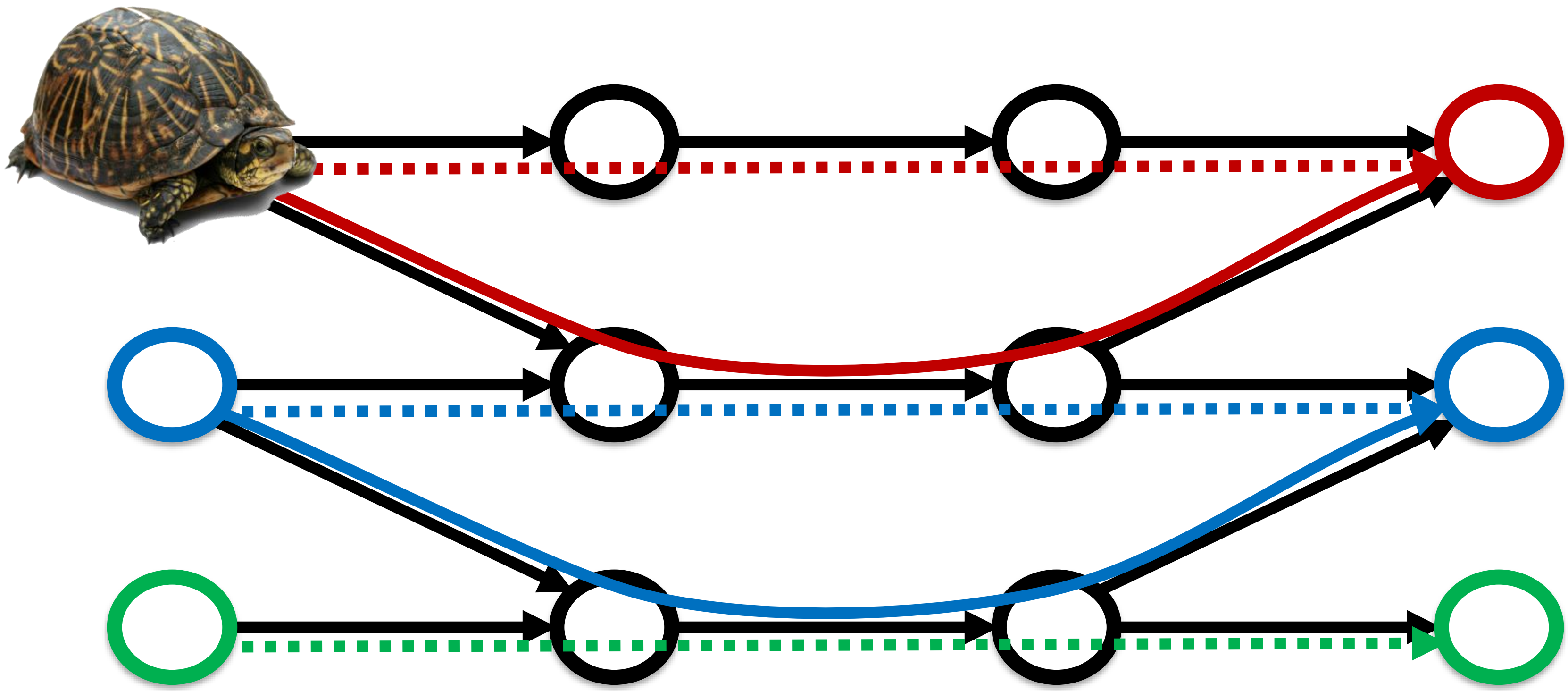
This would work



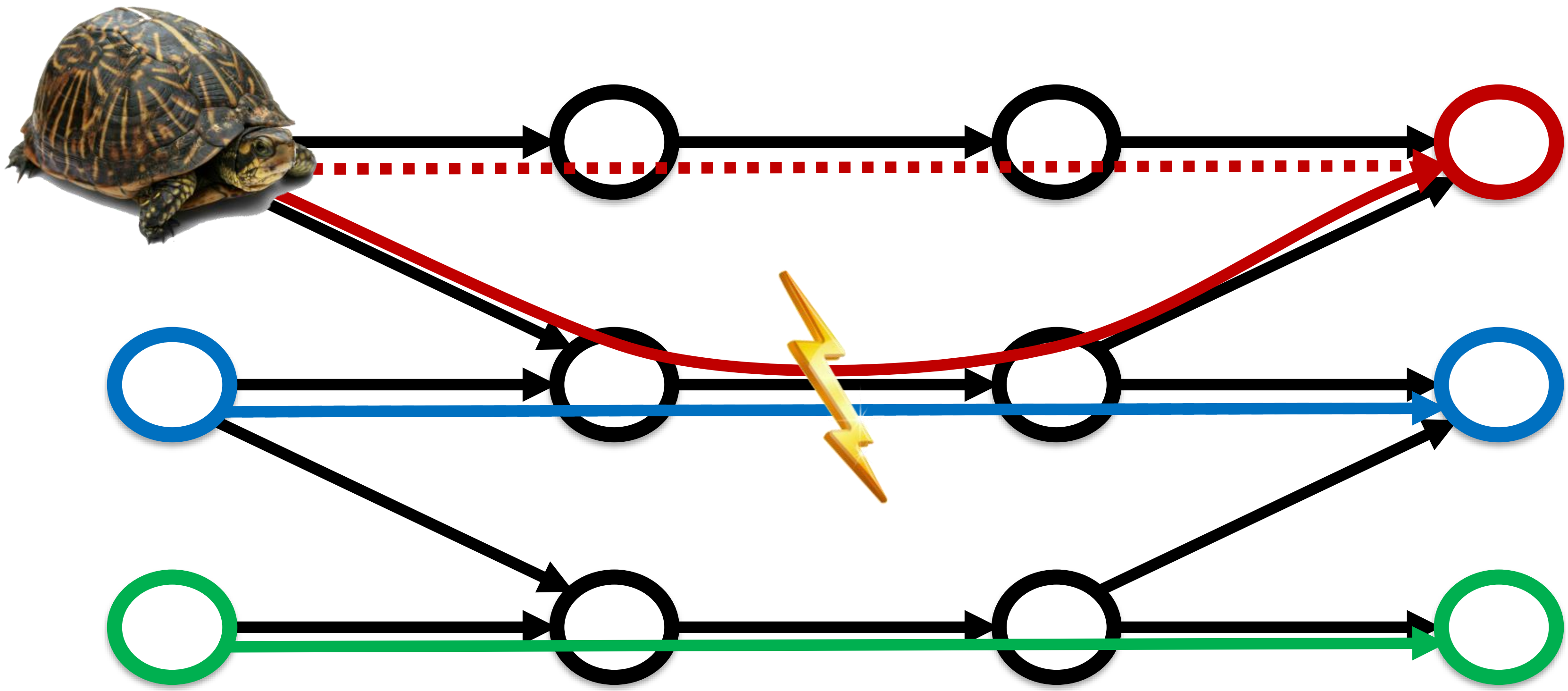
So lets go back



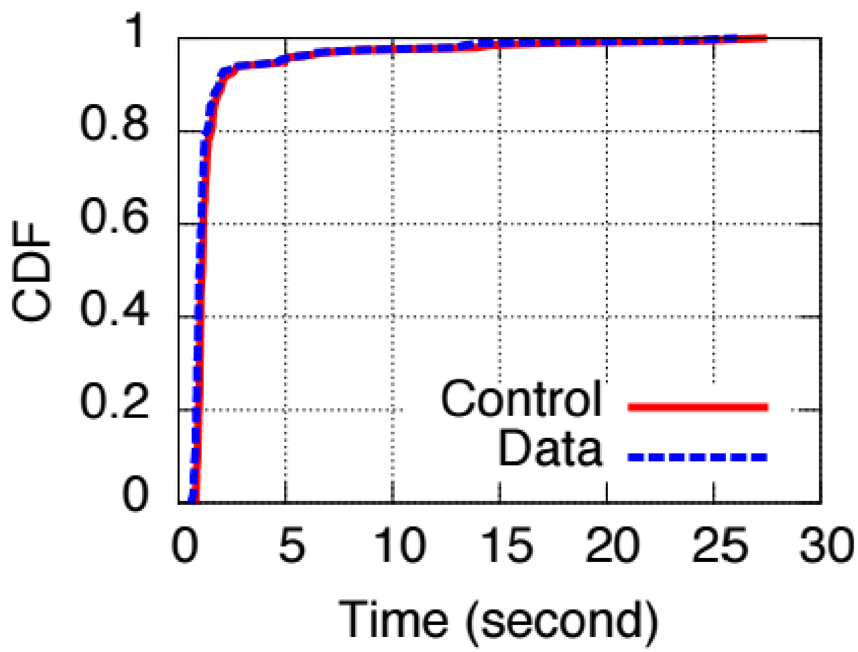
But Red is a bit Slow..



Congestion Again!

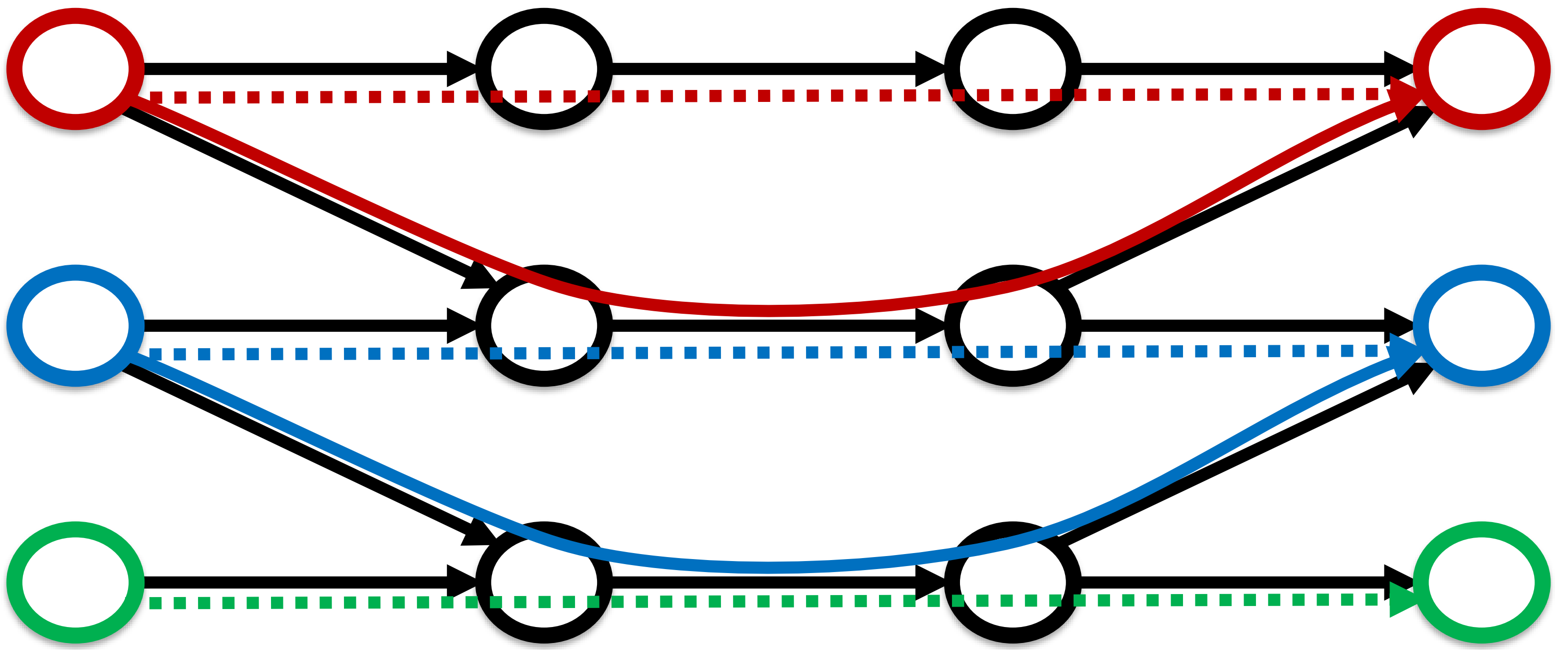


Appears in Practice

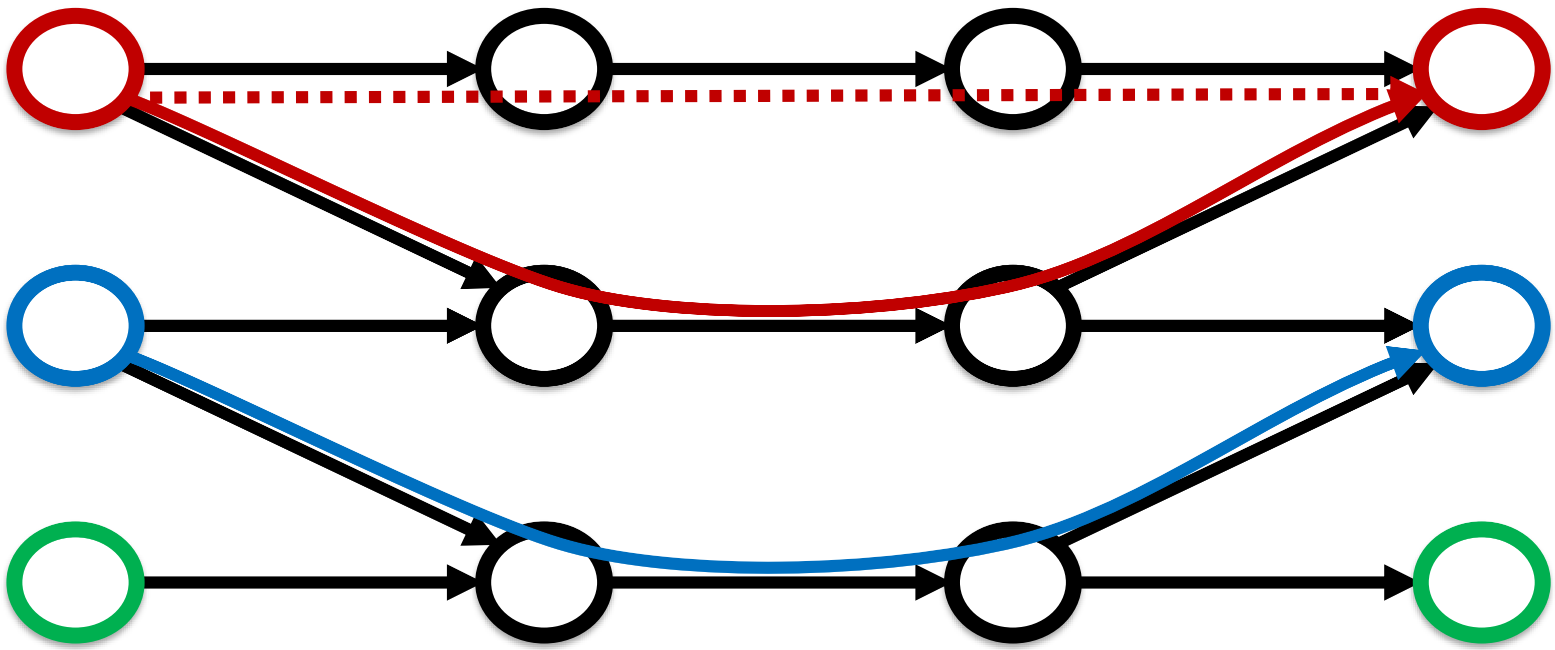


*"some switches can '**straggle**,' taking substantially **more time** than average (e.g., 10-**100x**) to apply an update"*
Jin et al., SIGCOMM 2014

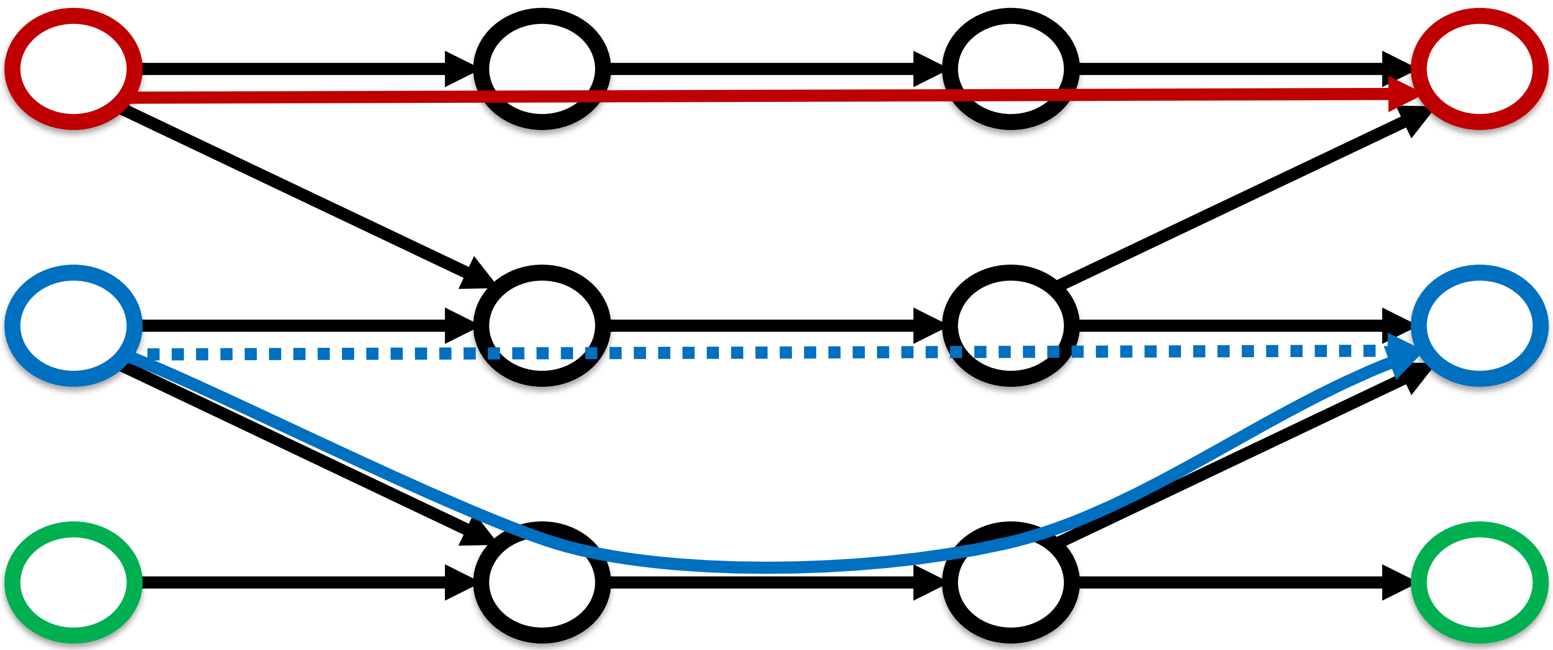
So lets go Back …



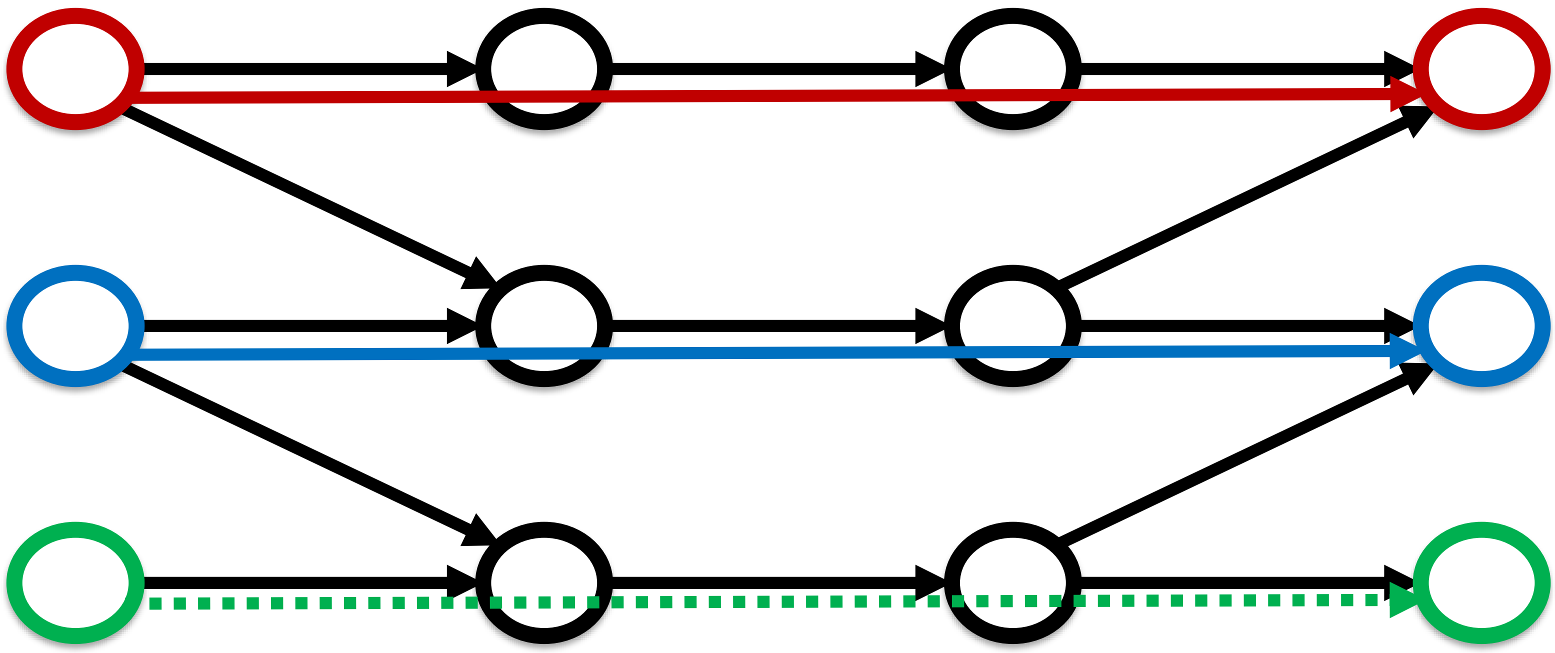
First, Red switches



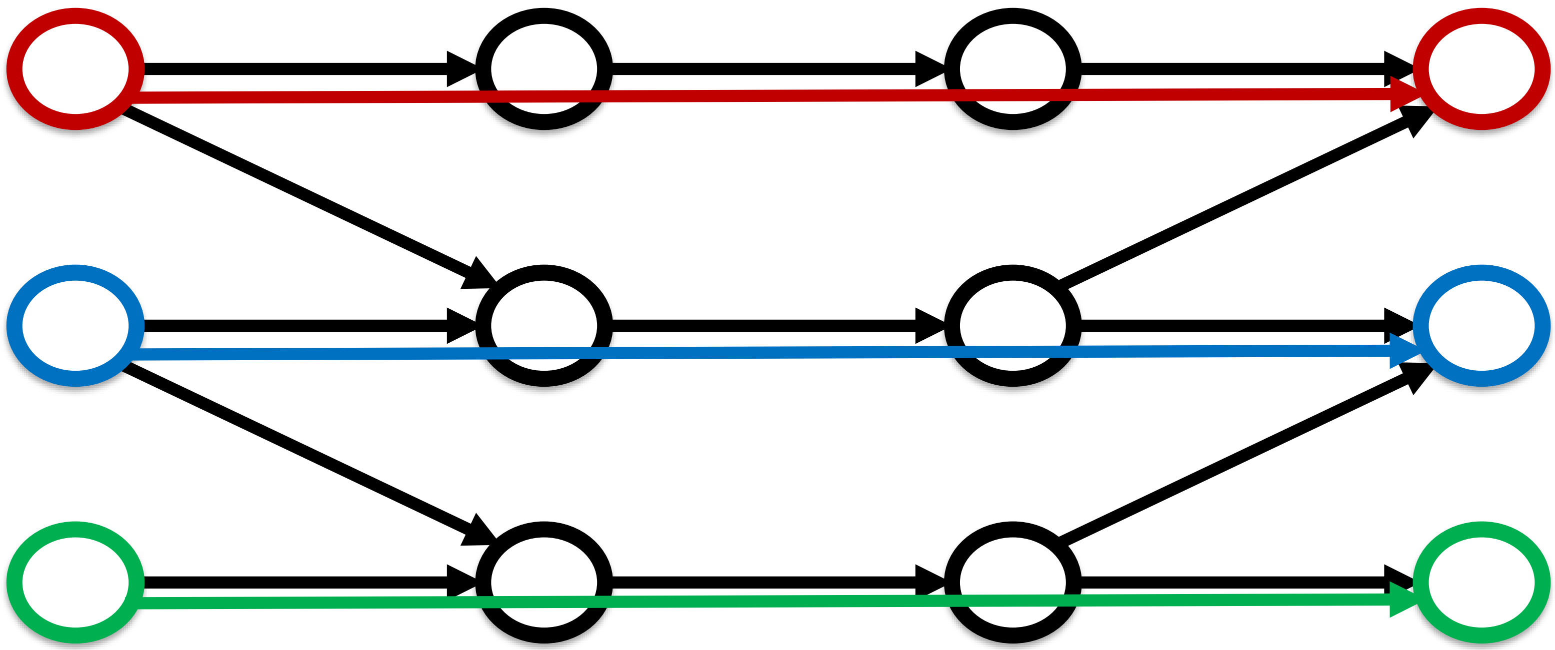
Then, Blue …



And then, Green …



Done



Consistent Migration of Flows

Introduced in *SWAN* (Hong et al., SIGCOMM 2013)

Idea: Flows can be on the **old** or **new** route

For all edges: $\sum_{\forall F} \max(\mathbf{old}, \mathbf{new}) \leq capacity$

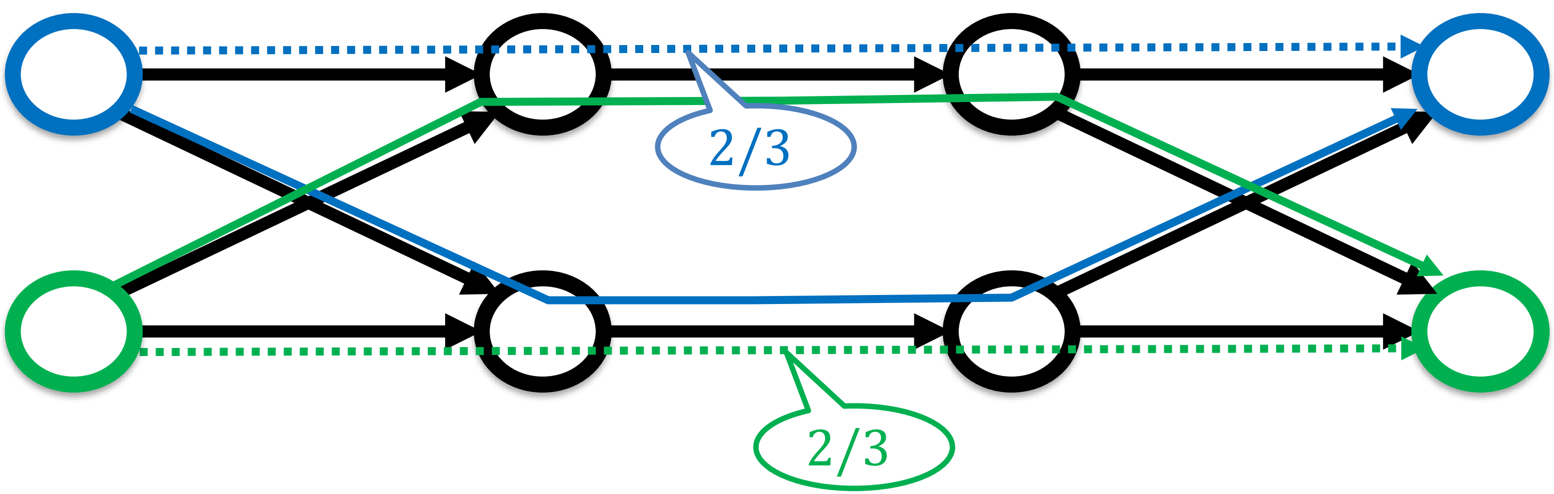
Consistent Migration of Flows

Introduced in *SWAN* (Hong et al., SIGCOMM 2013)

Idea: Flows can be on the **old** or **new** route

For all edges: $\sum_{\forall F} \max(\mathbf{old}, \mathbf{new}) \leq capacity$

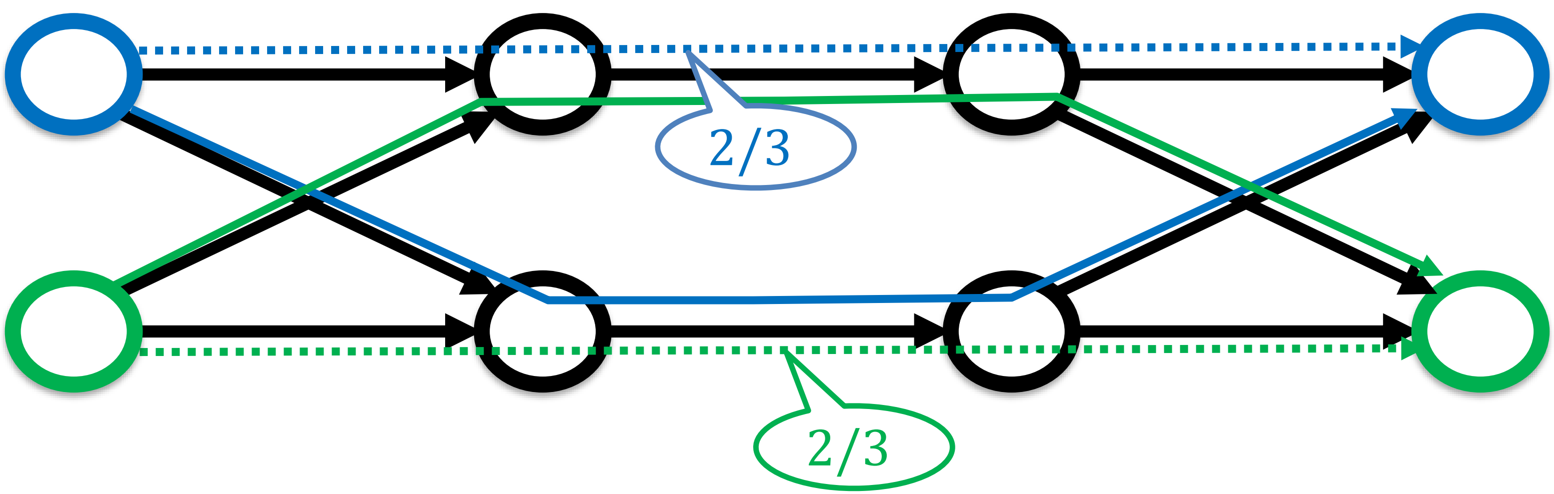
No ordering exists $(2/3 + 2/3 > 1)$



Consistent Migration of Flows

Approach of *SWAN*: use slack x (i.e., %)

- Here $x = 1/3$
- Move slack $x \Rrightarrow \lceil 1/x \rceil - 1$ staged partial moves

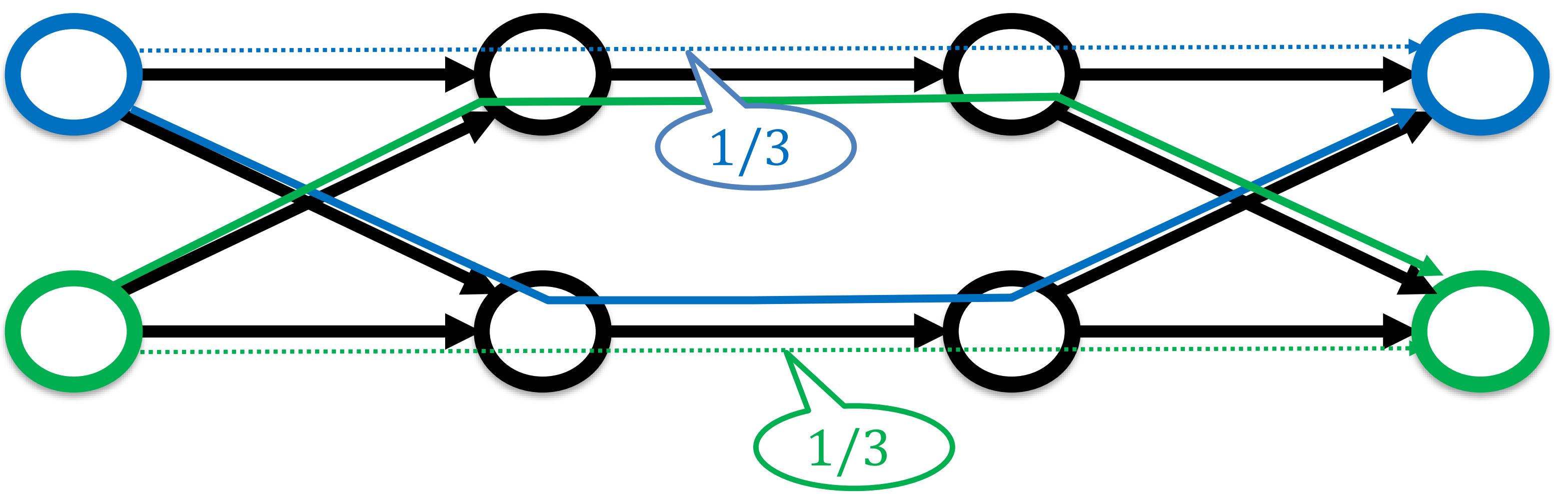


Consistent Migration of Flows

Approach of *SWAN*: use slack x (i.e., %)

- Here $x = 1/3$
- Move slack $x \Rrightarrow \lceil 1/x \rceil - 1$ staged partial moves

Update 1 of 2



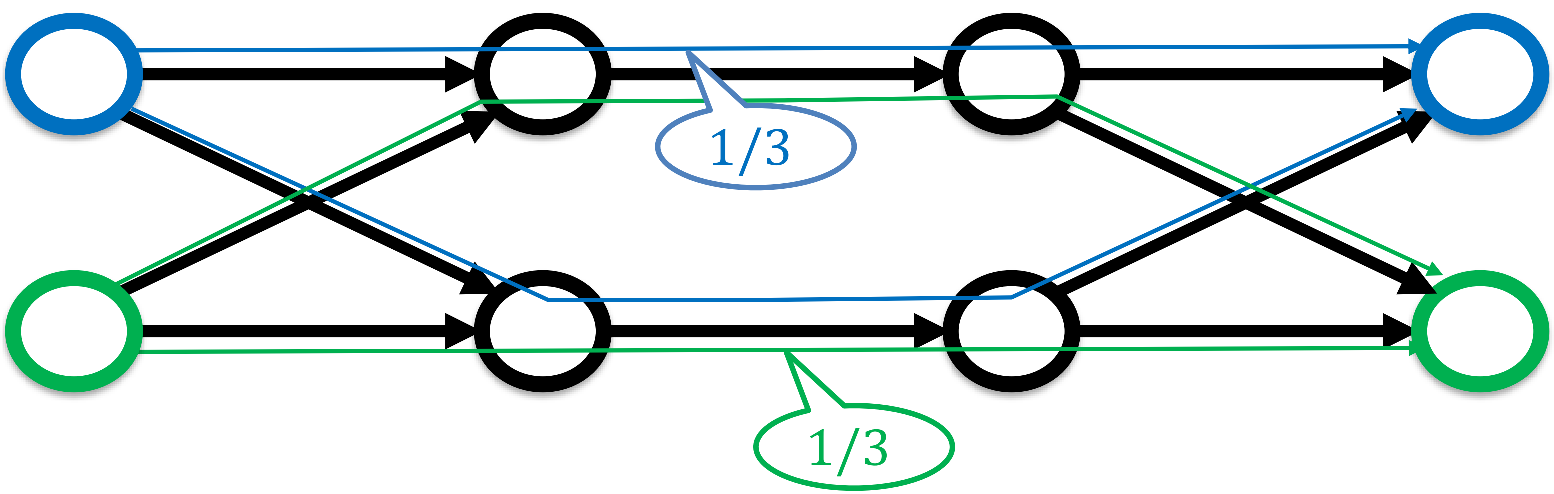
Consistent Migration of Flows

Approach of *SWAN*: use slack x (i.e., %)

Here $x = 1/3$

Move slack $x \Rrightarrow \lceil 1/x \rceil - 1$ staged partial moves

Update 1 of 2



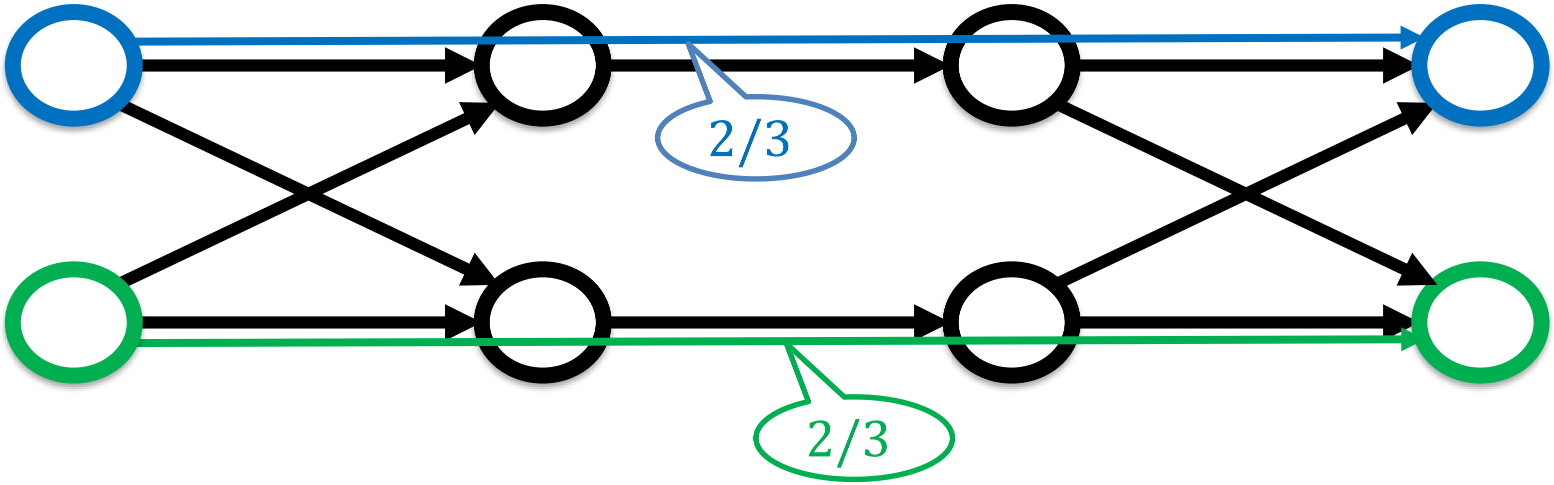
Consistent Migration of Flows

Approach of *SWAN*: use slack x (i.e., %)

Here $x = 1/3$

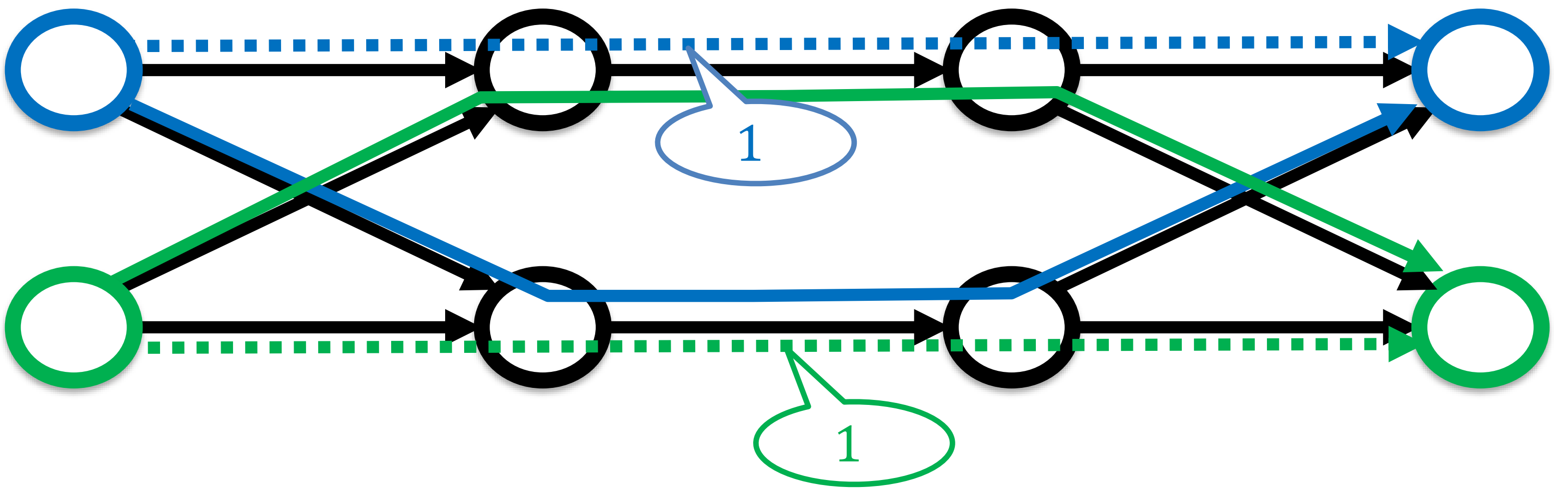
Move slack $x \Rrightarrow \lceil 1/x \rceil - 1$ staged partial moves

Update 2 of 2



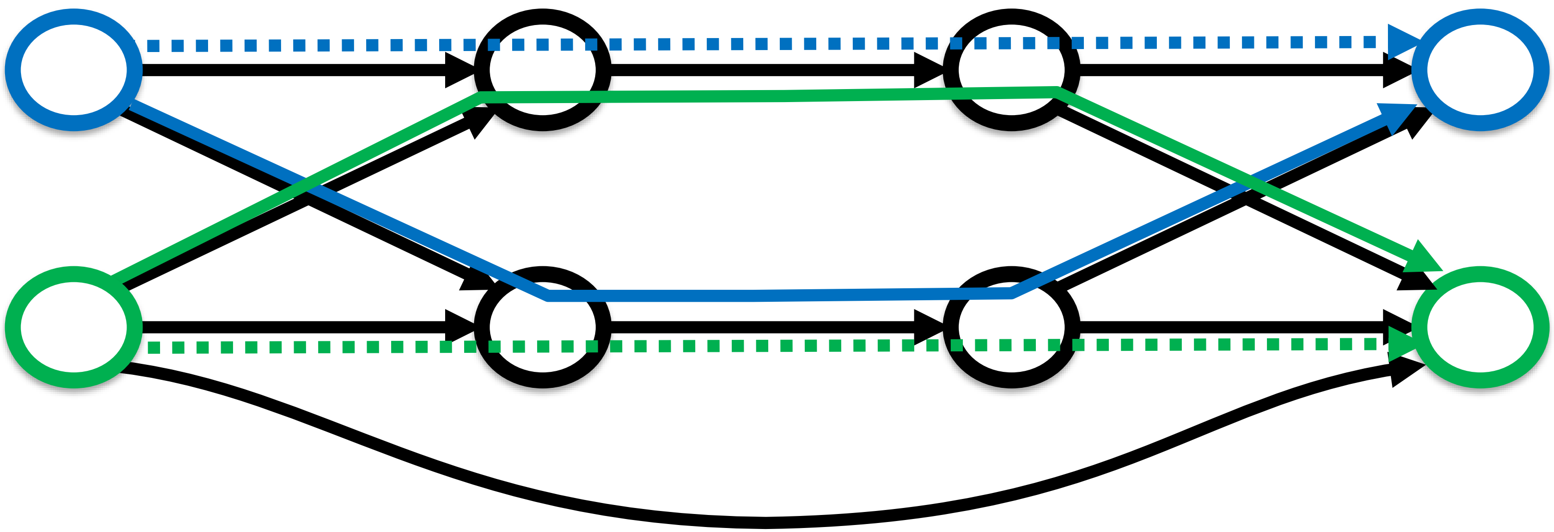
Consistent Migration of Flows

No slack on flow edges?



Consistent Migration of Flows

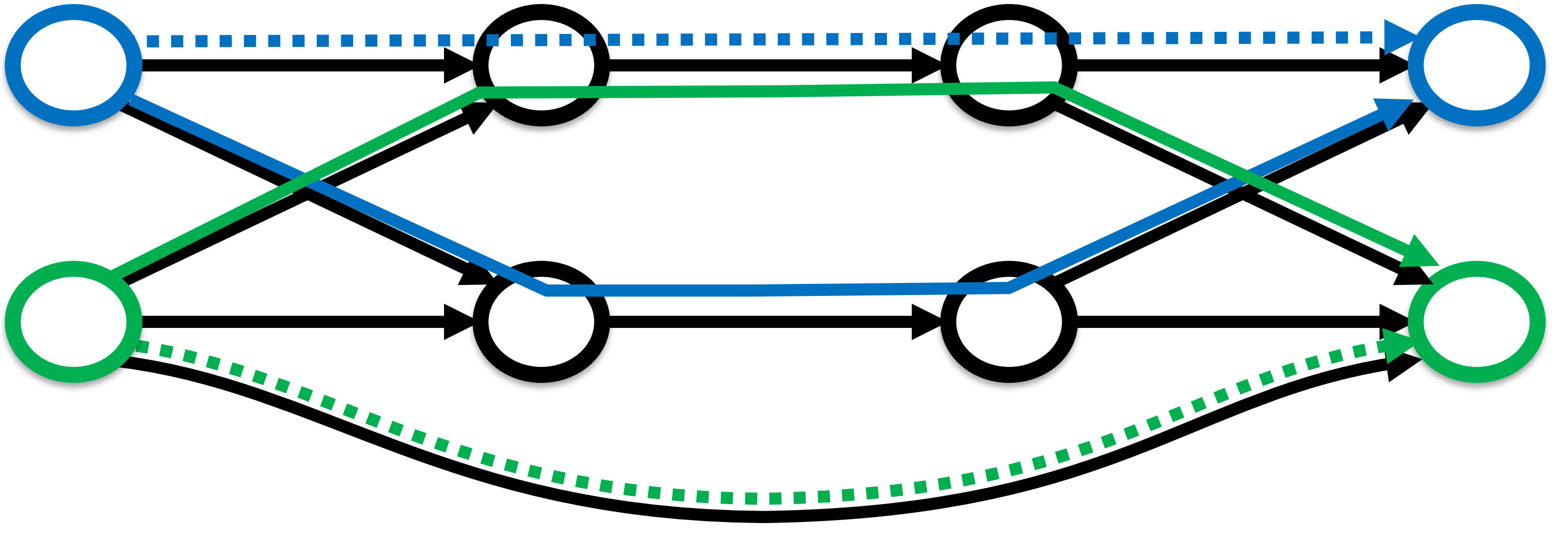
Alternate routes?



Consistent Migration of Flows

Think: variable swapping of b & g

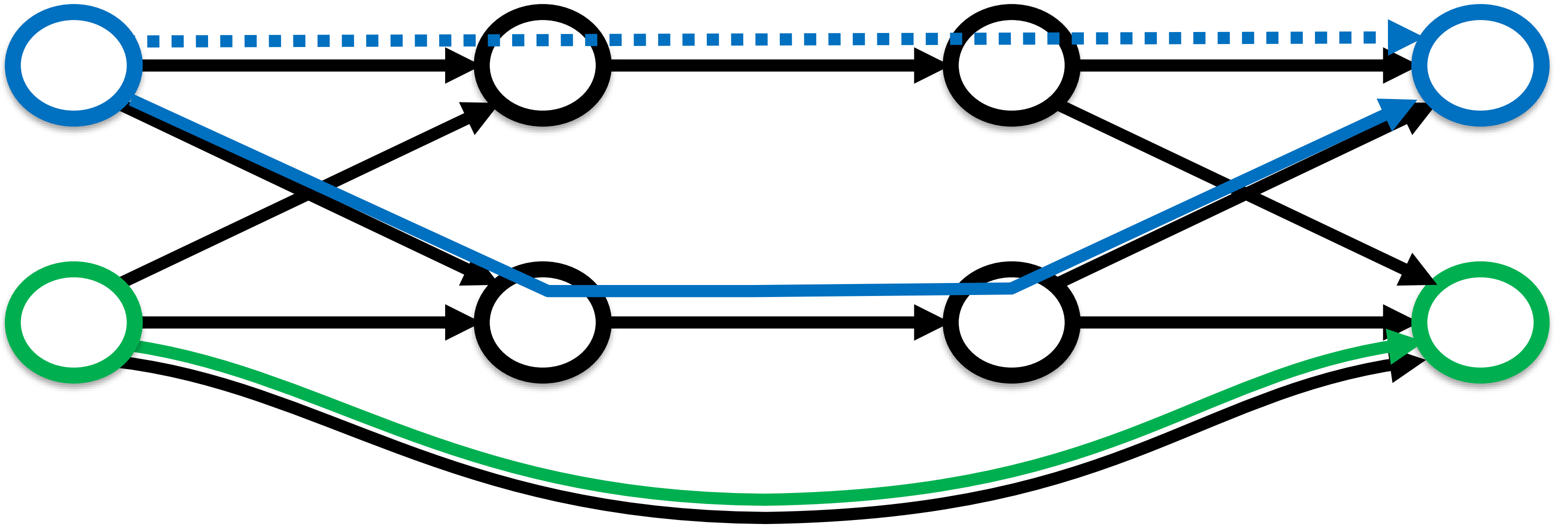
1. $x \coloneqq b$, 2. $\mathrm{b} \coloneqq g$, 3. $g \coloneqq x$



Consistent Migration of Flows

Think: variable swapping of b & g

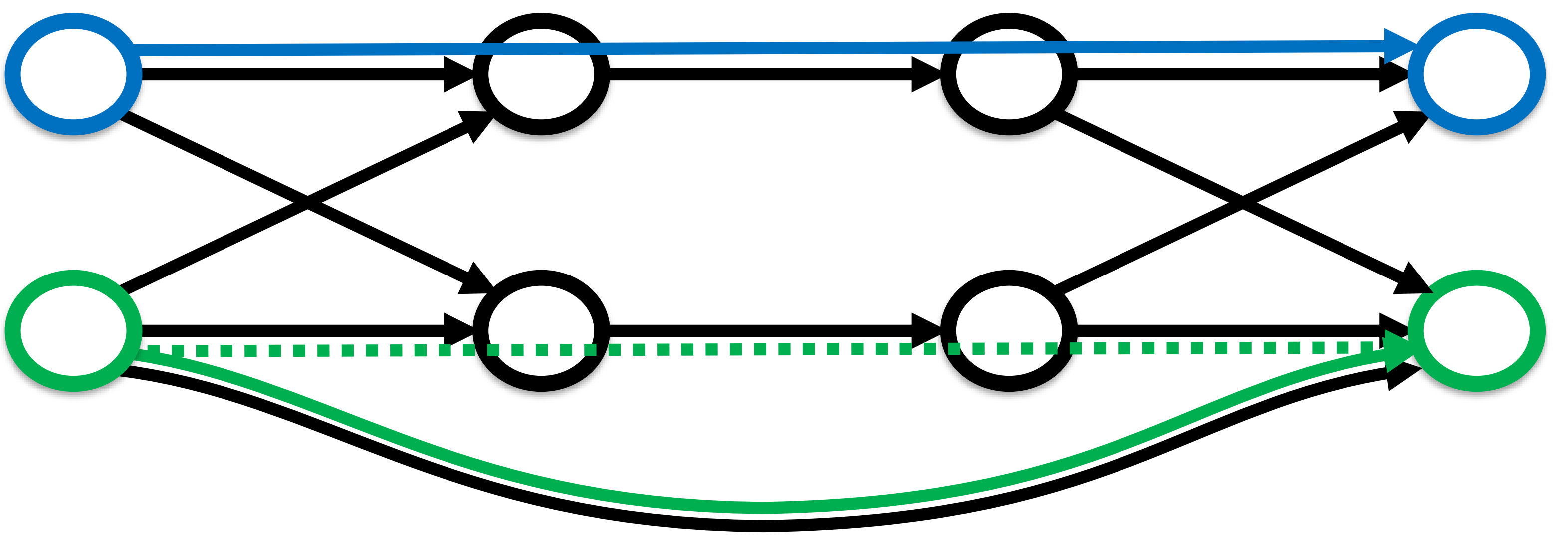
1. $x \coloneqq b$, 2. b $\coloneqq g$, 3. $g \coloneqq x$



Consistent Migration of Flows

Think: variable swapping of b & g

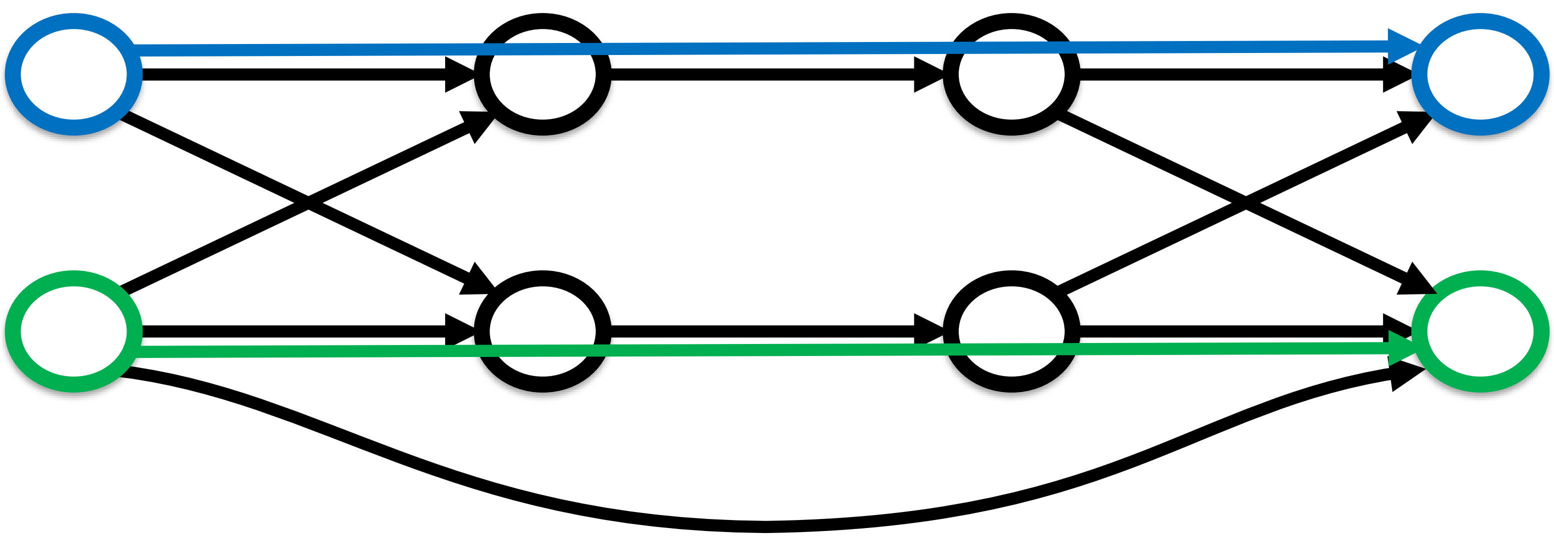
1. $x \coloneqq b$, 2. $\mathrm{b} \coloneqq g$, 3. $g \coloneqq x$



Consistent Migration of Flows

SWAN: LP-approach with binary search

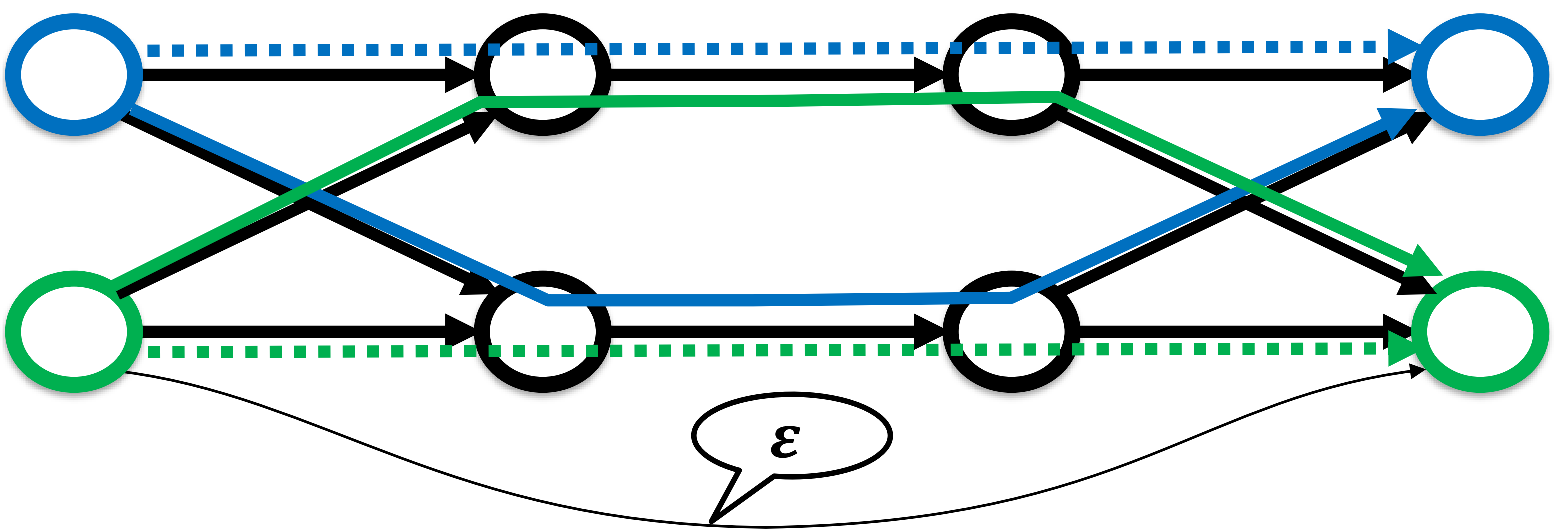
1 update? 2 updates? 4 updates? …



Consistent Migration of Flows

SWAN: LP-approach with binary search

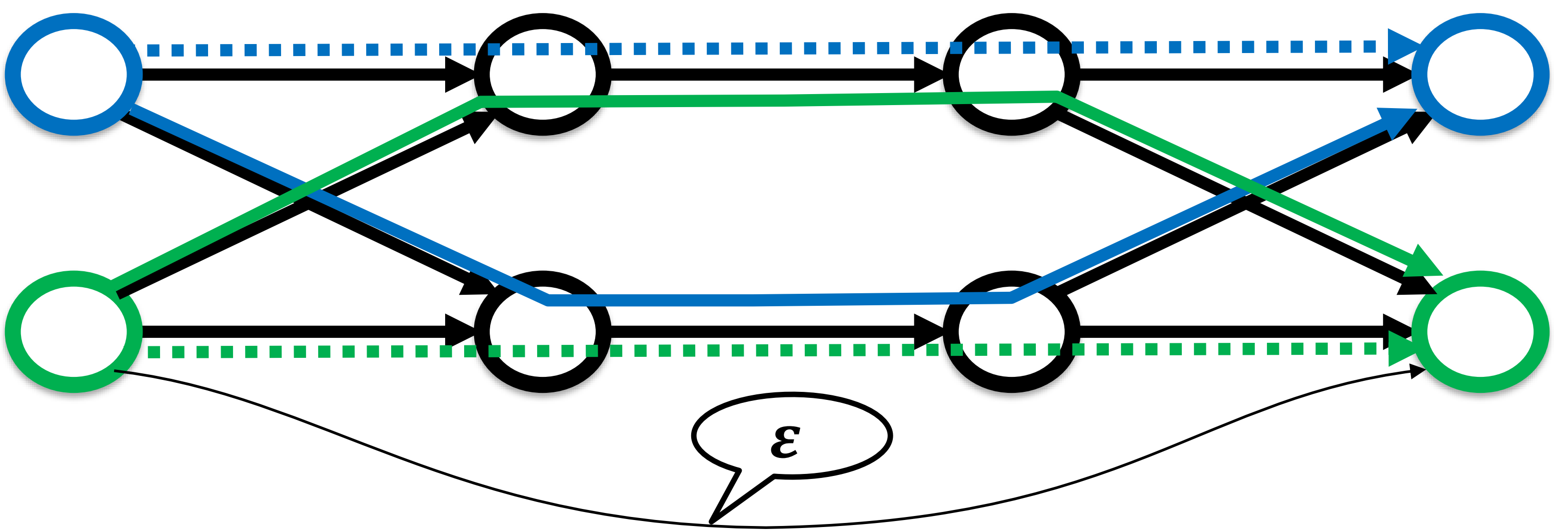
1 update? 2 updates? 4 updates? …



Consistent Migration of Flows

SWAN: LP-approach with binary search

$\Theta(1/\varepsilon)$ updates ☹



Consistent Migration of Flows

Open problem: Can we decide in (**polynomial**) time?

To Slack or not to Slack?

Slack of x on all flow edges?

$\lceil 1/x \rceil - 1$ updates

To Slack or not to Slack?

What if not?

Try to create slack

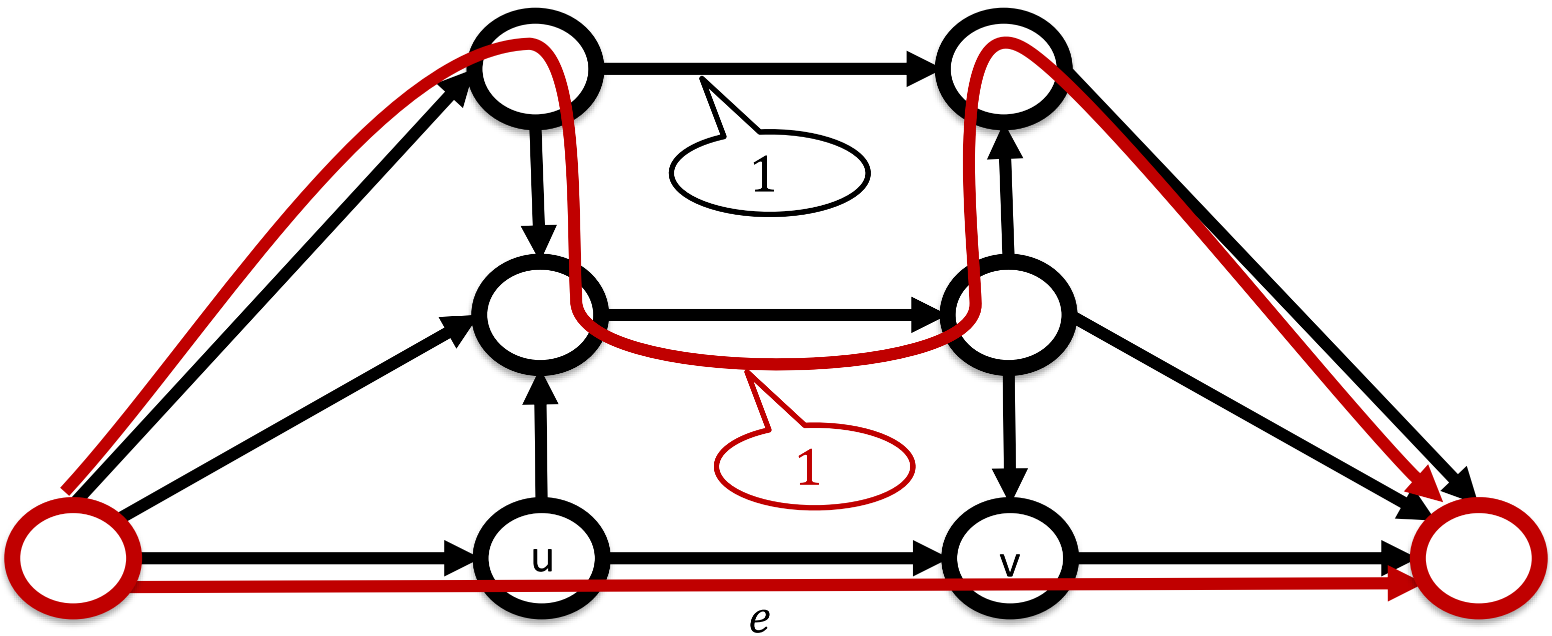
To Slack or not to Slack?

Combinatorial approach

Augmenting paths

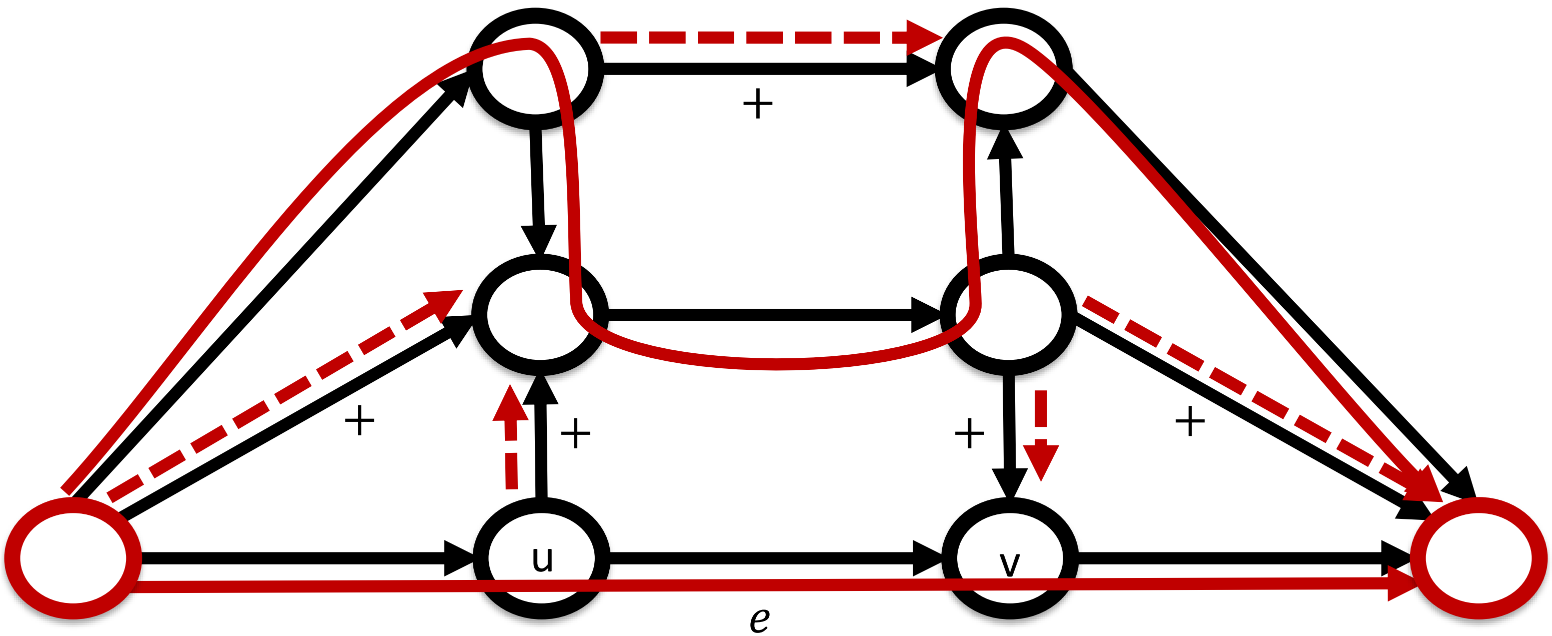
Combinatorial Approach

Move single commodities at a time



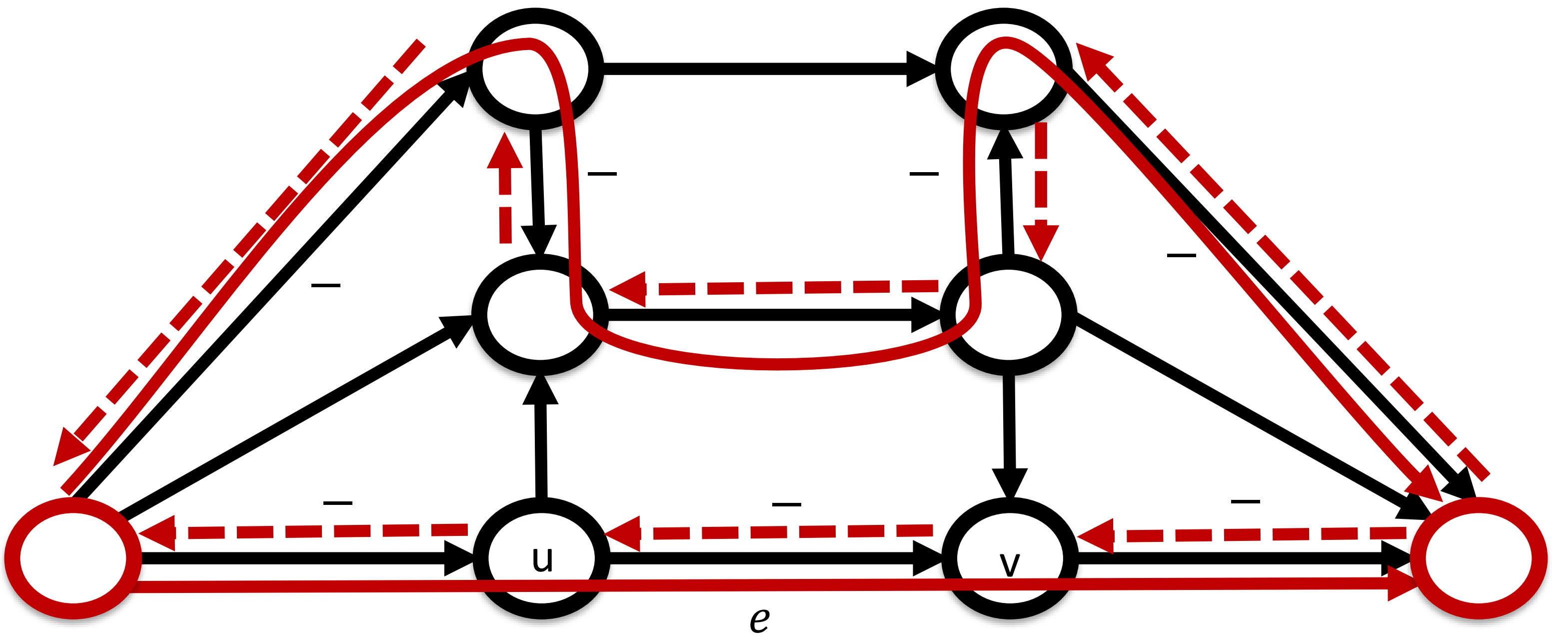
Combinatorial Approach

Where to increase flow?



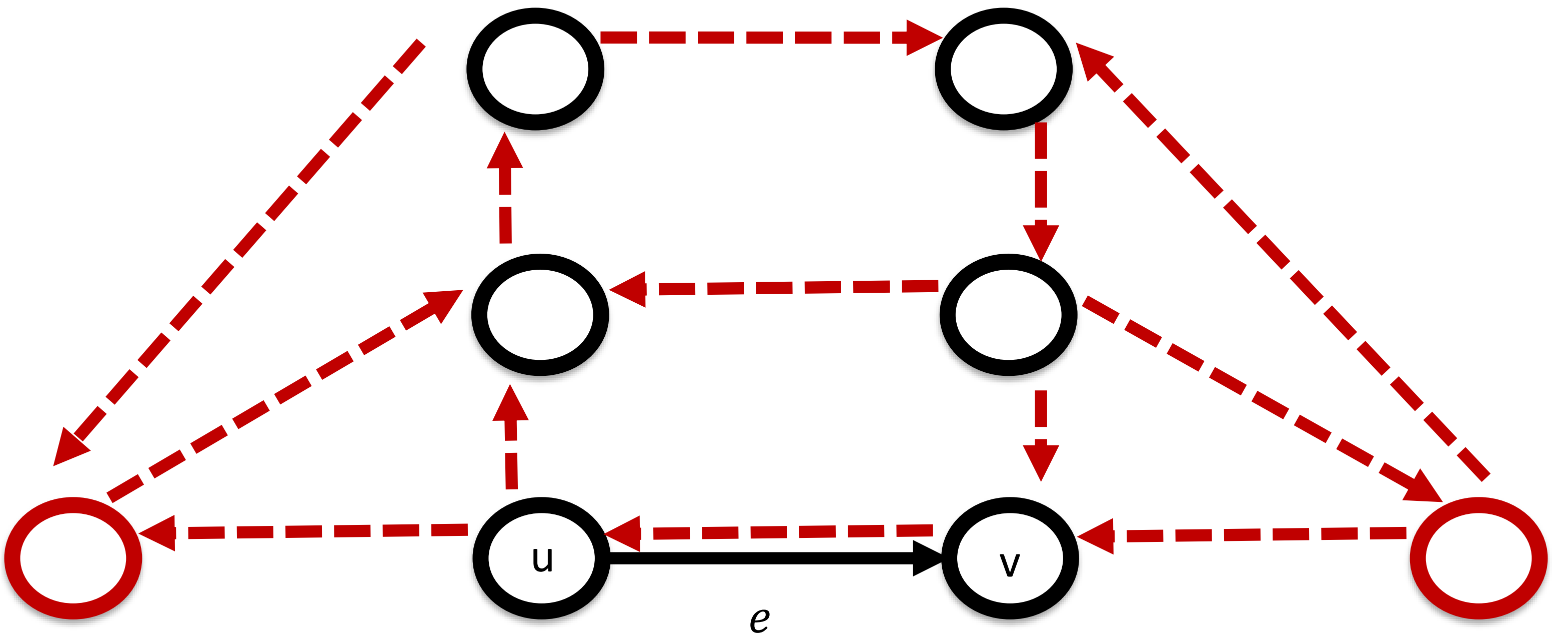
Combinatorial Approach

Where to push back flow?



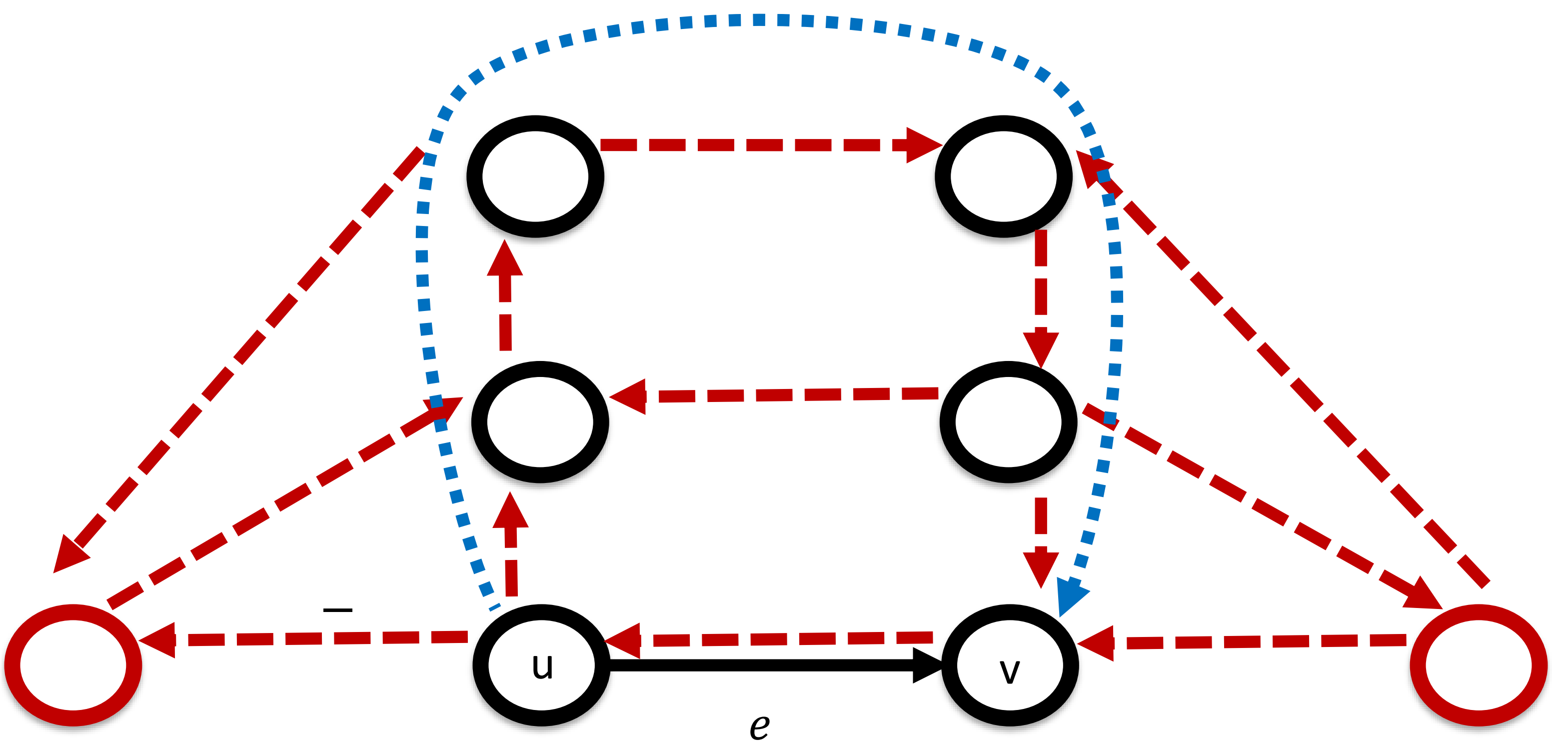
Combinatorial Approach

Resulting residual network



Combinatorial Approach

We found an augmenting path $\Rightarrow$ create slack on e



High-level Algorithm Idea

No slack on flow edges? Find augmenting paths

- On both initial and desired state
- Success? Use *SWAN* method to migrate

Can’t create slack on some flow edge?

- Consistent migration impossible
- By contradiction (else augmenting paths would create slack)

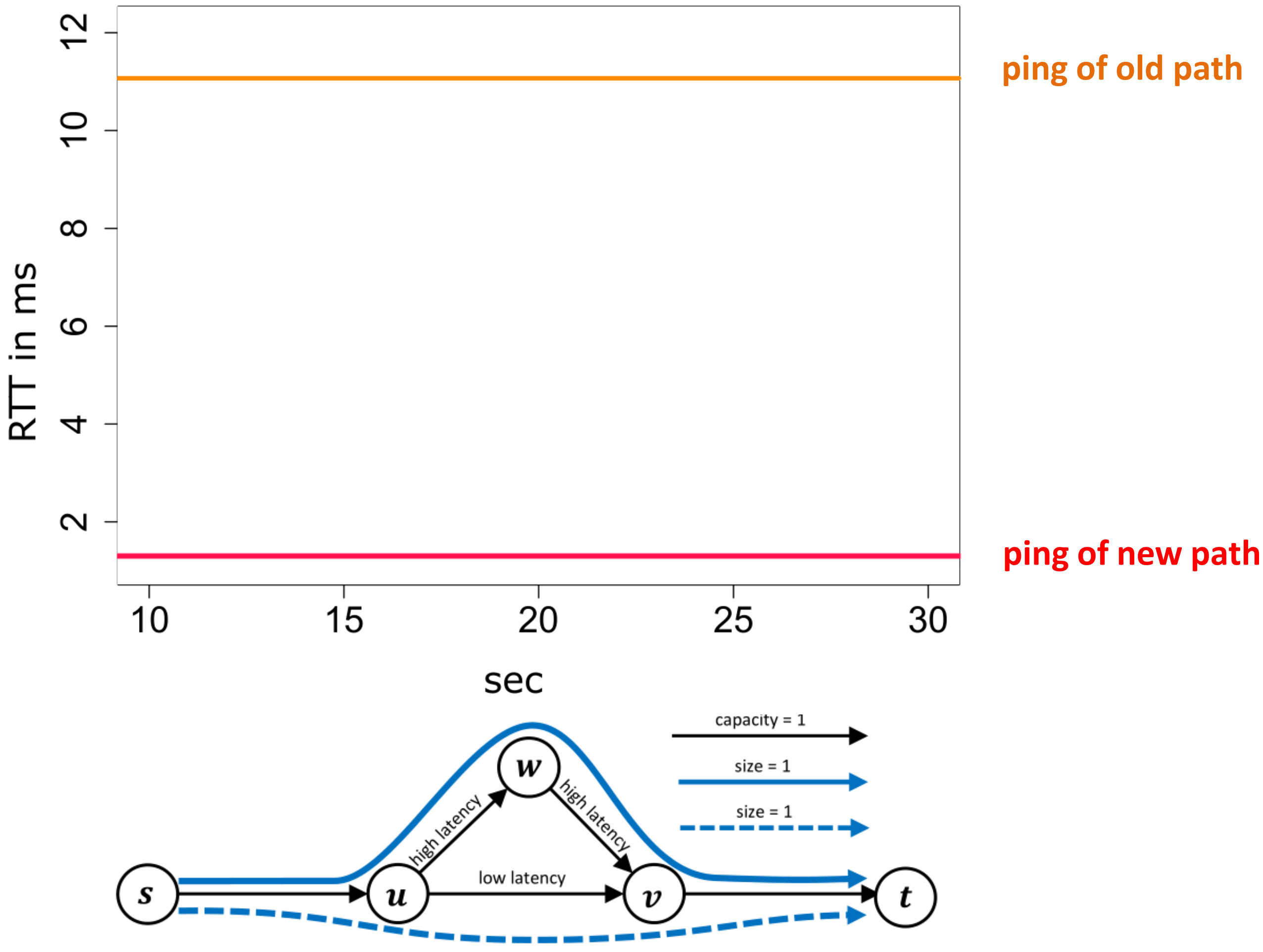
Runtime: $O(Fm^3)$

- (F being #commodities, m being #edges)

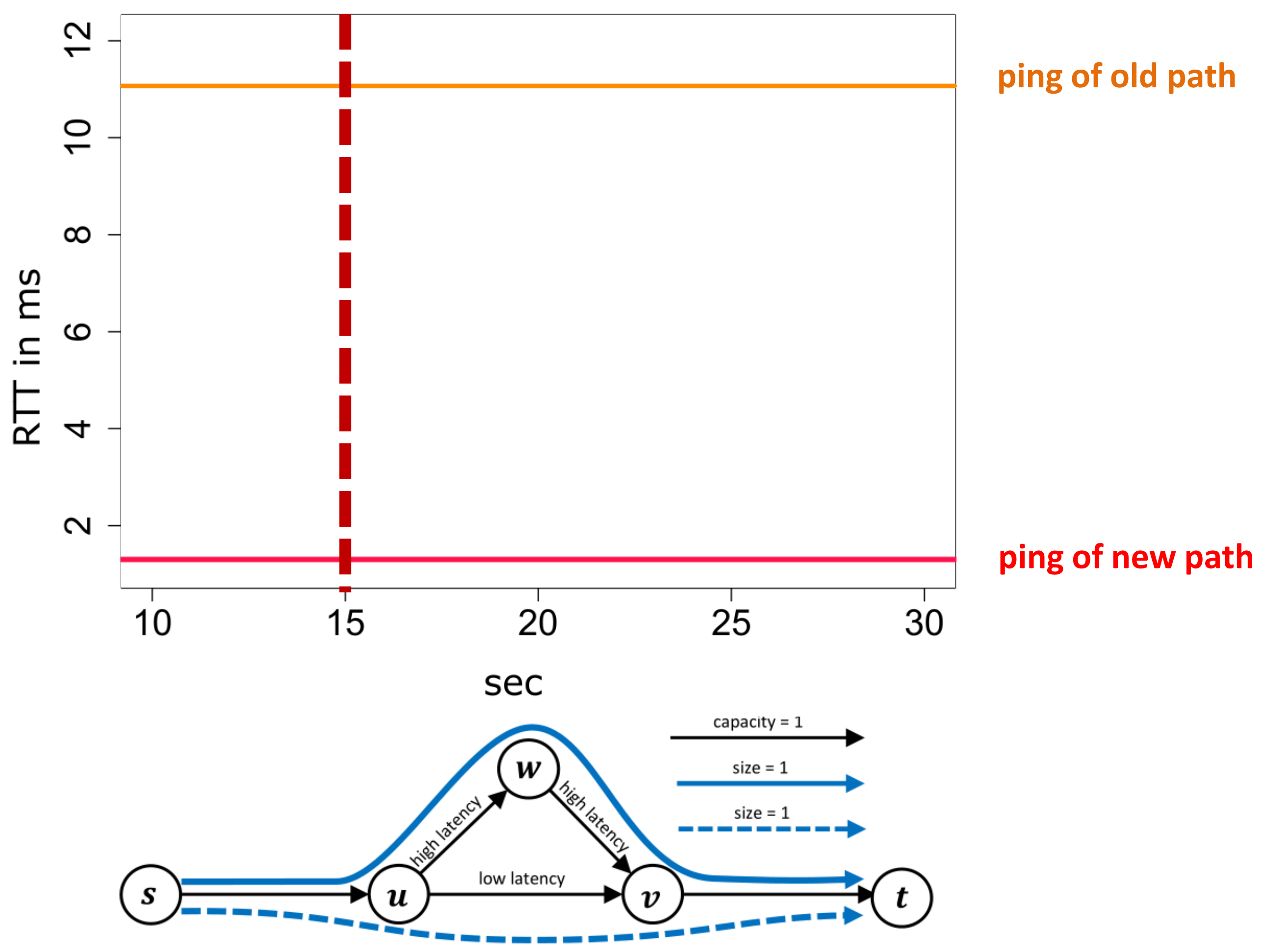
Are we done?

Consistent Migration = Lossless Migration?

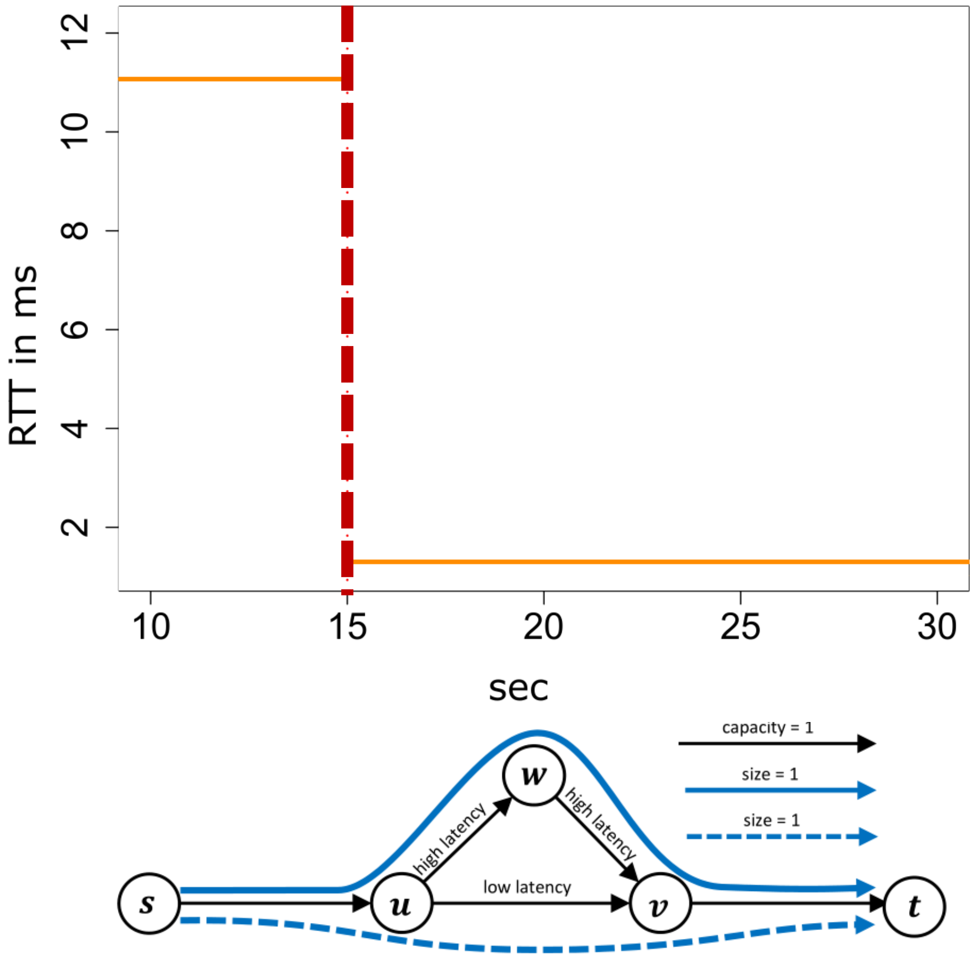
Moving Flows with High Latency



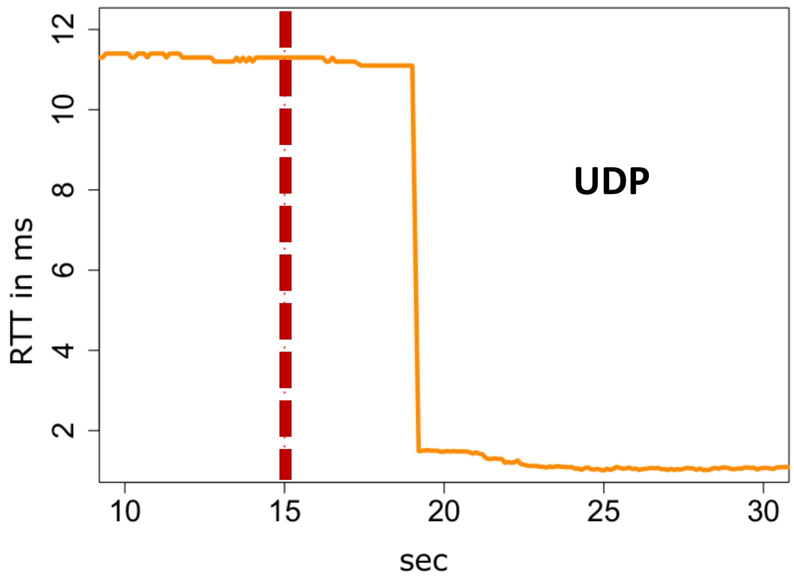
Moving Flows with High Latency

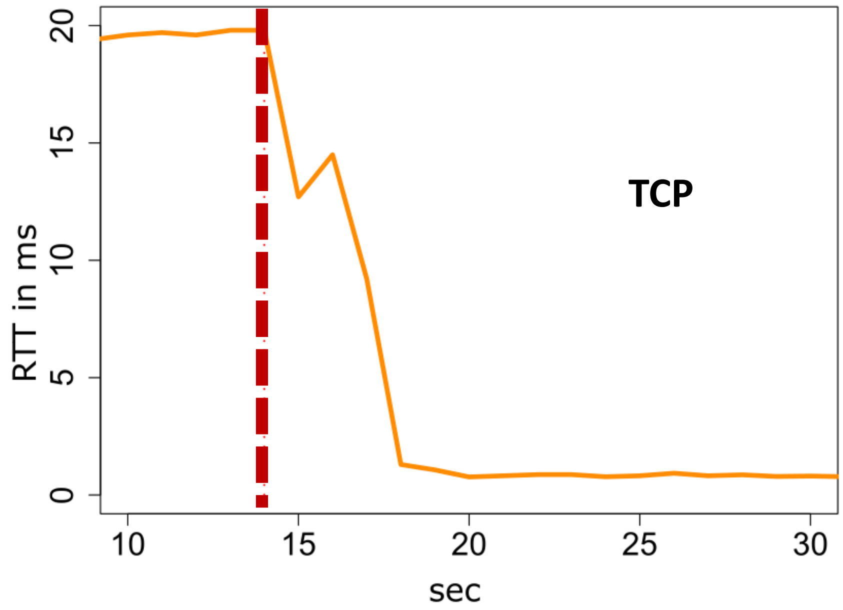


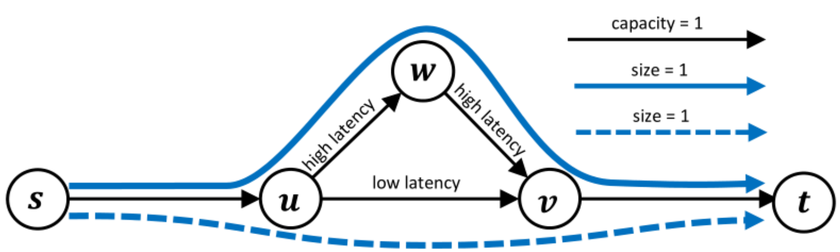
Moving Flows with High Latency



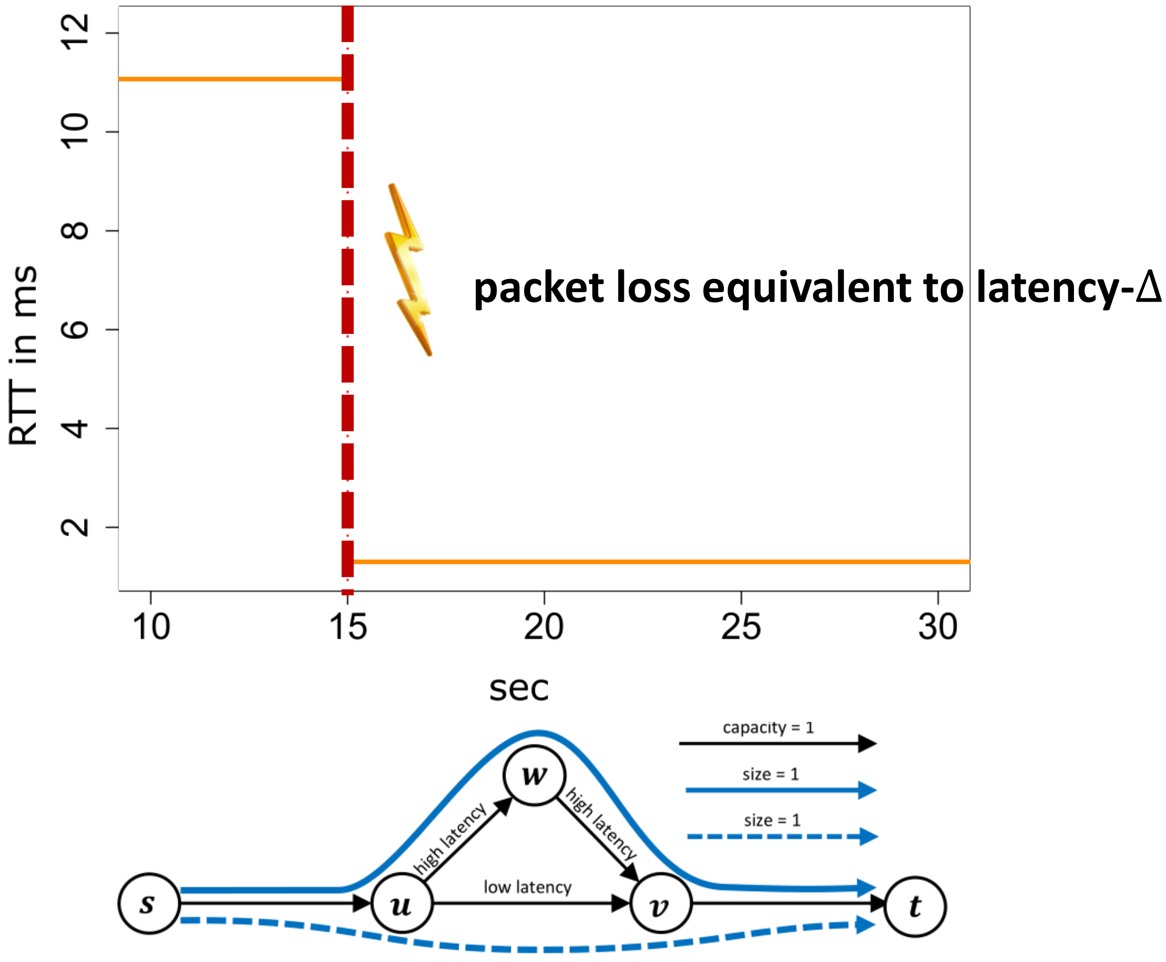
Moving Flows with High Latency







Moving Flows with High Latency



Overview and Outlook

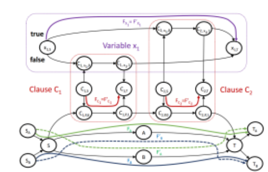
Consistent migration of flows

Decidable in **polynomial** time

Lossless migration

Fixed delays: **NP-hard**, **Arbitrary** delays: **Polynomially** decidable

Unsplittable/Integer flow migration: **NP-hard**



Migrate to new **demands**

Single-dest: **Polynomial** #updates, Multi-Dest: **Ongoing work**

Don’t disturb my Flows: Consistent Migration of Flows in SDNs

<u>Klaus-Tycho Förster</u>

Joint work with Sebastian Brandt, Laurent Vanbever, and Roger Wattenhofer