On Consistent Migration of Flows in SDNs

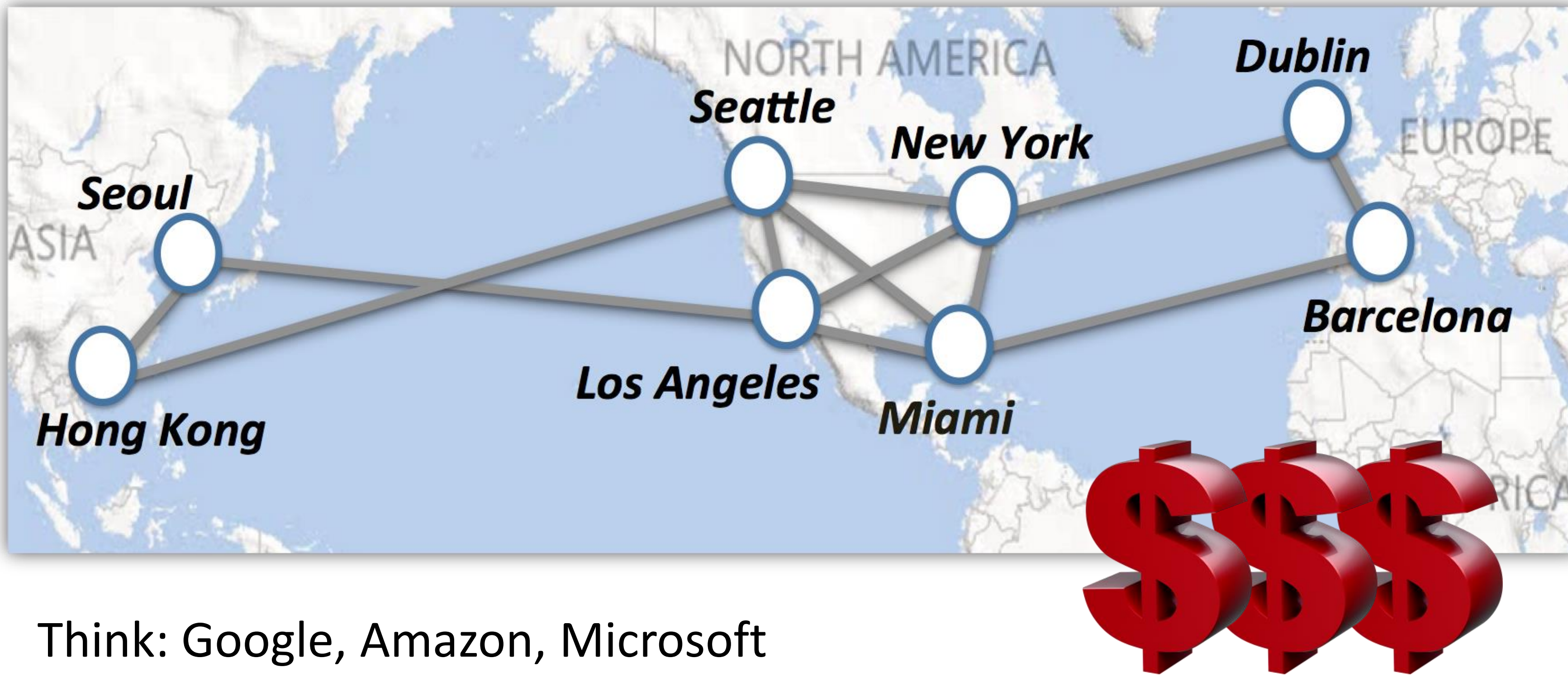
*Sebastian Brandt, **Klaus-Tycho Förster**, Roger Wattenhofer*

April 12, 2016 @ INFOCOM 2016 – San Francisco

The Internet

Central Control?

Own WAN = Expensive



Think: Google, Amazon, Microsoft

Software **D**efined **N**etworking (**SDN**)

Network Updates

old network
rules

network updates

new network
rules

Network Updates

old network
rules

network updates

new network
rules

State of the art: (Partial) moves of flows using linear programming (LPs), e.g.,
SWAN [Hong et al., SIGCOMM 2013], *zUPDATE* [Liu et al., SIGCOMM 2013]
Dionysus [Jin et al., SIGCOMM 2014]

Network Updates

old network
rules

network updates

new network
rules

Open problems:

When are network updates in a consistent manner *possible*?

How can we decide *fast*?

Network Updates

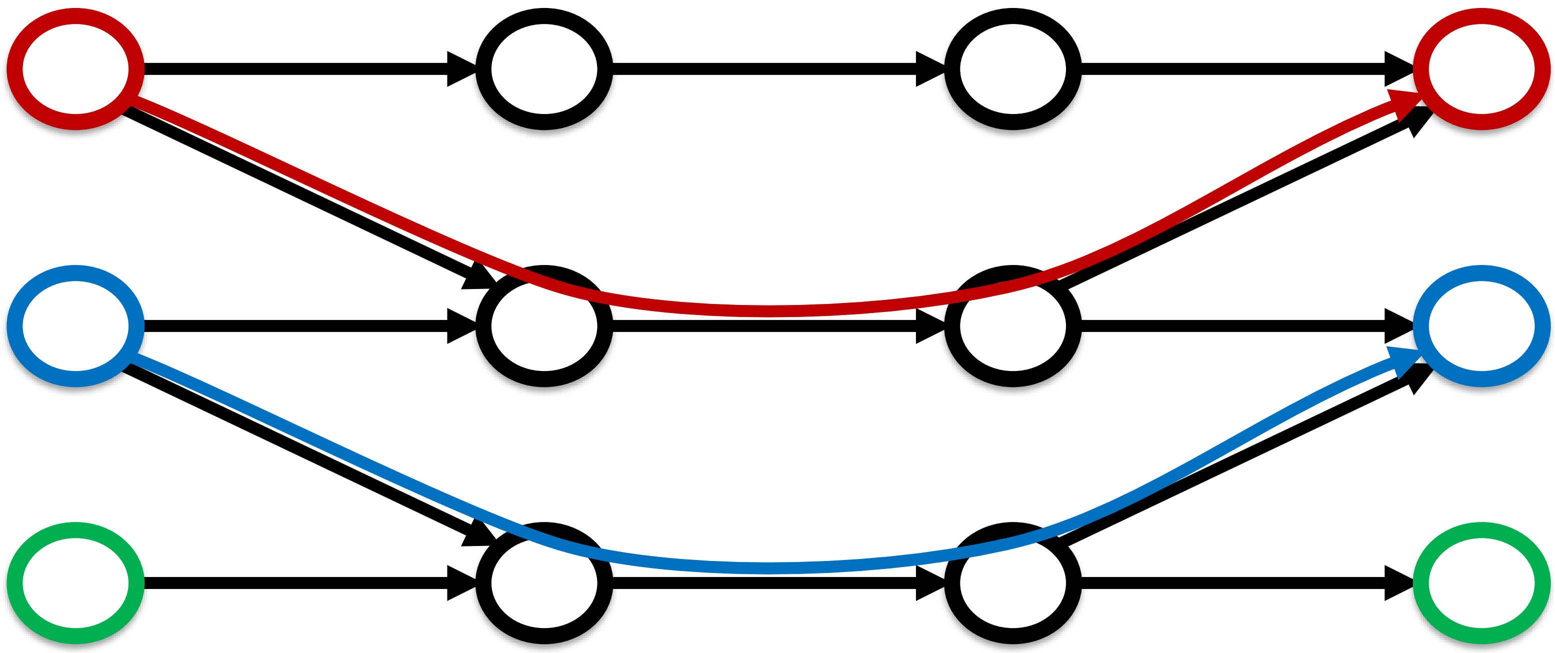
old network
rules

network updates

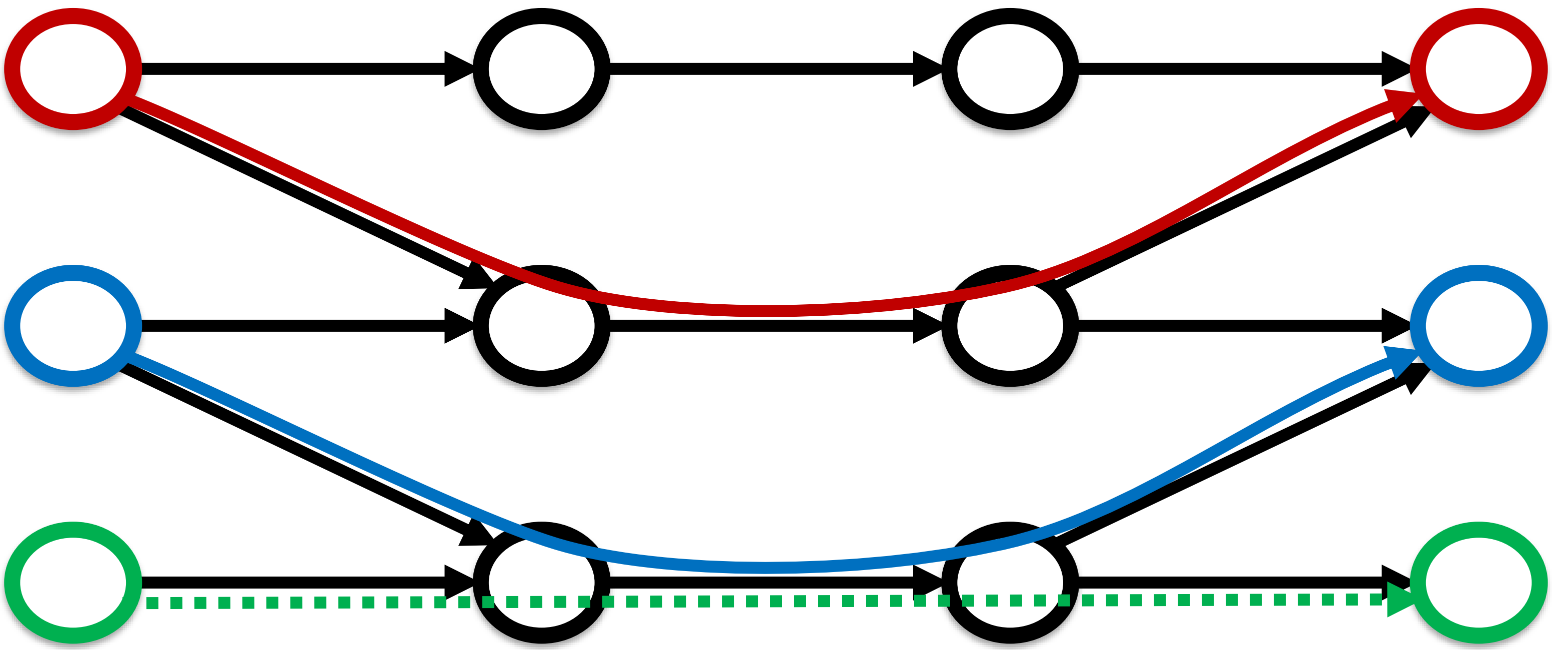
new network
rules

This paper: Addresses the case of splittable multi-commodity flows

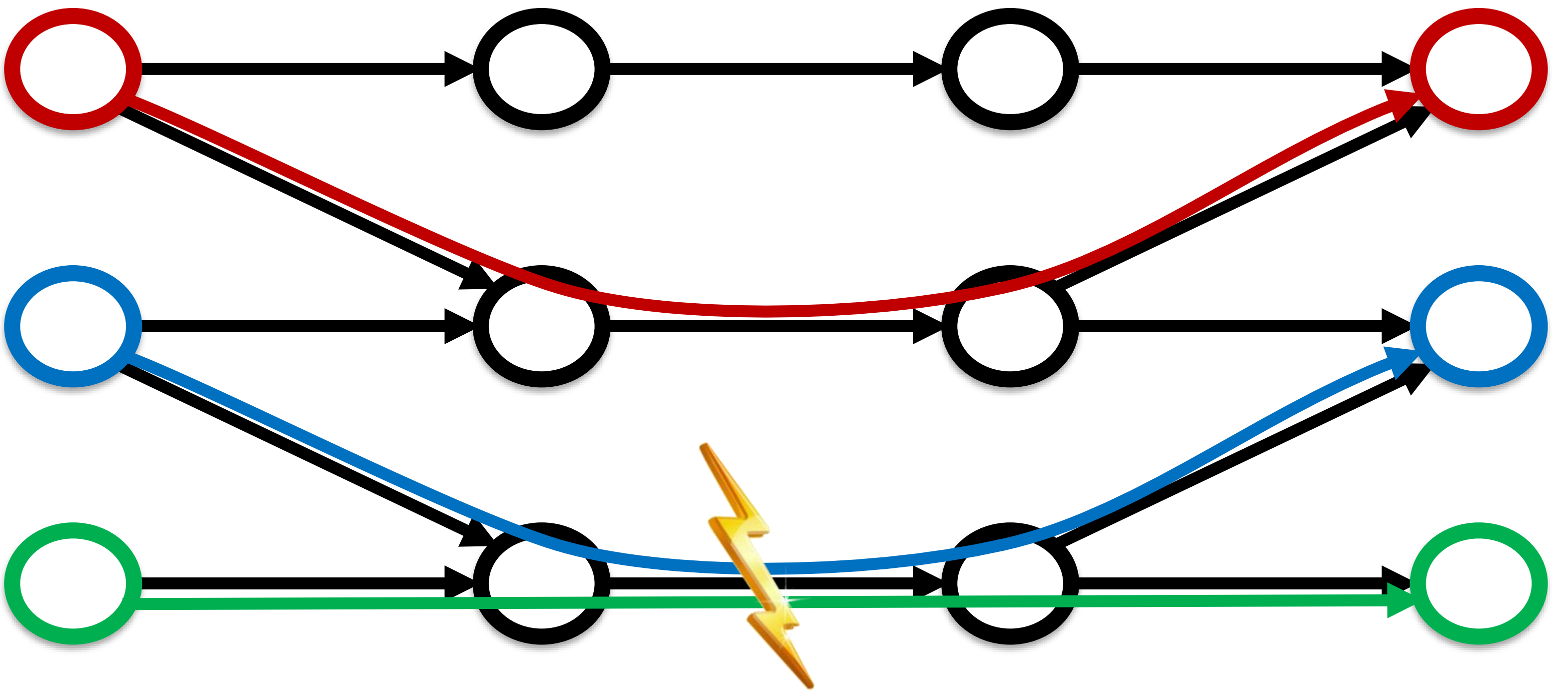
A Small Sample Network



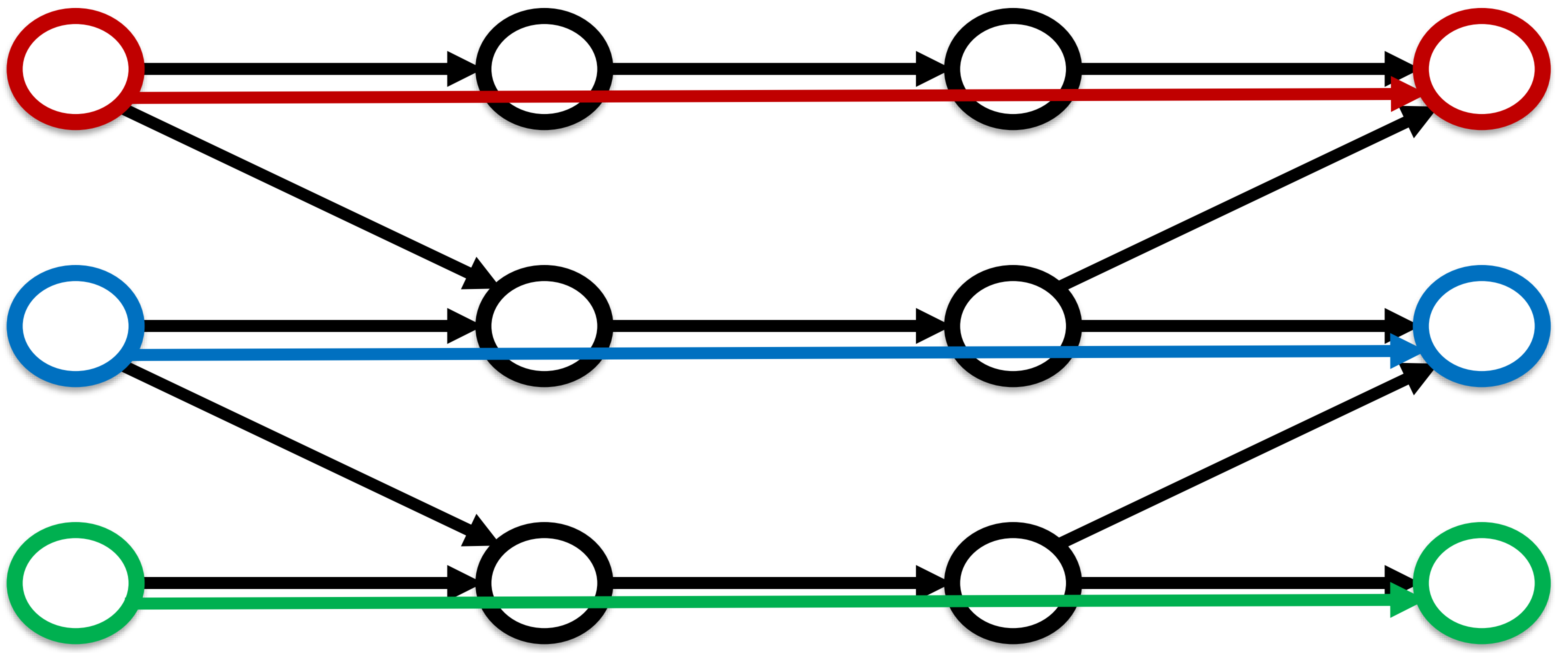
Green wants to send as well



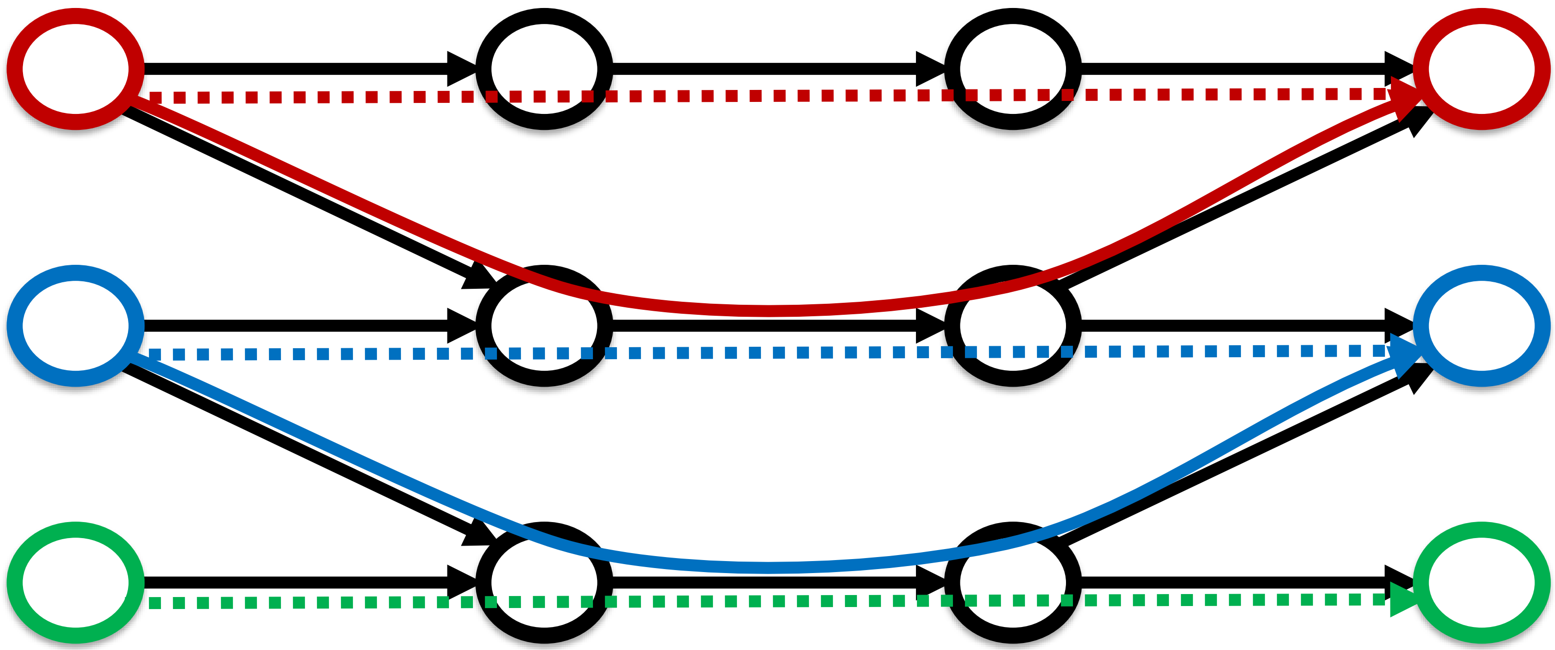
Congestion!



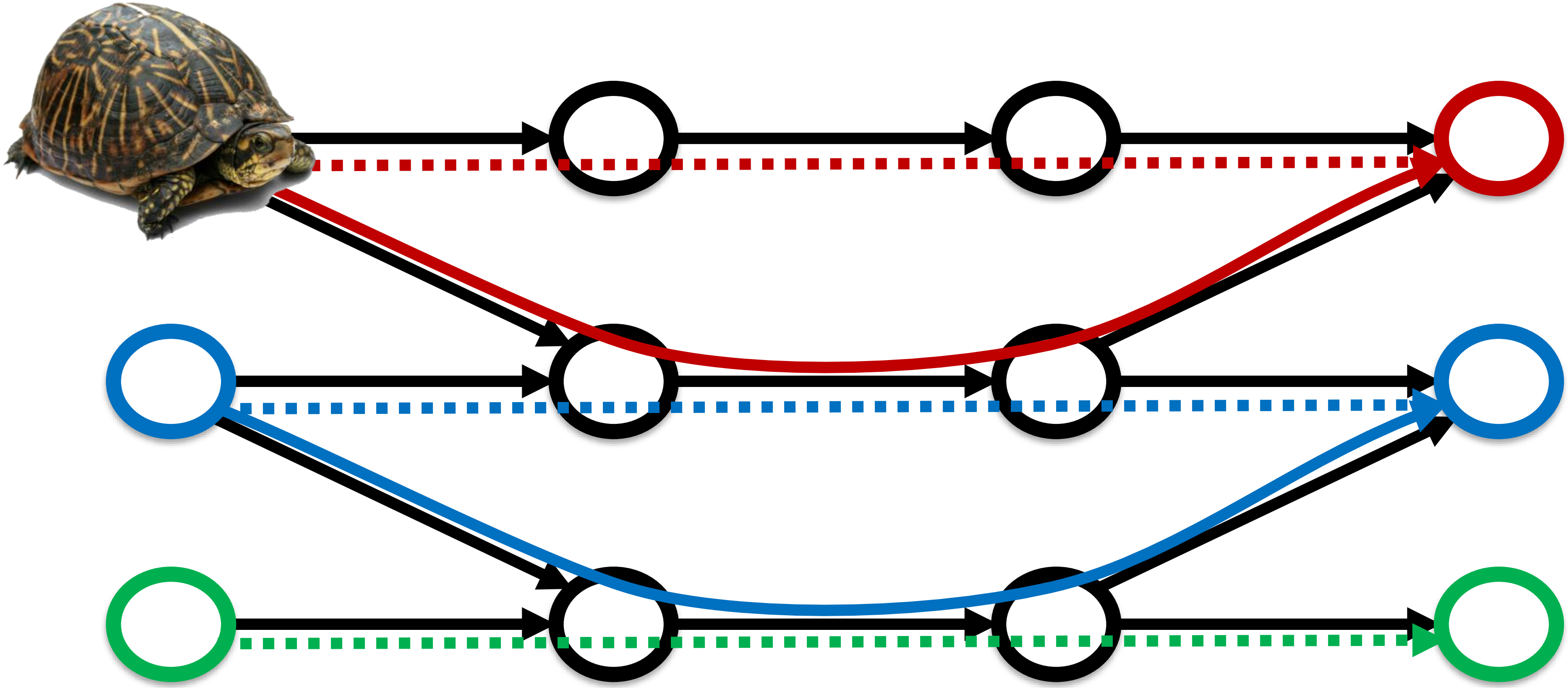
This would work



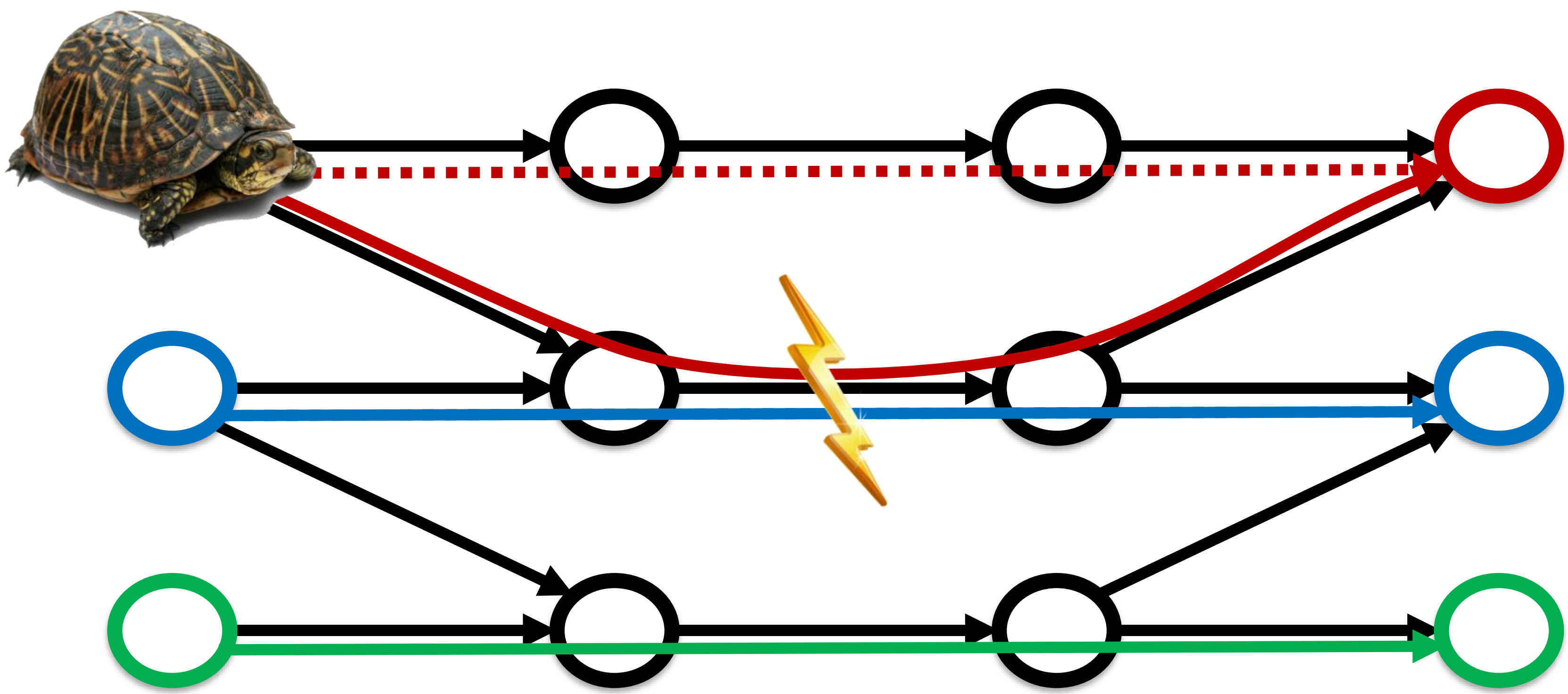
So lets go back



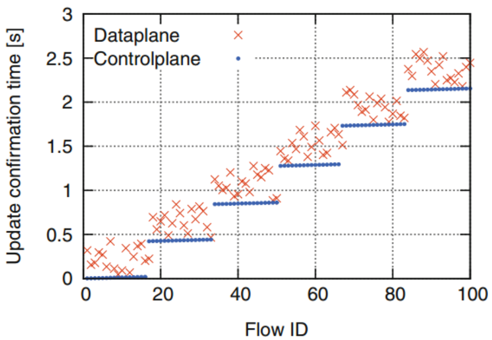
But Red is a bit Slow..



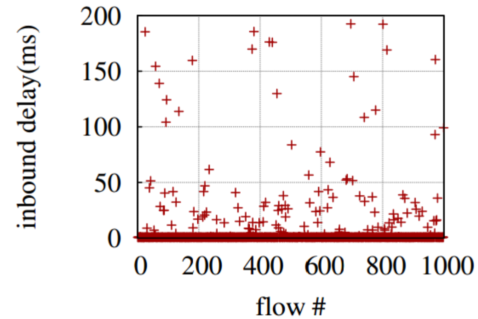
Congestion Again!



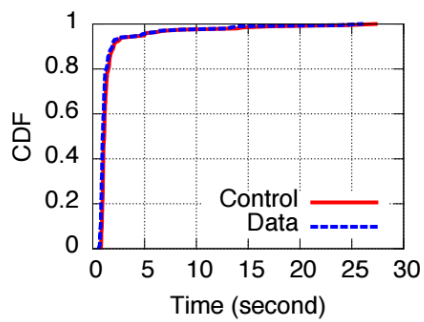
Appears in Practice



"Data plane ***updates may fall behind*** *the control plane acknowledgments and may be even* ***reordered****."*
Kuzniar et al., PAM 2015

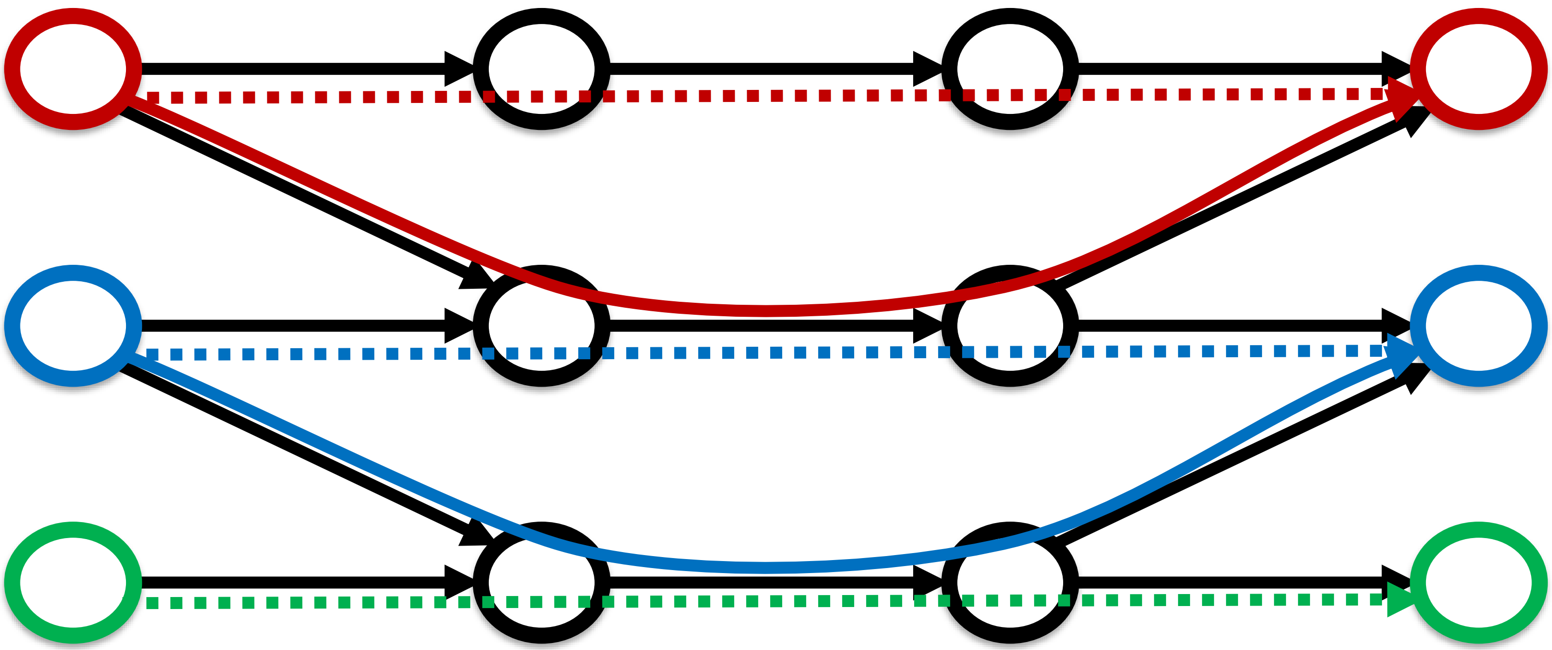


"…the inbound latency is ***quite variable*** *with a […] standard deviation of 31.34ms…"*
He et al., SOSR 2015

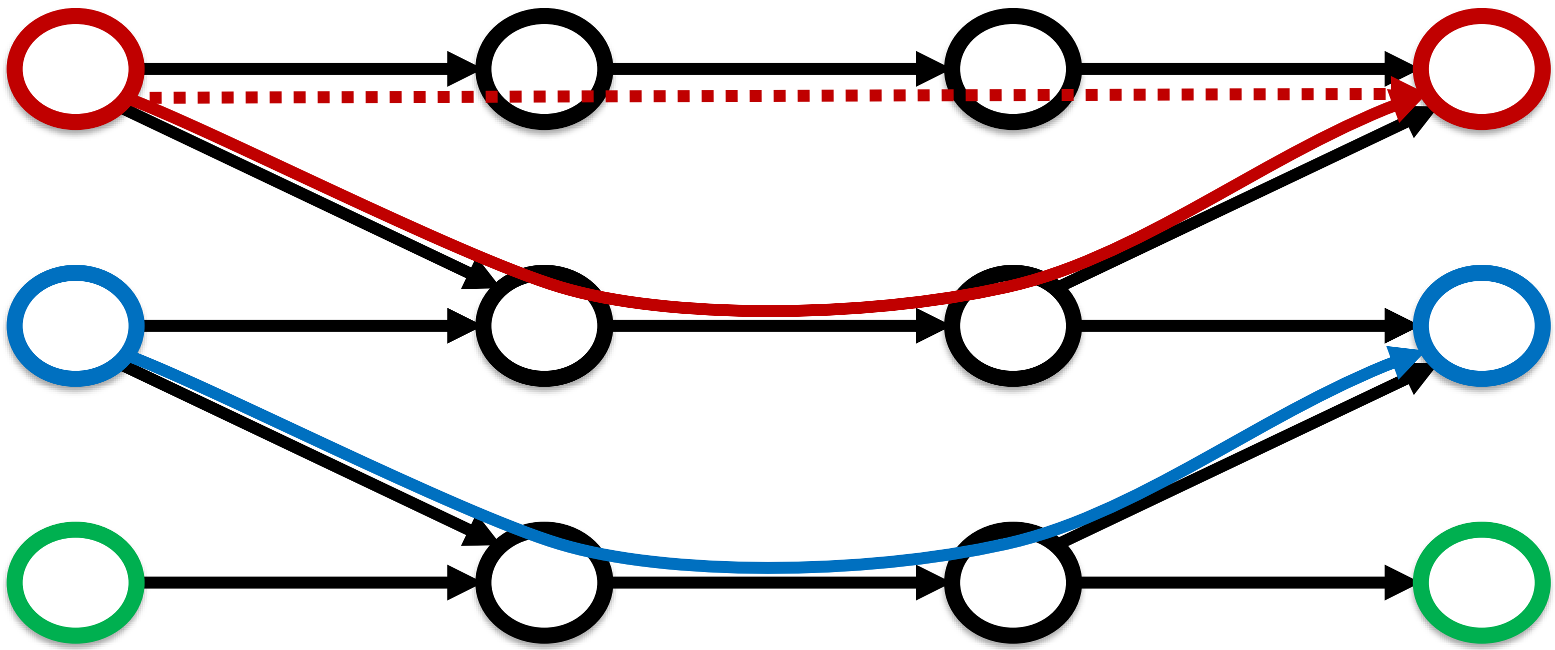


*"some switches can '****straggle****,' taking substantially* ***more time*** *than average (e.g., 10-****100x****) to apply an update"*
Jin et al., SIGCOMM 2014

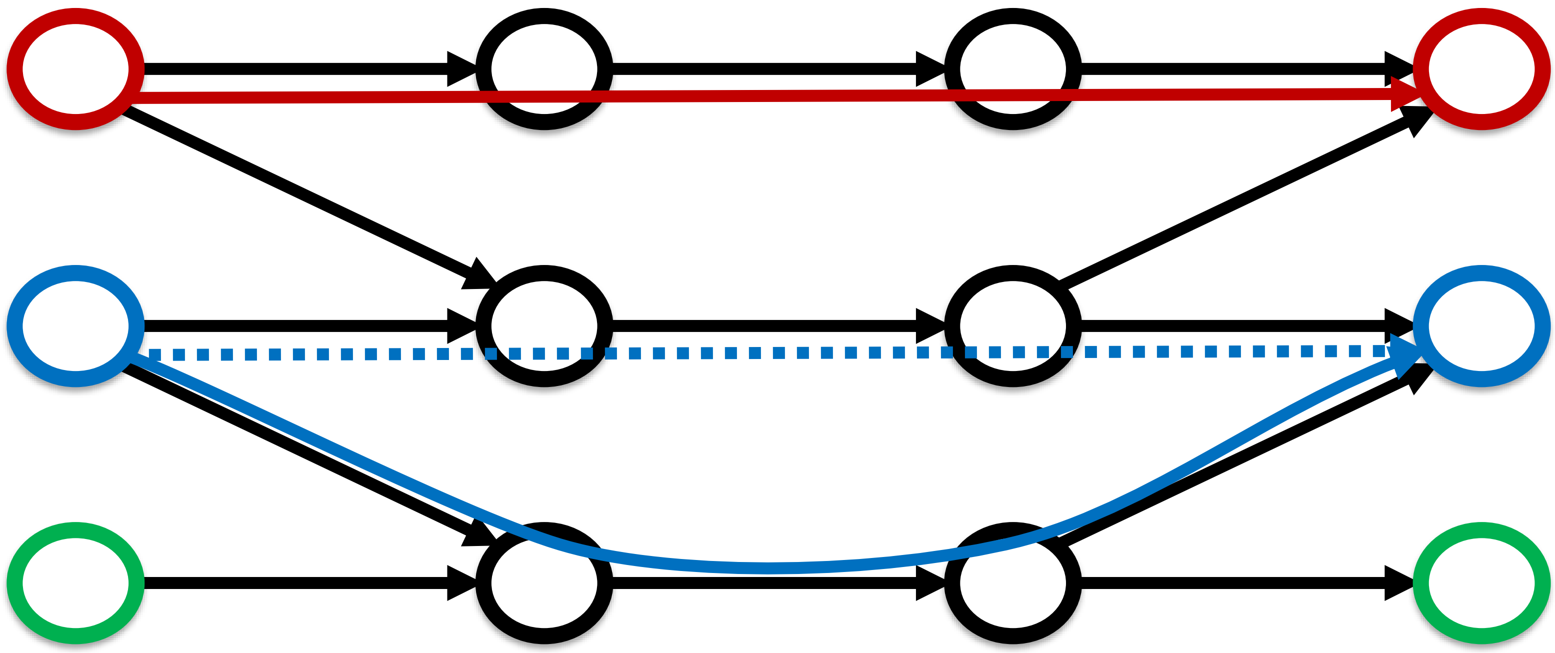
So lets go Back …



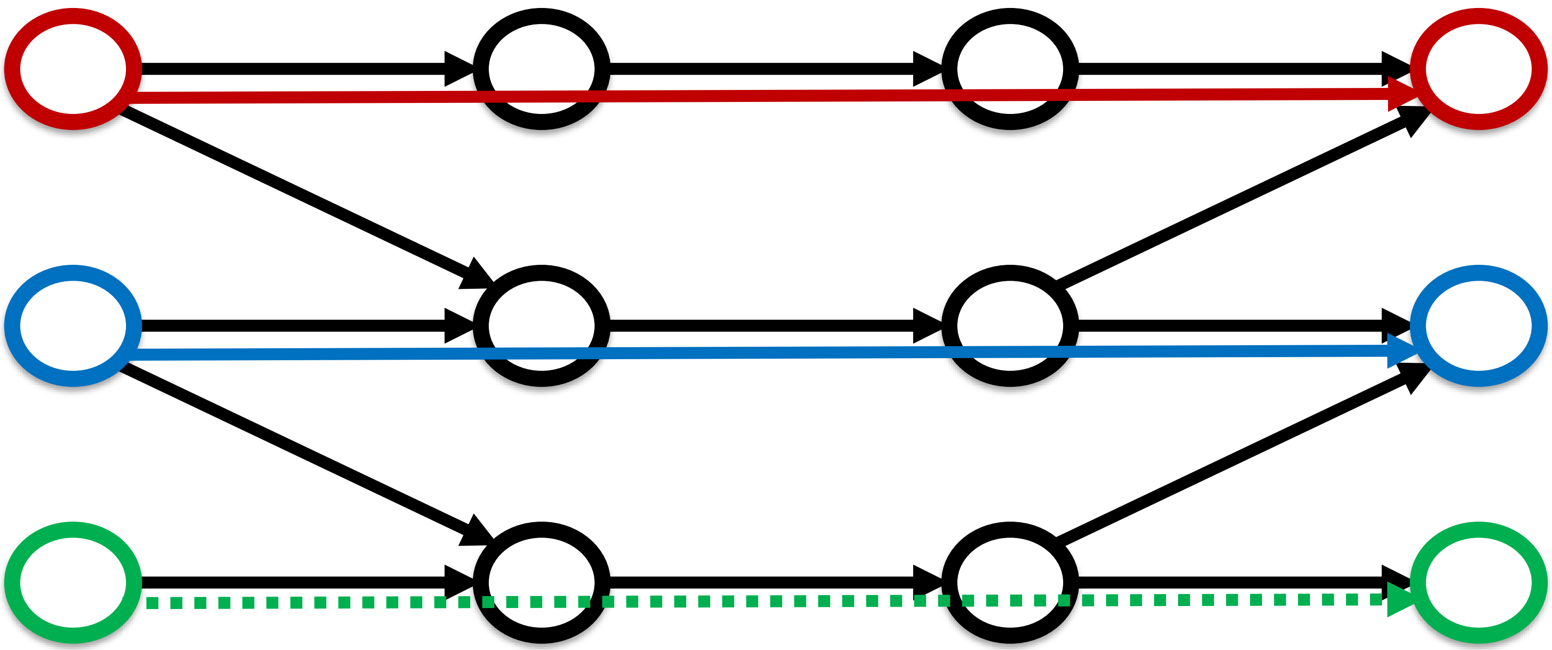
First, Red switches



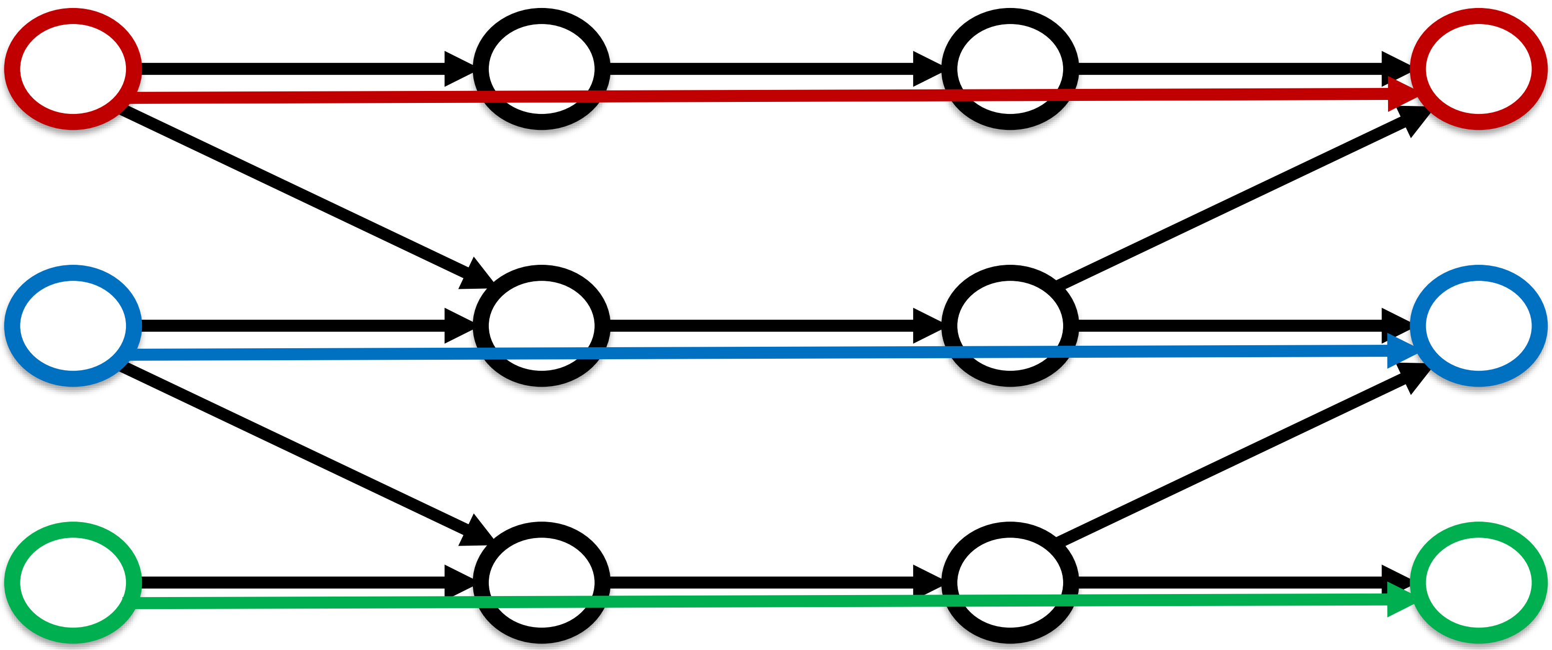
Then, Blue …



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: $\sum_{\forall F} \max\left(\mathbf{old}, \mathbf{new}\right) \leq capacity$

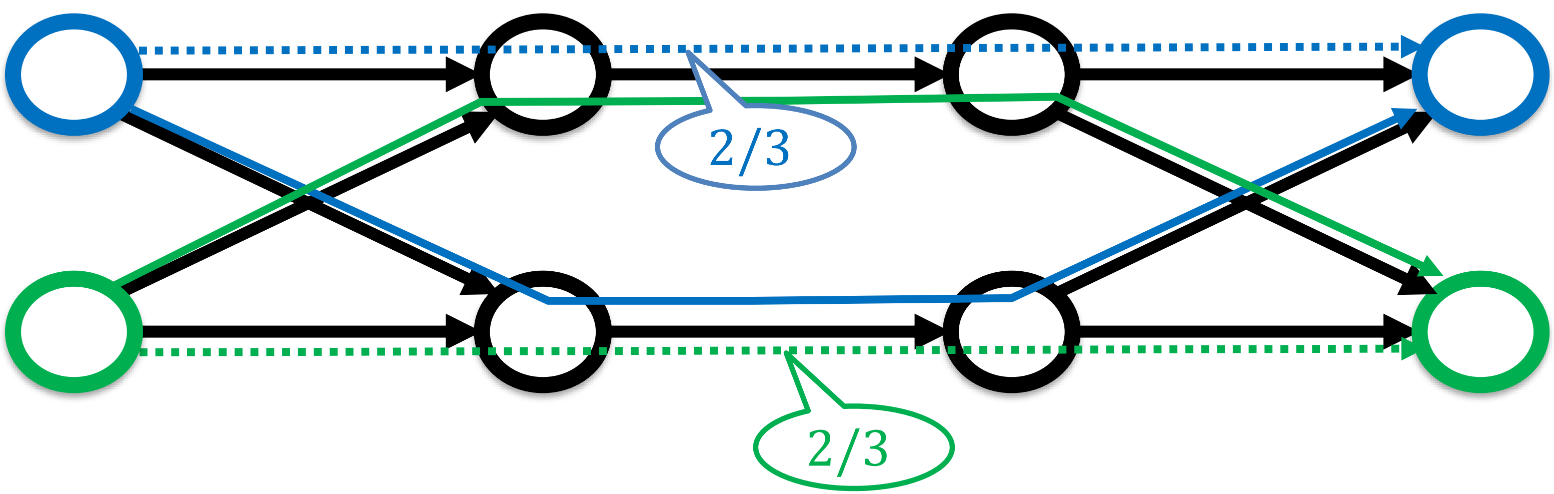
Consistent Migration of Flows

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

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

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

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

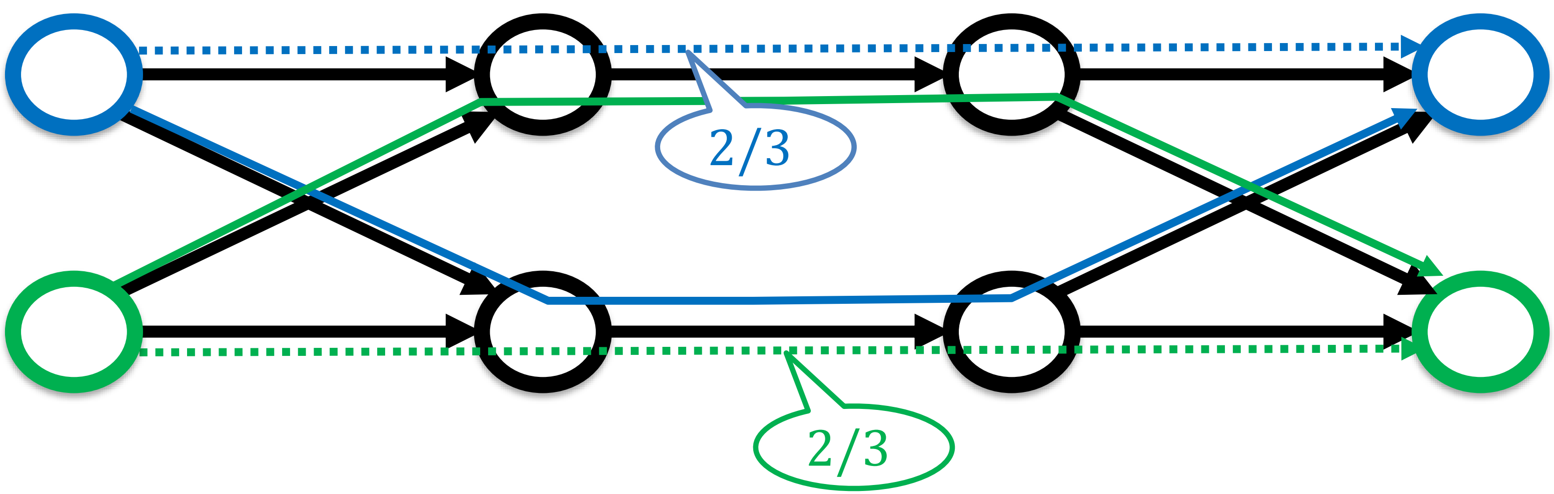


Consistent Migration of Flows

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

Here $x = 1/3$

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



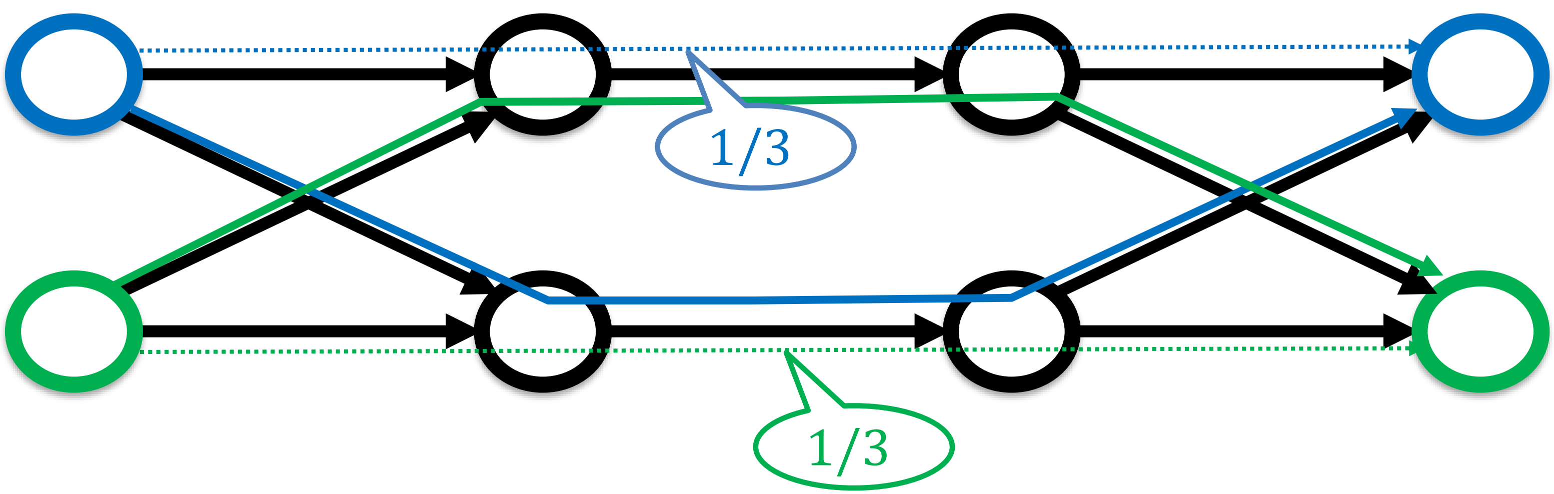
Consistent Migration of Flows

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



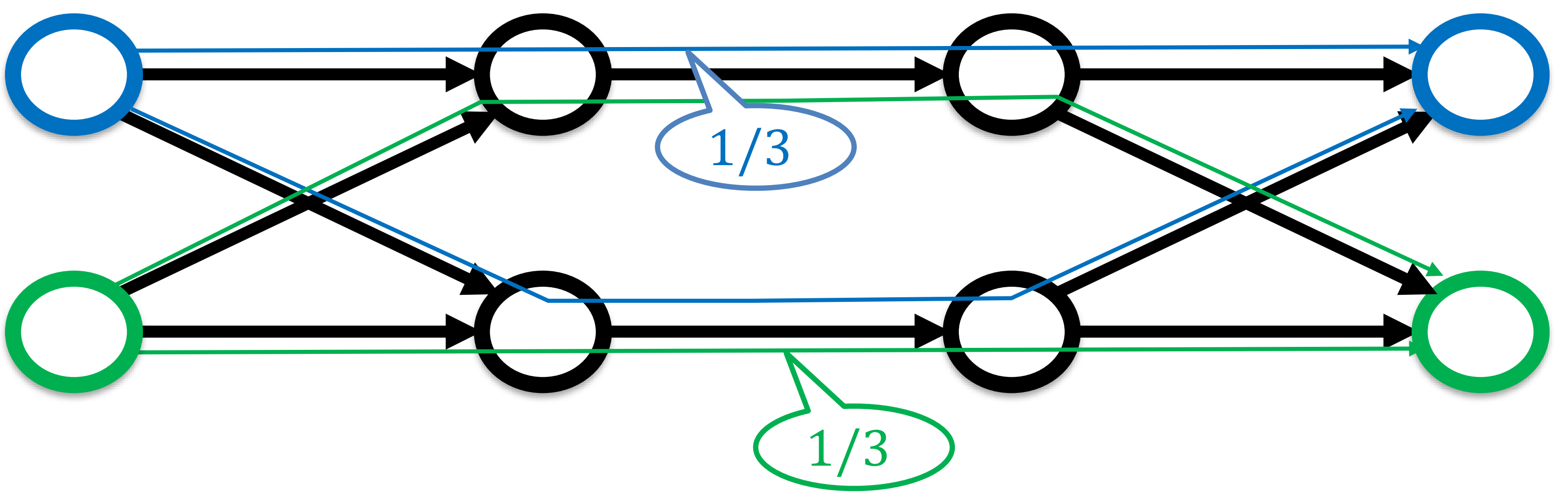
Consistent Migration of Flows

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



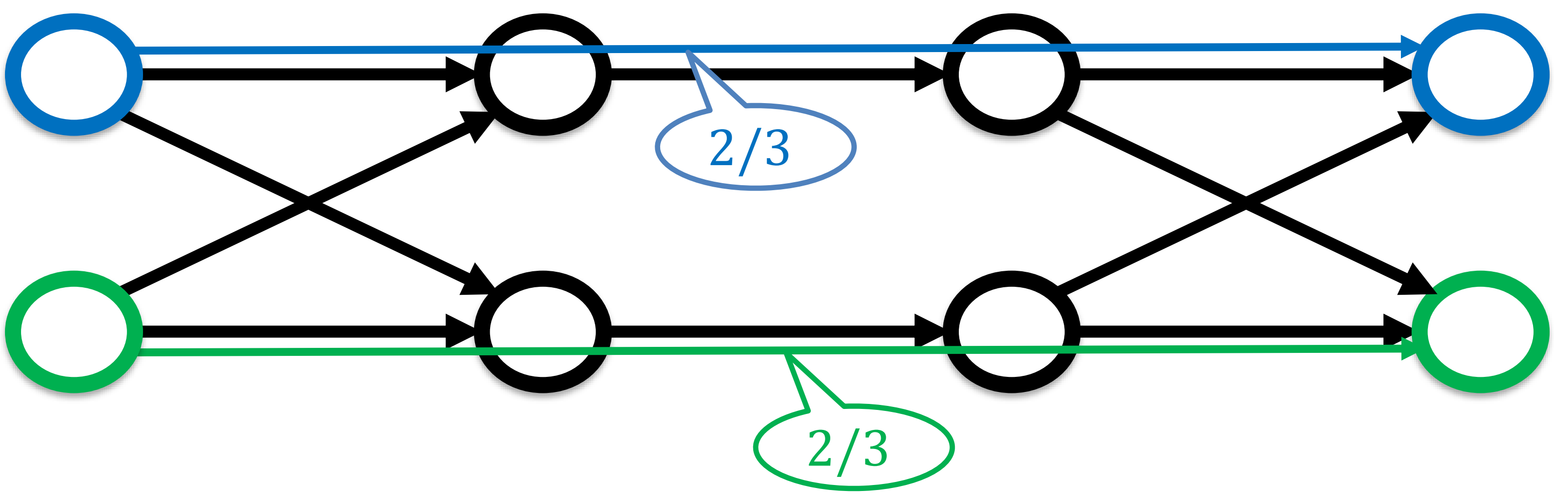
Consistent Migration of Flows

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

Here $x = 1/3$

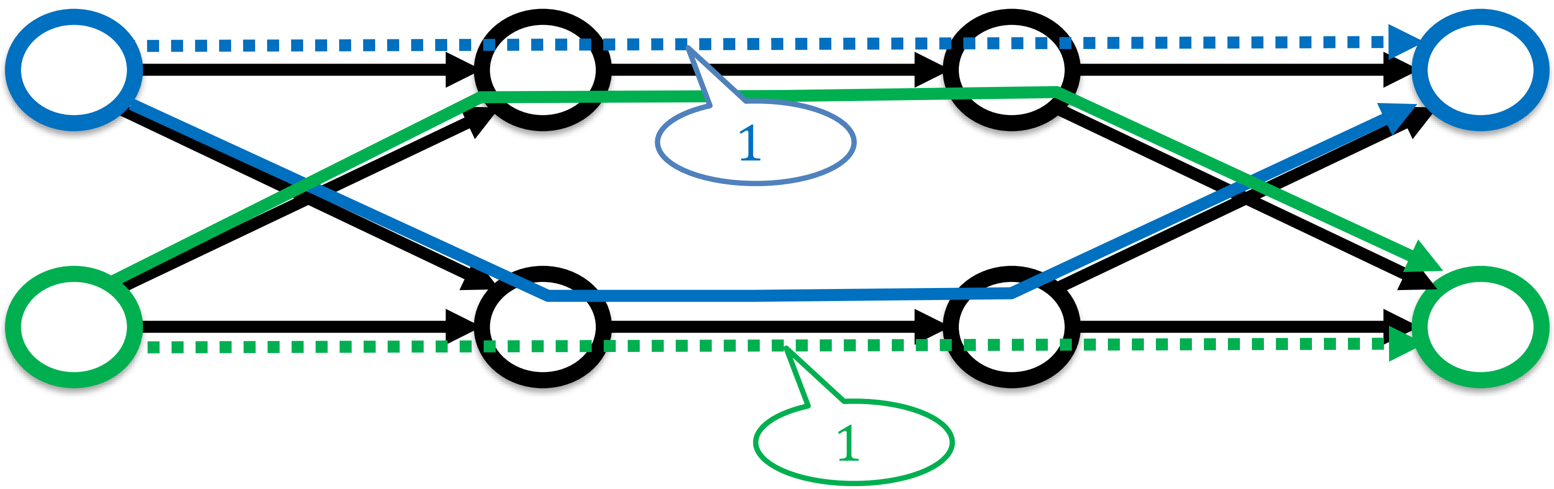
Move slack $x \Rrightarrow \lceil 1/x \rceil - 1$ staged partial moves

Update 2 of 2



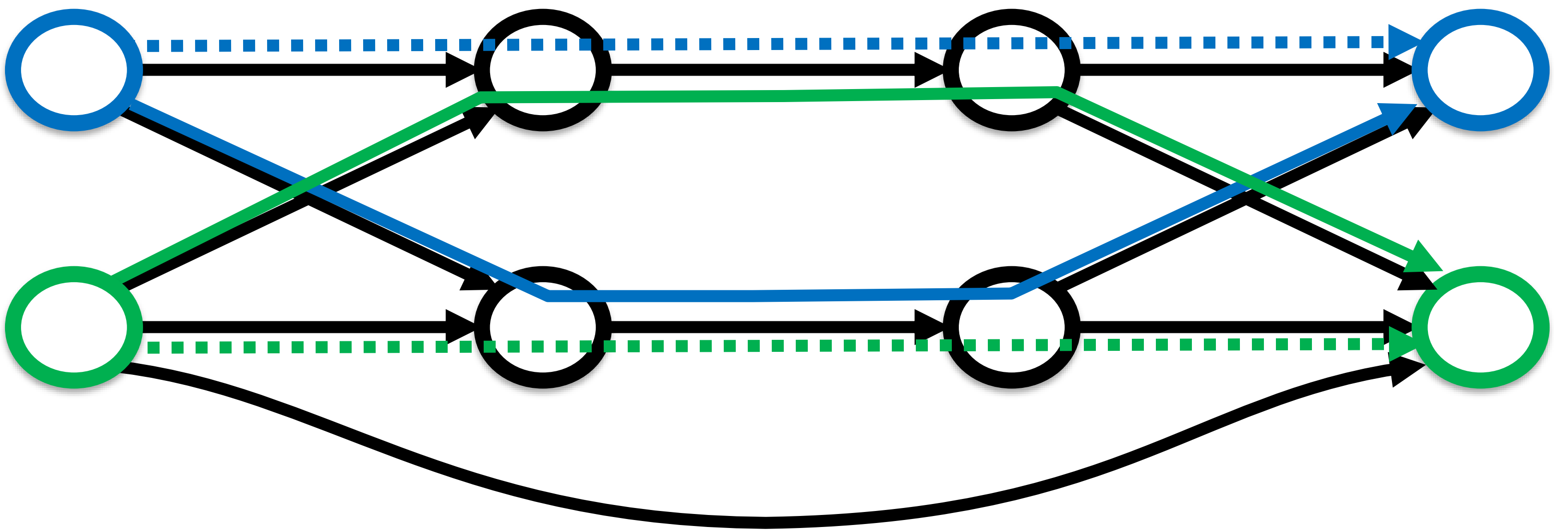
Consistent Migration of Flows

No slack on flow edges?



Consistent Migration of Flows

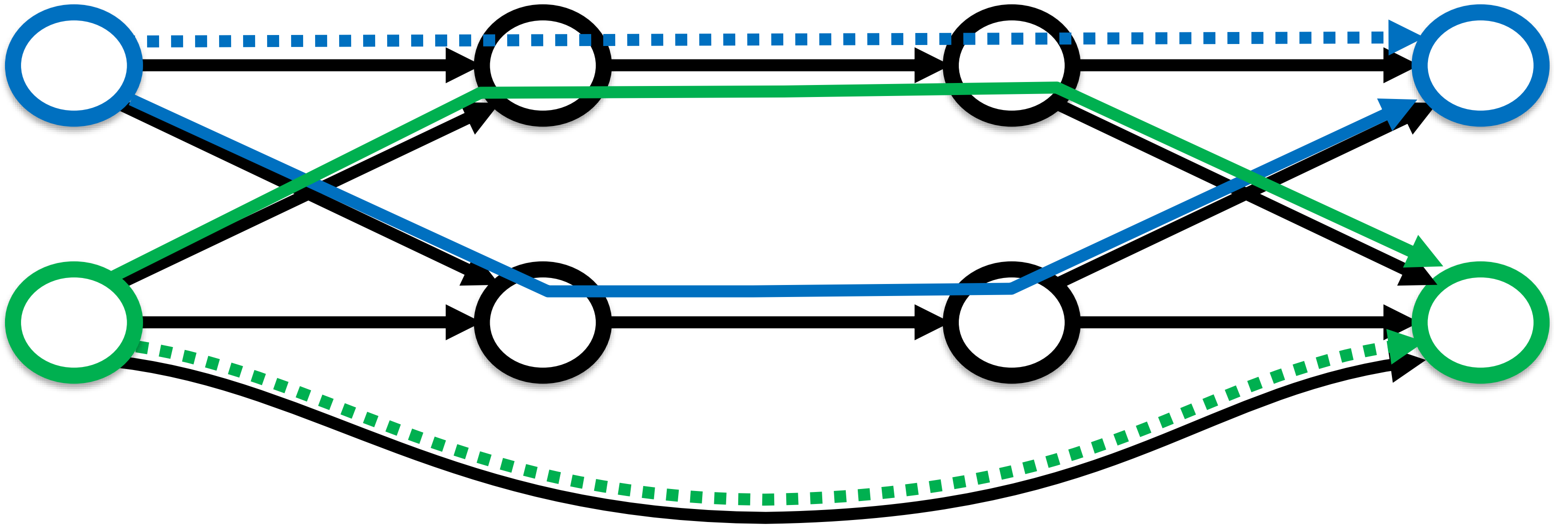
Alternate routes?



Consistent Migration of Flows

Think: variable swapping of b & g

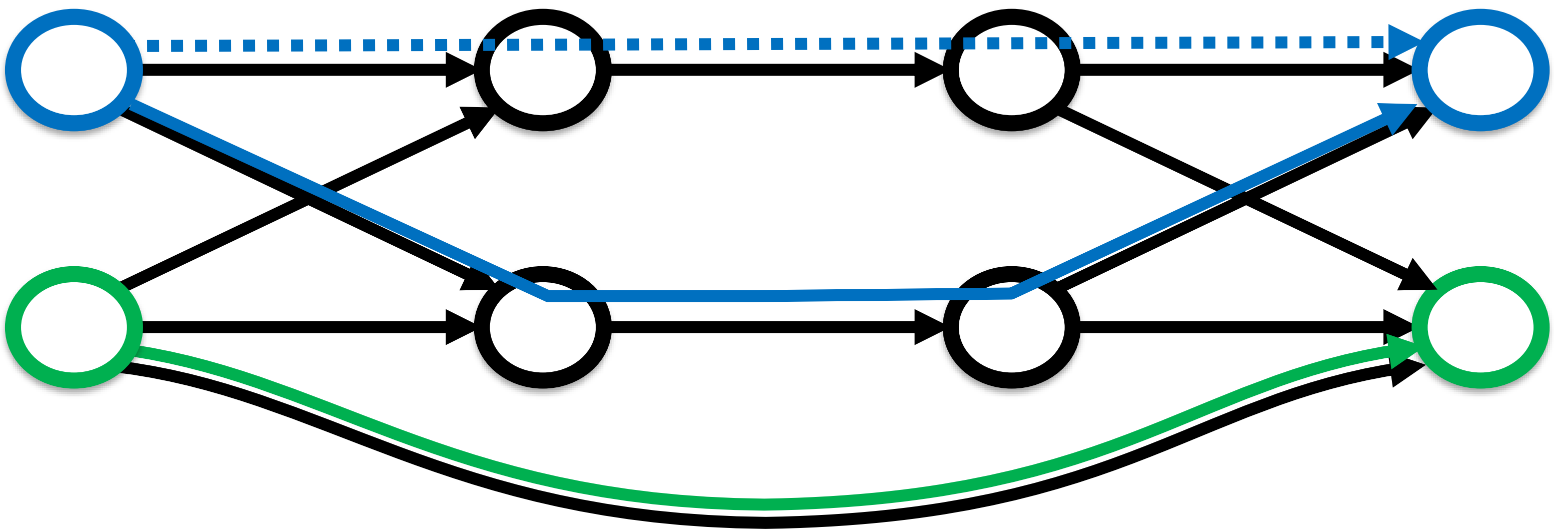
1. $x \coloneqq b$, 2. $\mathrm{b} \coloneqq g$, 3. $g \coloneqq x$



Consistent Migration of Flows

Think: variable swapping of b & g

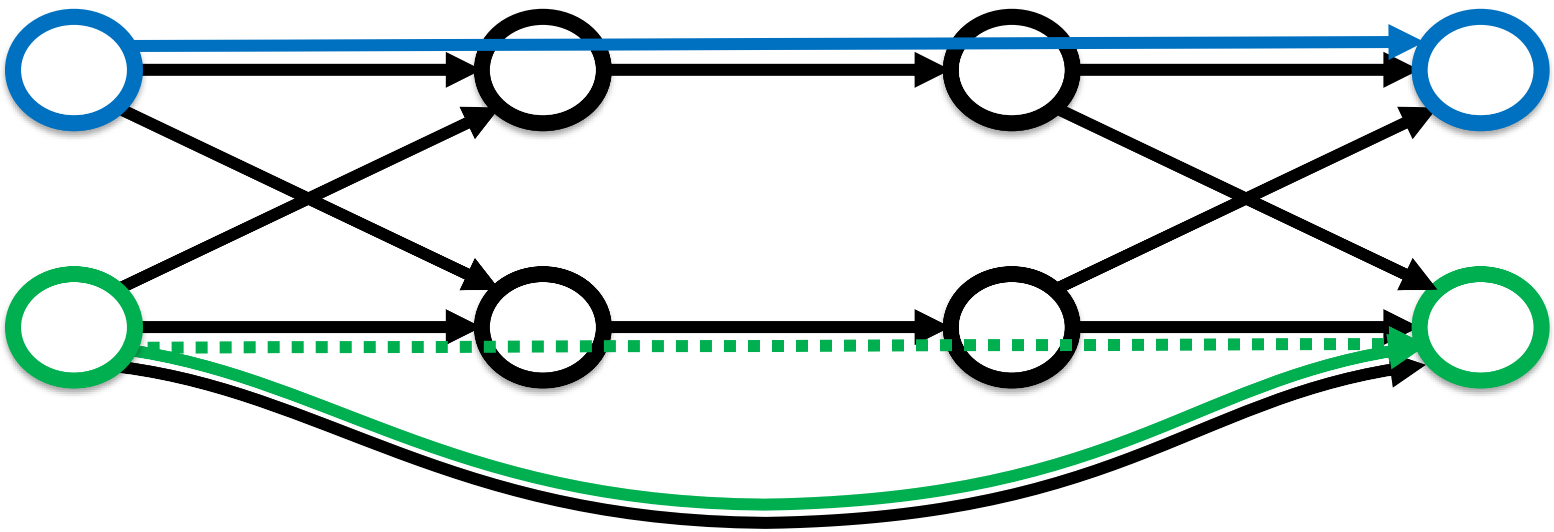
1. $x \coloneqq b$, 2. $\mathrm{b} \coloneqq g$, 3. $g \coloneqq x$



Consistent Migration of Flows

Think: variable swapping of b & g

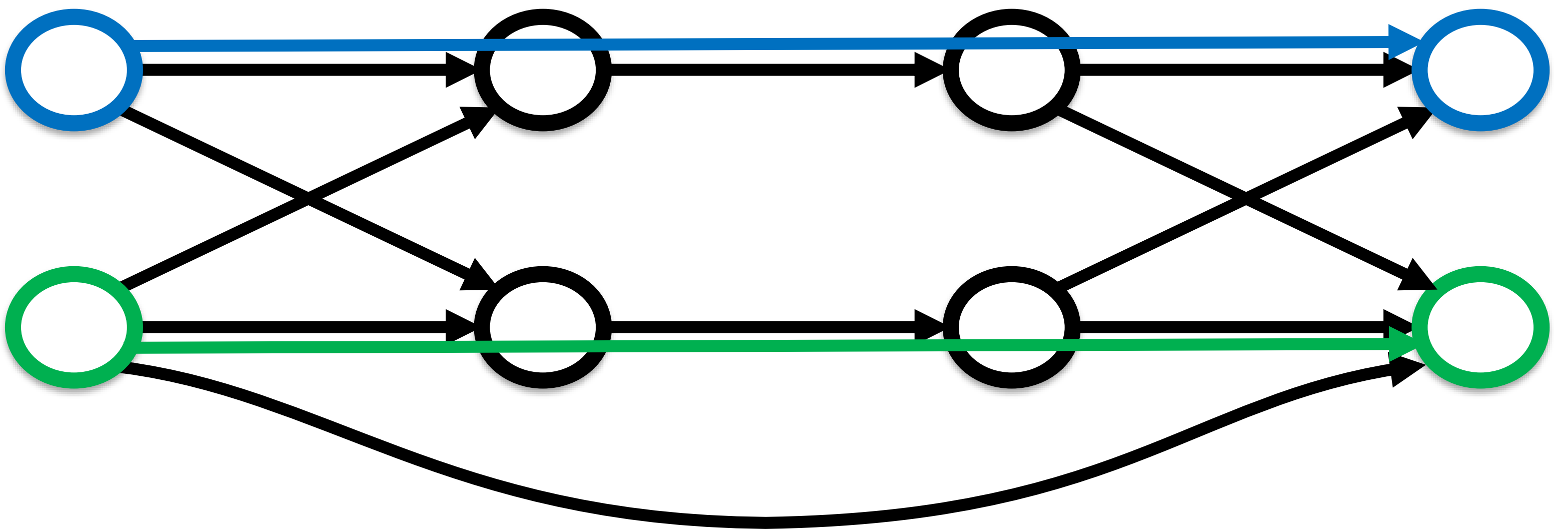
1. $x \coloneqq b$, 2. $\mathrm{b} \coloneqq g$, 3. $g \coloneqq x$



Consistent Migration of Flows

SWAN: LP-approach with binary search

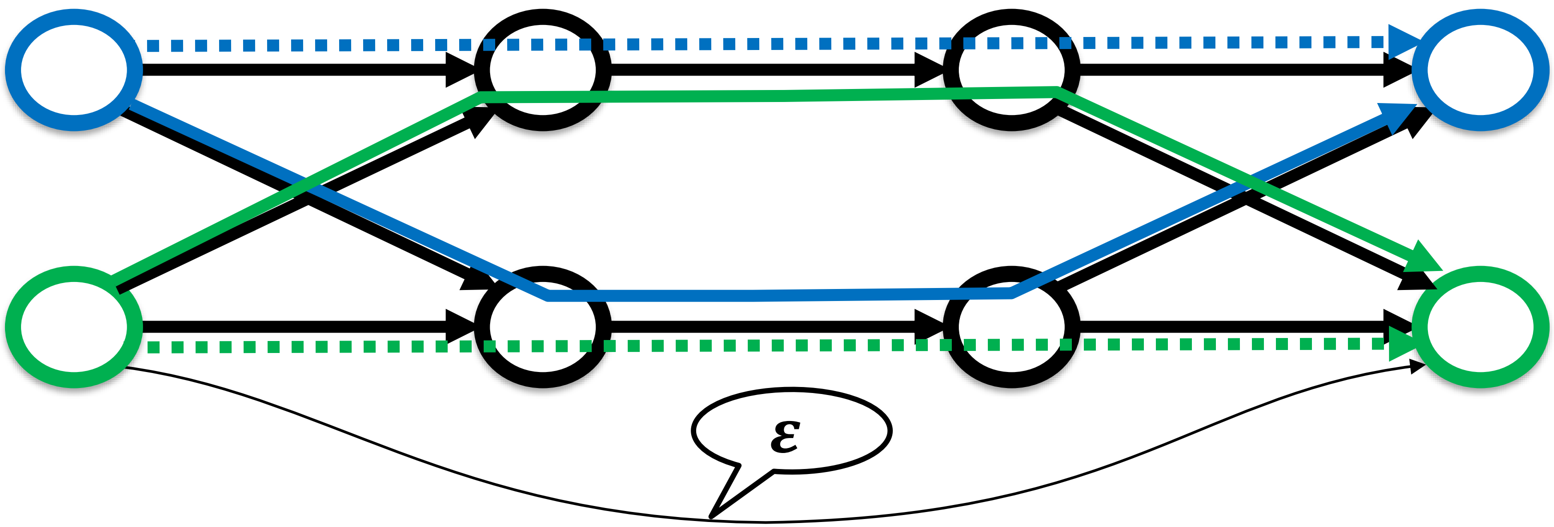
1 update? 2 updates? 4 updates? …



Consistent Migration of Flows

SWAN: LP-approach with binary search

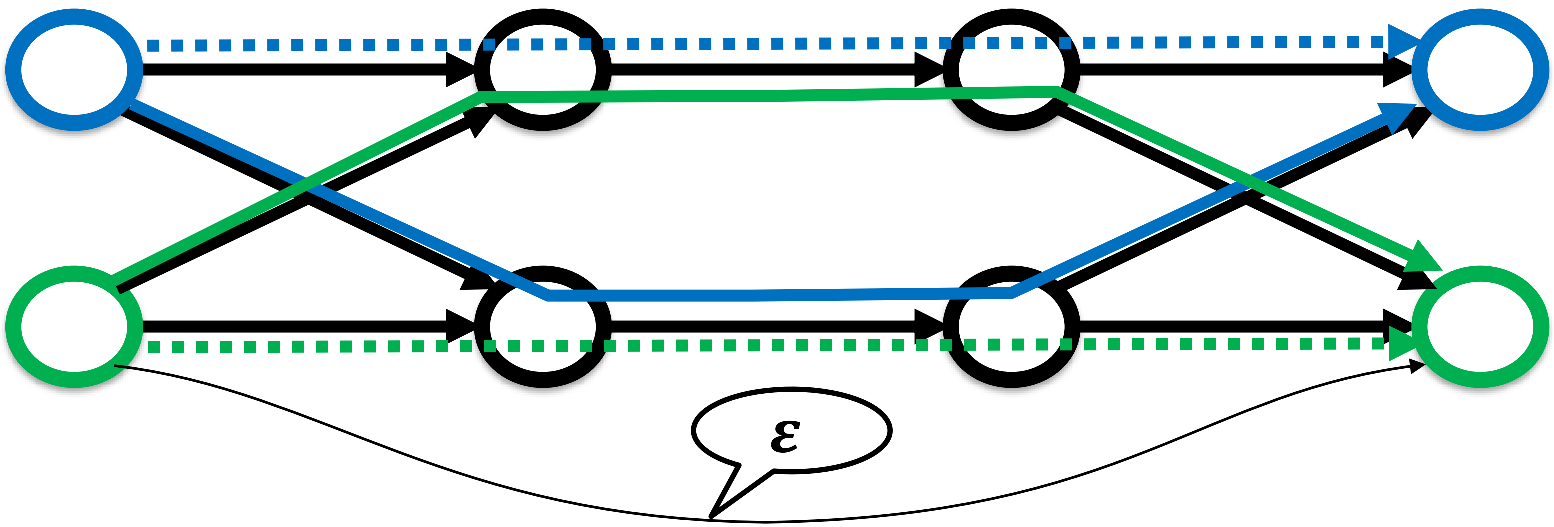
1 update? 2 updates? 4 updates? …



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?

Overview of the Remaining Talk

1. **Yes, we can (decide in polynomial time)**

2. What to do if we cannot migrate consistently?

3. Last: NP-hardness for unsplittable flows

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

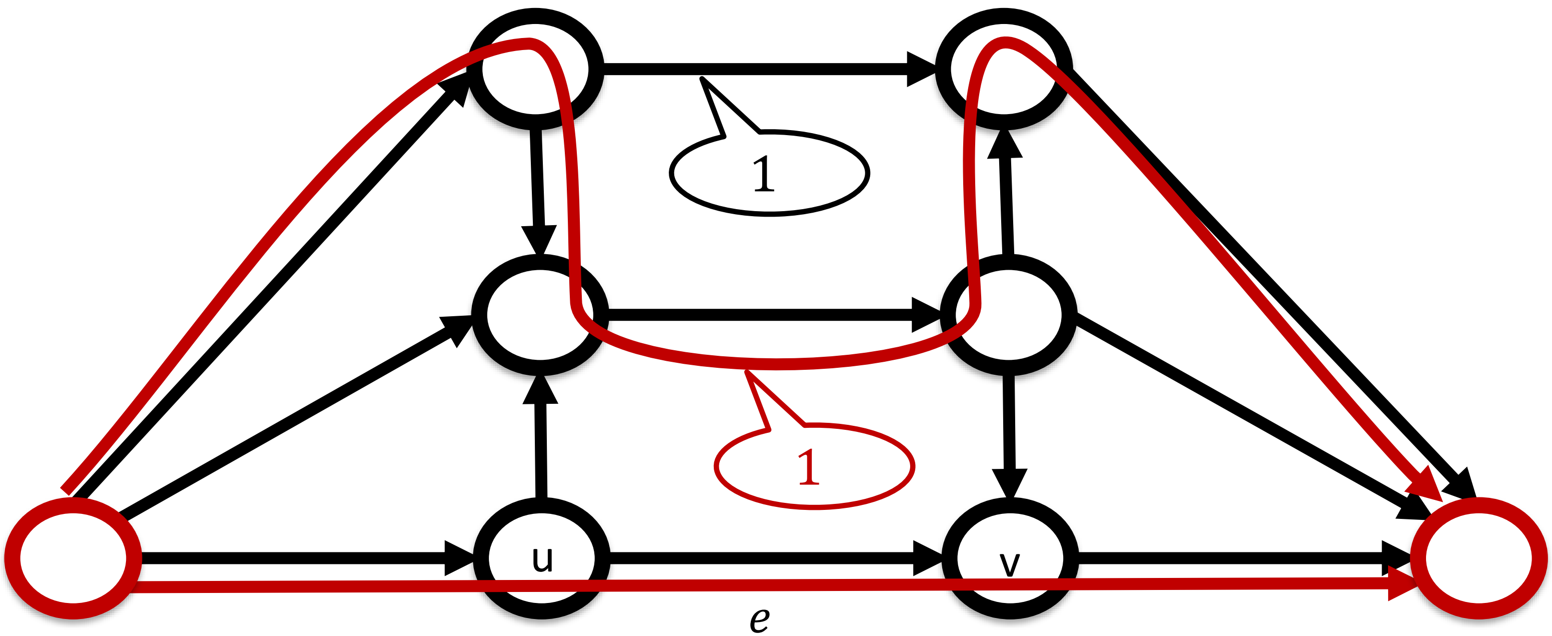
To Slack or not to Slack?

Combinatorial approach

Augmenting paths

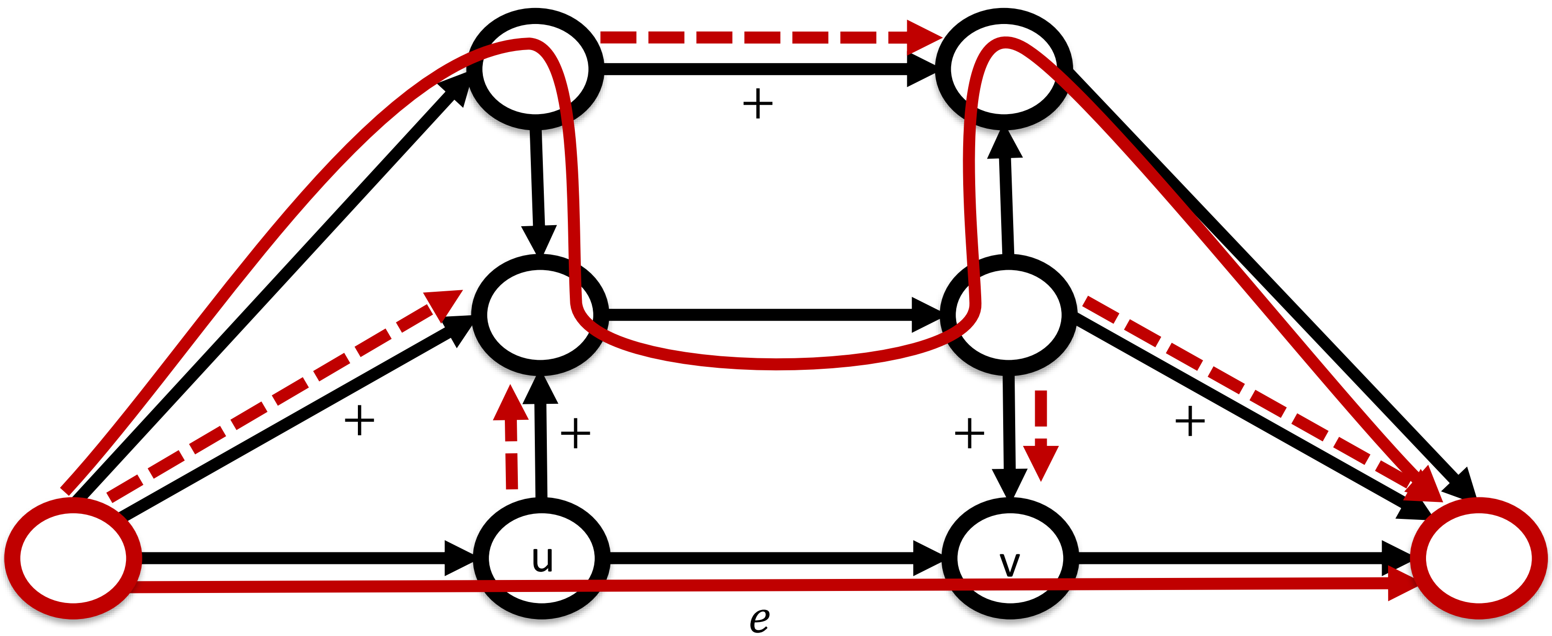
Combinatorial Approach

Move single commodities at a time



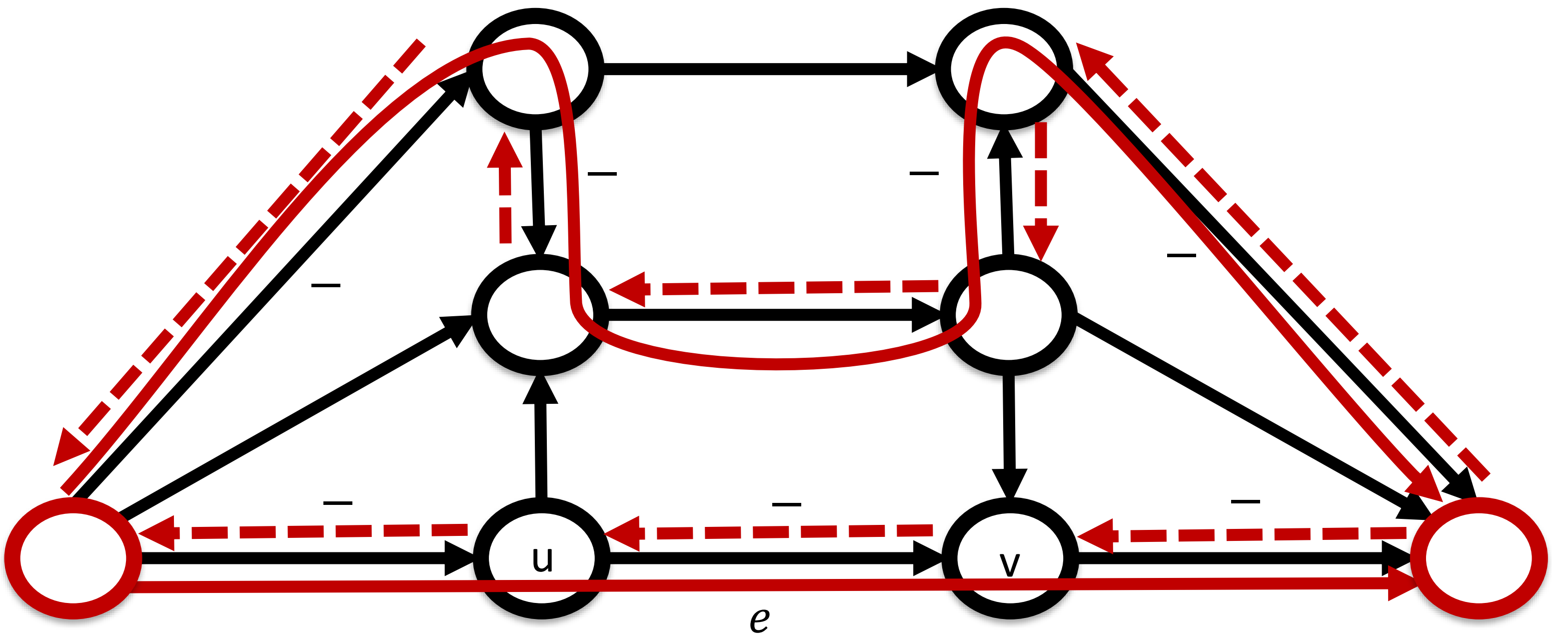
Combinatorial Approach

Where to increase flow?



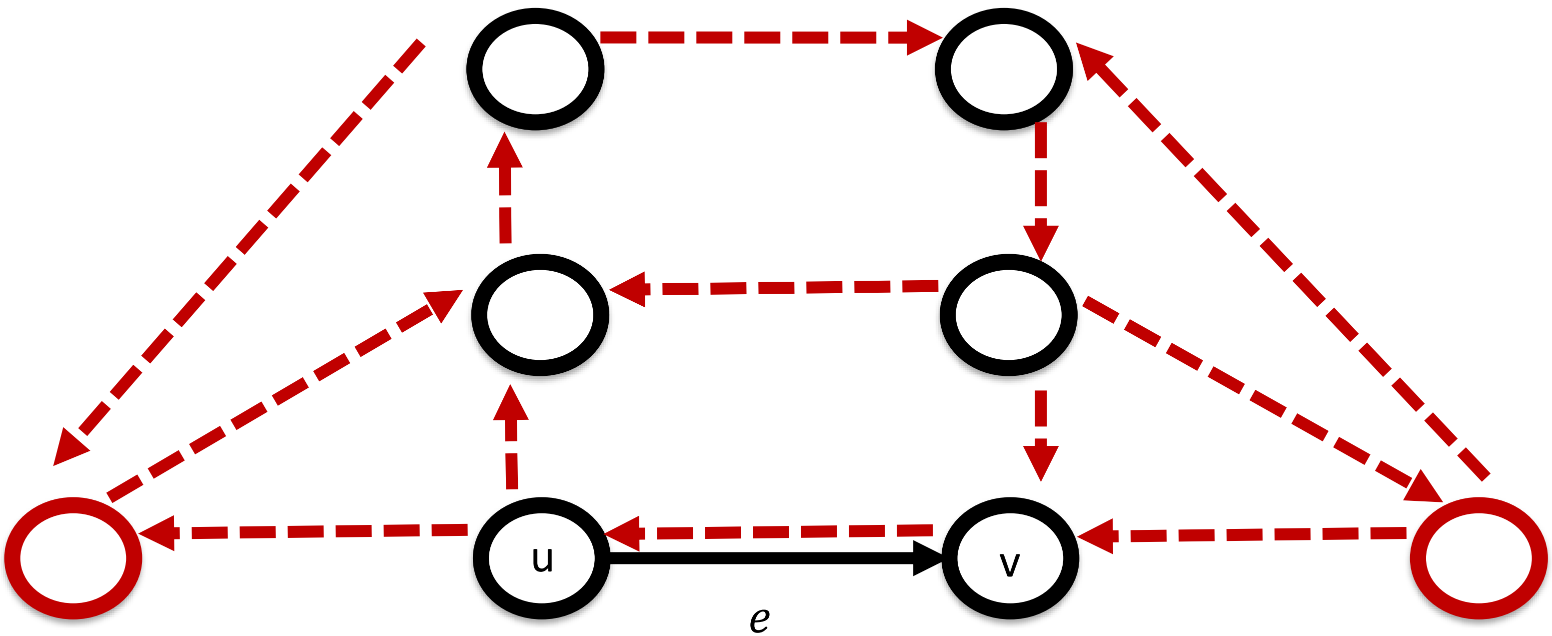
Combinatorial Approach

Where to push back flow?



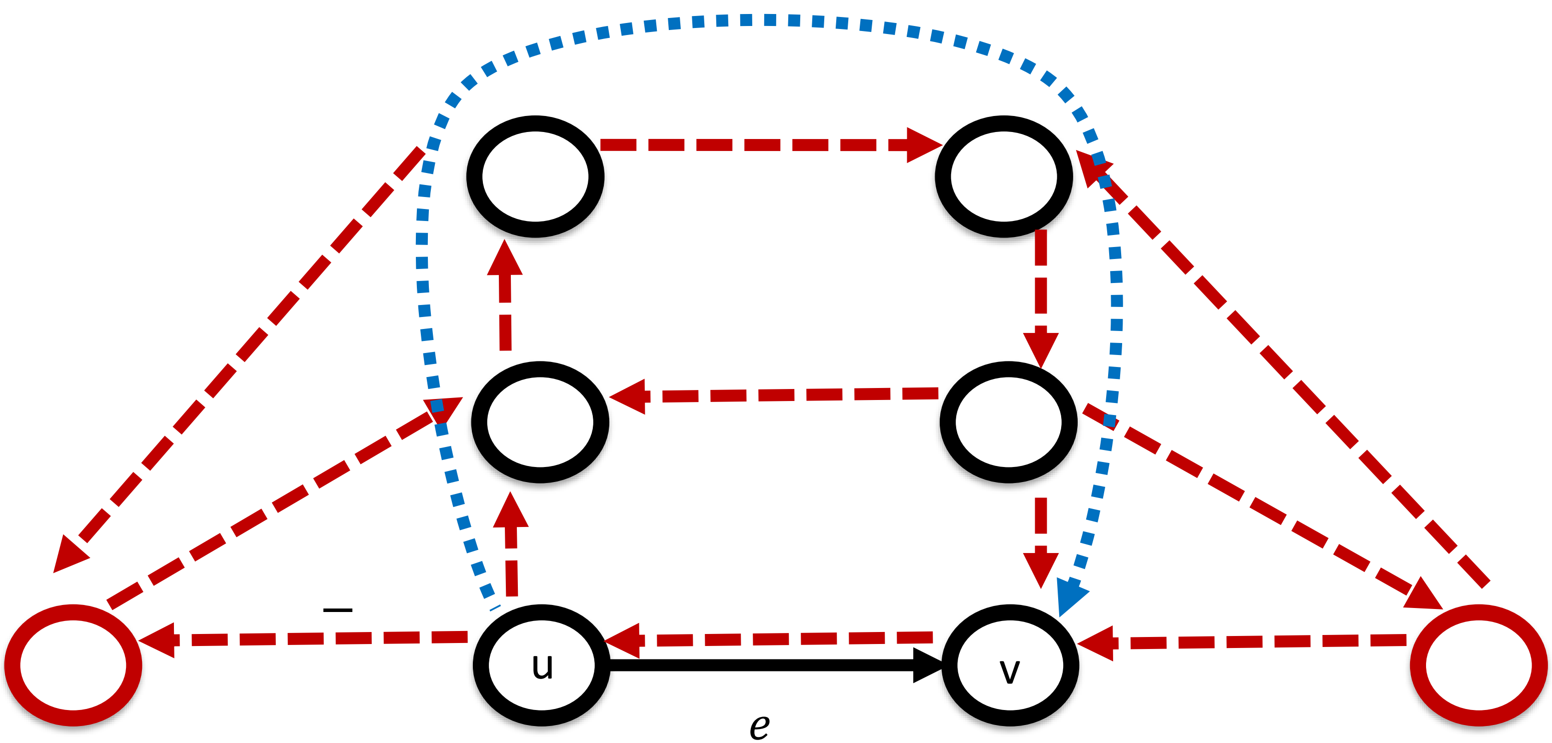
Combinatorial Approach

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)

Overview of the Remaining Talk

1. Yes, we can (decide in polynomial time)

2. **What to do if we cannot migrate consistently?**

3. Last: NP-hardness for unsplittable flows

What to do if we cannot Migrate Consistently?

Option 1: Migrate, but reduce congestion

- *B4:* Optimize (Jain et al., SIGCOMM 13)
- Dionysus: Rate-limit some flows (Jin et al., SIGCOMM 14)
- Time-based updates (E.g., Mizrahi et al., INFOCOM 15/16)
- …

What to do if we cannot Migrate Consistently?

Option 1: Migrate, but reduce congestion

B4: Optimize (Jain et al., SIGCOMM 13)

Dionysus: Rate-limit some flows (Jin et al., SIGCOMM 14)

Time-based updates (E.g., Mizrahi et al., INFOCOM 15/16)

… ***Wed, 08:30: Grand Ballroom B***

Option 2: Increase Demands Consistently

Maximize $\sum_{i:e_i \in \texttt{out}(s_1)} x_{i1}$
subject to

1) $\forall 1 \leq j \leq k \forall v \in V \setminus \{s_j, t_j\}$:
   $\sum_{i:e_i \in \texttt{out}(v)} x_{ij} = \sum_{i:e_i \in \texttt{in}(v)} x_{ij}$,
2) $\forall 2 \leq j \leq k : \sum_{i:e_i \in \texttt{out}(s_j)} x_{ij} = d_j = \sum_{i:e_i \in \texttt{in}(t_j)} x_{ij}$,
3) $\forall 1 \leq j \leq k : \sum_{i=1}^{k} x_{ij} \leq c(e_j)$,
4) $\forall 1 \leq i \leq m$ s.t. $e_i \in E_{\texttt{fix}} \forall 1 \leq j \leq k : x_{ij} = F_j(e_i)$,
5) $\sum_{i:e_i \in \texttt{in}(s_1)} x_{i1} = 0$.

Idea: Don’t change where no slack is possible

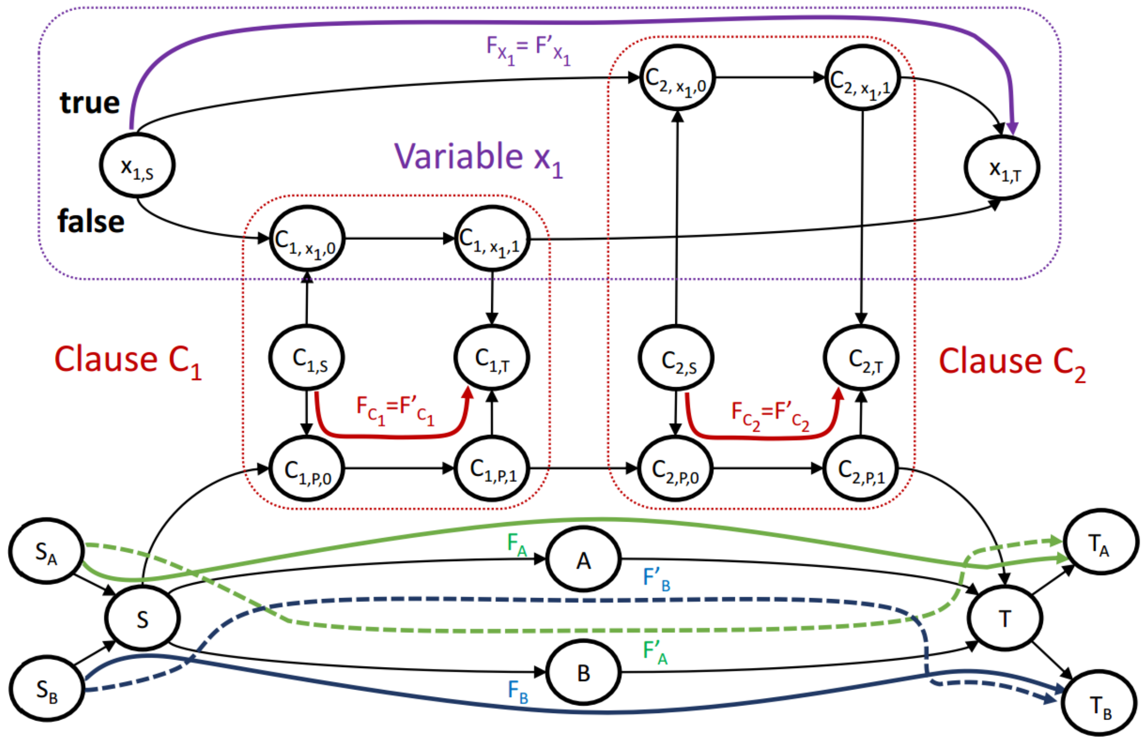
Overview of the Remaining Talk

1. Yes, we can (decide in polynomial time)

2. What to do if we cannot migrate consistently?

3. **Last: NP-hardness for unsplittable flows**

NP-Hardness for Unsplittable Flows



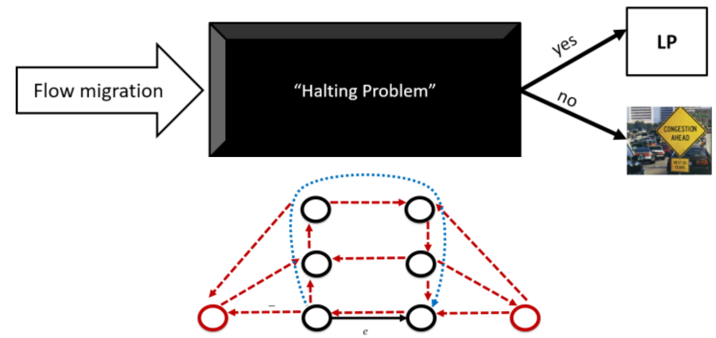
Reduction from 3-Satisfiability

(here: $(x_1) \wedge (\neg x_1)$)

Summary

Consistent migration of flows

Decidable in polynomial time



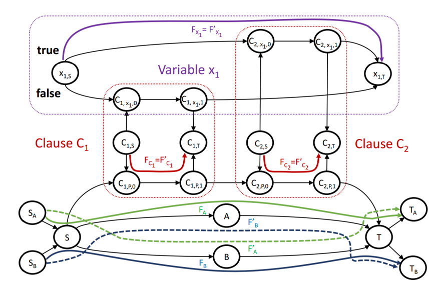
No consistent migration possible?

Can use LP to maximize demands

Maximize $\sum_{i:e_i\in\mathrm{out}(s_1)} x_{i1}$
subject to
1) $\forall 1\le j\le k \forall v\in V\setminus\{s_j,t_j\}:$
$\sum_{i:e_i\in\mathrm{out}(v)} x_{ij} = \sum_{i:e_i\in\mathrm{in}(v)} x_{ij}$,
2) $\forall 2\le j\le k: \sum_{i:e_i\in\mathrm{out}(s_j)} x_{ij} = d_j = \sum_{i:e_i\in\mathrm{in}(t_j)} x_{ij}$,
3) $\forall 1\le j\le k: \sum_{i=1}^{k} x_{ij} \le c(e_j)$,
4) $\forall 1\le i\le m$ s.t. $e_i\in E_{\mathrm{fix}} \forall 1\le j\le k: x_{ij} = F_j(e_i)$,
5) $\sum_{i:e_i\in\mathrm{in}(s_1)} x_{i1} = 0$.

Unsplittable flow migration

NP-hard



On Consistent Migration of Flows in SDNs

*Sebastian Brandt, **<u>Klaus-Tycho Förster</u>**, Roger Wattenhofer*

April 12, 2016 @ INFOCOM 2016 – San Francisco