universität
wien

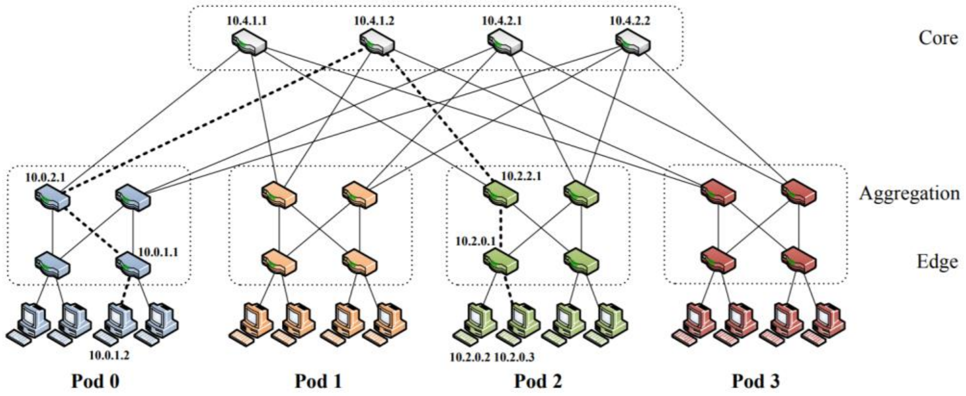
Efficient Non-Segregated Routing for Reconfigurable Demand-Aware Networks

Thomas Fenz, Klaus-Tycho Foerster, Stefan Schmid, and Anaïs Villedieu

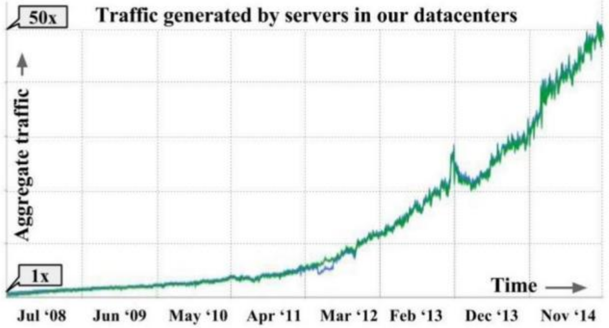
Today’s Data Center Topologies

- Often *Clos*-based (e.g. *Fat-tree*)
  - Goal: optimize for all-to-all communication
    - Idea: Obtain good bisection bandwidth

From: Al-Fares et al. 2008



- However, traffic is growing at unprecedented rates
  - What can we do?
  - Exponentially bigger networks?

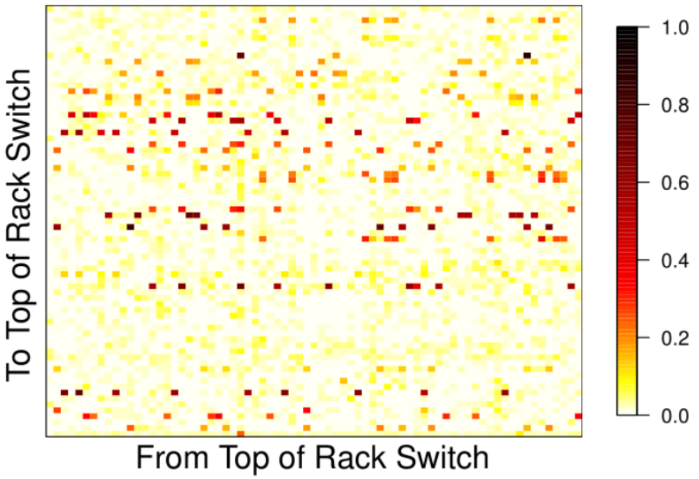


From Google’s Datacenter Network. Singh at al., SIGCOMM’15

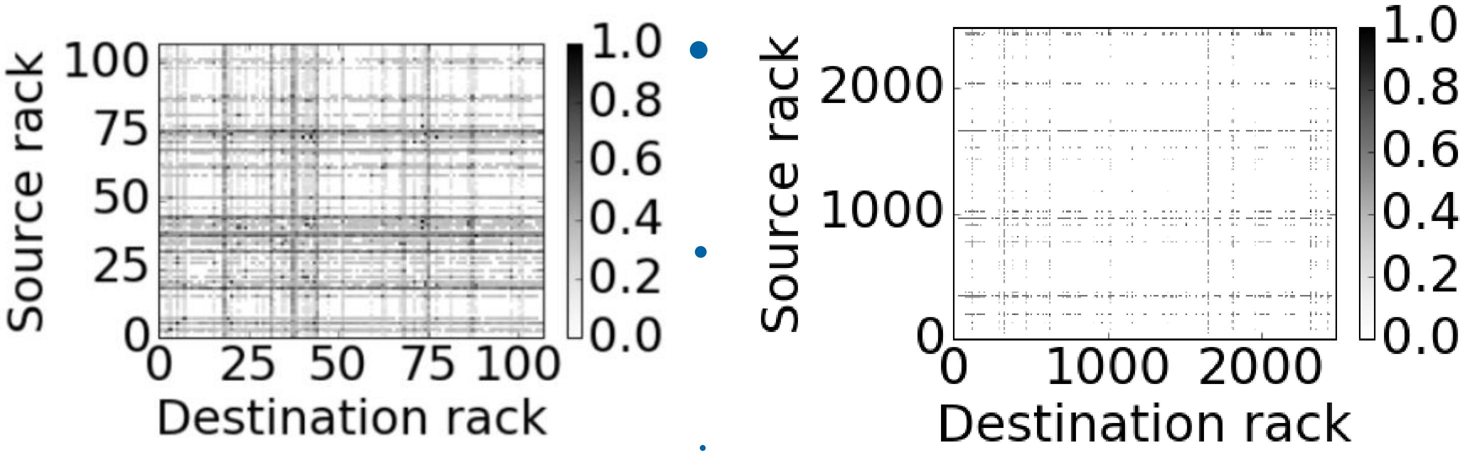
Data Center Traffic ≠ Uniform

- However, DCN traffic is often ***not*** all-to-all

“Data reveal that 46-99% of the rack pairs exchange no traffic at all”



Traffic demands (normalized) between ToR switches. Halperin et al., SIGCOMM’11

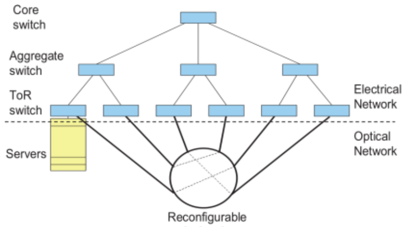


Heatmap of rack to rack traffic. Color intensity is log-scale and normalized. Ghobadi et al., SIGCOMM’16

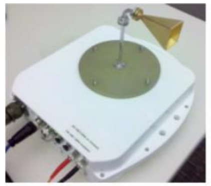
Programmable Physical Layer

What is different?

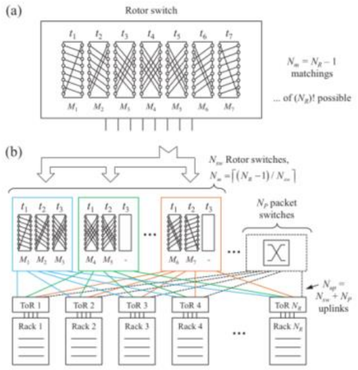
Motivation for Hybrid/Reconfigurable Data Center Topologies



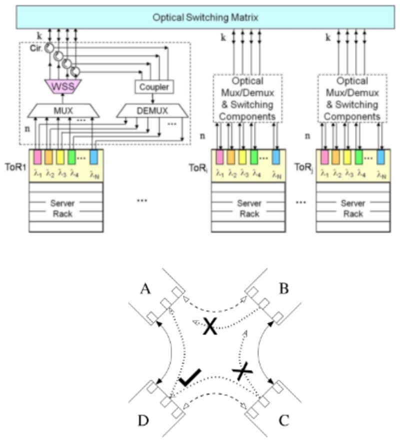
c-Through



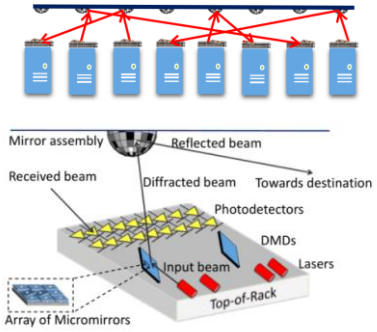
Flyways


(a)
Rotor switch
t_1 t_2 t_3 t_4 t_5 t_6 t_7
M_1 M_2 M_3 M_4 M_5 M_6 M_7
$N_m = N_R - 1$ matchings
... of $(N_R)!$ possible
(b)
N_{sw} Rotor switches, $N_m = \lceil (N_R - 1)/N_{sw} \rceil$
N_P packet switches
$N_{up} = N_{sw} + N_P$ uplinks
ToR 1
ToR 2
ToR 3
ToR 4
ToR N_R
Rack 1
Rack 2
Rack 3
Rack 4
Rack N_R

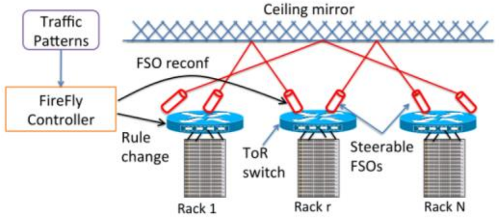
Rotornet


Optical Switching Matrix
k
Cir.
WSS
Coupler
MUX
DEMUX
n
Optical Mux/Demux & Switching Components
ToR1
ToR$_i$
ToR$_j$
λ_1 λ_2 λ_3 λ_4 λ_N
Server Rack
A
B
C
D

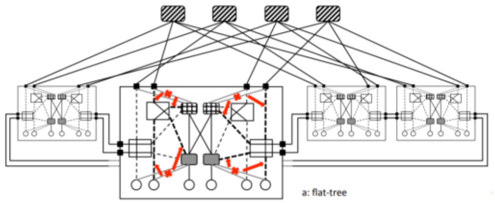
Proteus/OSA



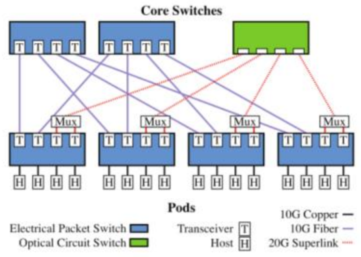
ProjecToR



FireFly



Flat-tree

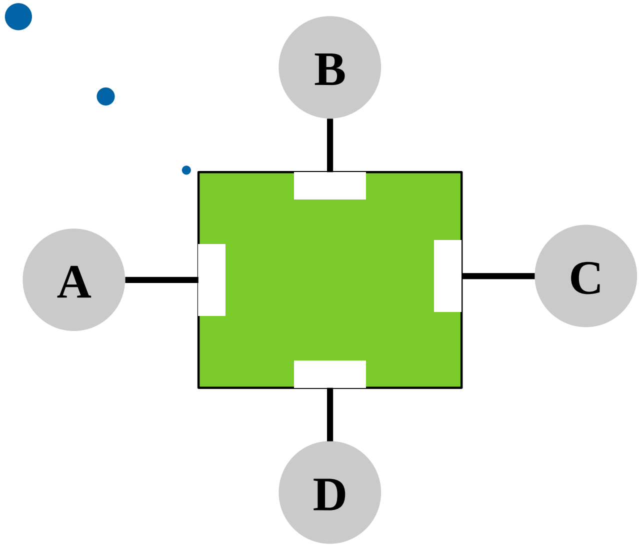


Helios

Reconfigurable Switch

It‘s a Match(ing)!

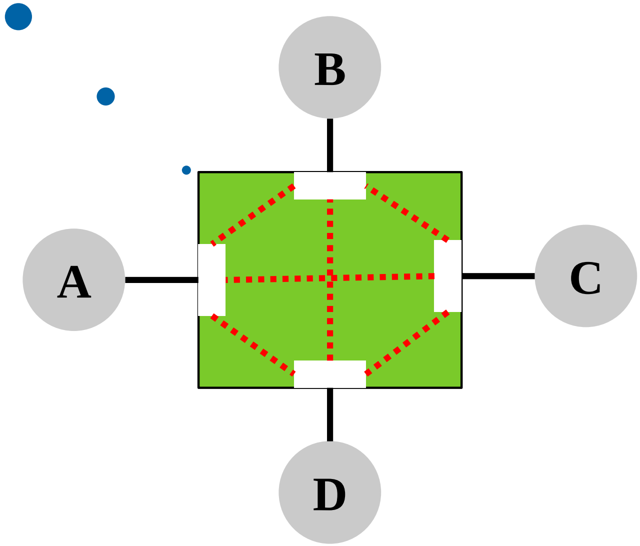
- Idea: Create “physical” connections



Reconfigurable Switch

It‘s a Match(ing)!

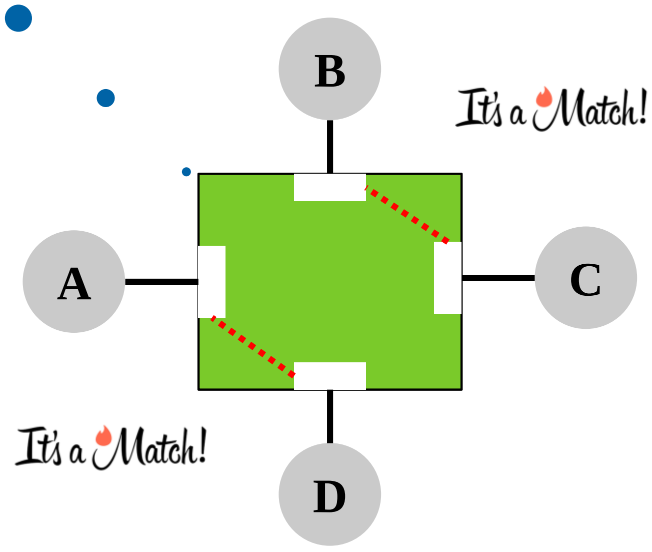
- Idea: Create “physical” connections
  - Difference: Not all-to-all switch
    - E.g. just 1 connection per node



Reconfigurable Switch

It‘s a Match(ing)!

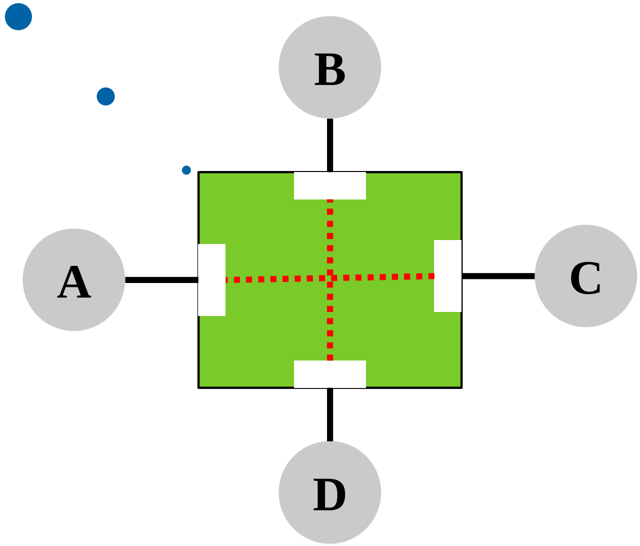
- Idea: Create “physical” connections
  - Difference: Not all-to-all switch
    - E.g. just 1 connection per node

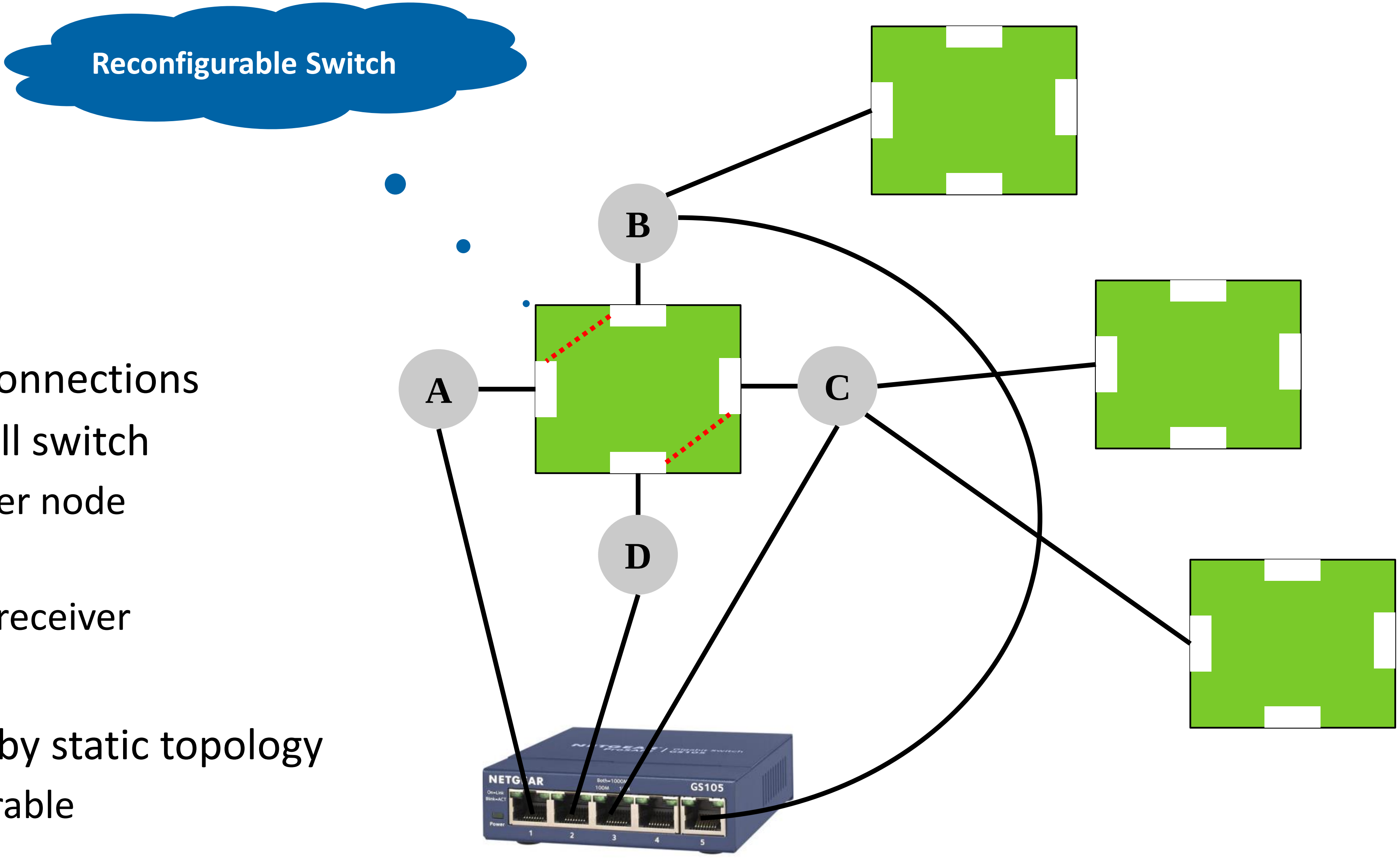


Reconfigurable Switch

It‘s a Match(ing)!

- Idea: Create “physical” connections
  - Difference: Not all-to-all switch
    - E.g. just 1 connection per node





It‘s a Match(ing)!

- Idea: Create “physical” connections
  - Difference: Not all-to-all switch
    - E.g. just 1 connection per node
      - Or many more than 1
      - Or separated sender/receiver

- Basic connectivity often by static topology
  - Hybrid: Static+Reconfigurable

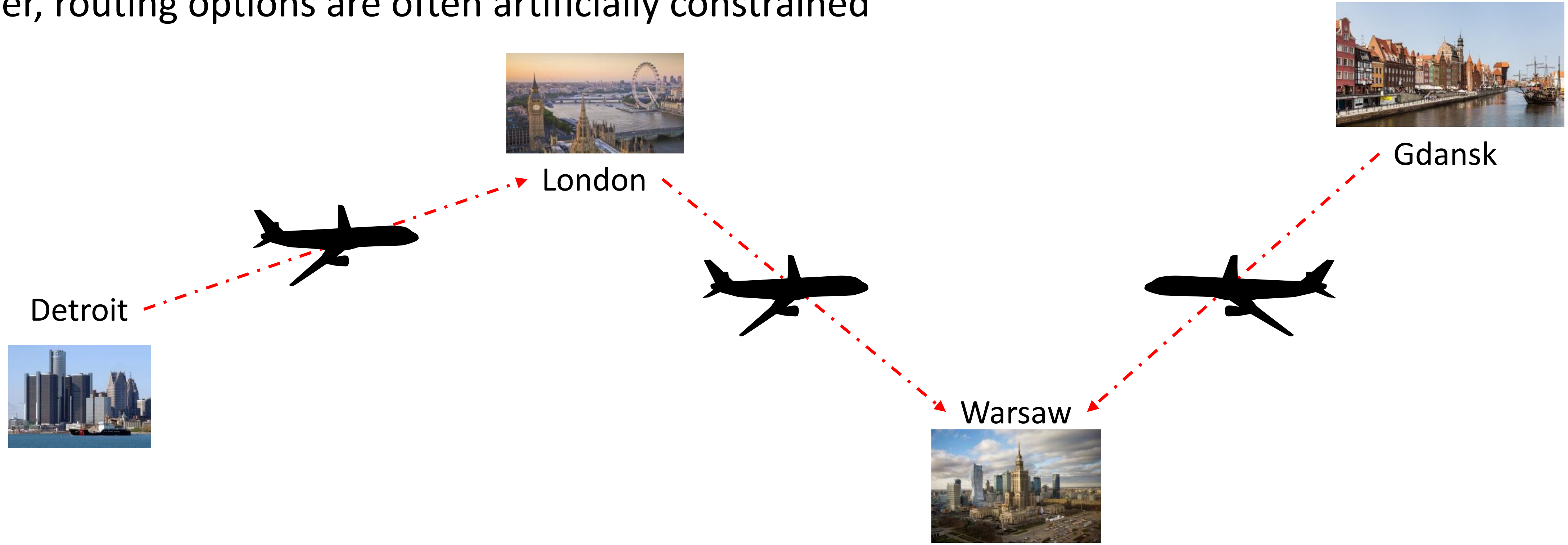
- Reconfigurable switches 1) can be **large/diverse** and 2) the network can contain **many**

Routing Policy Restrictions

- However, routing options are often artificially constrained

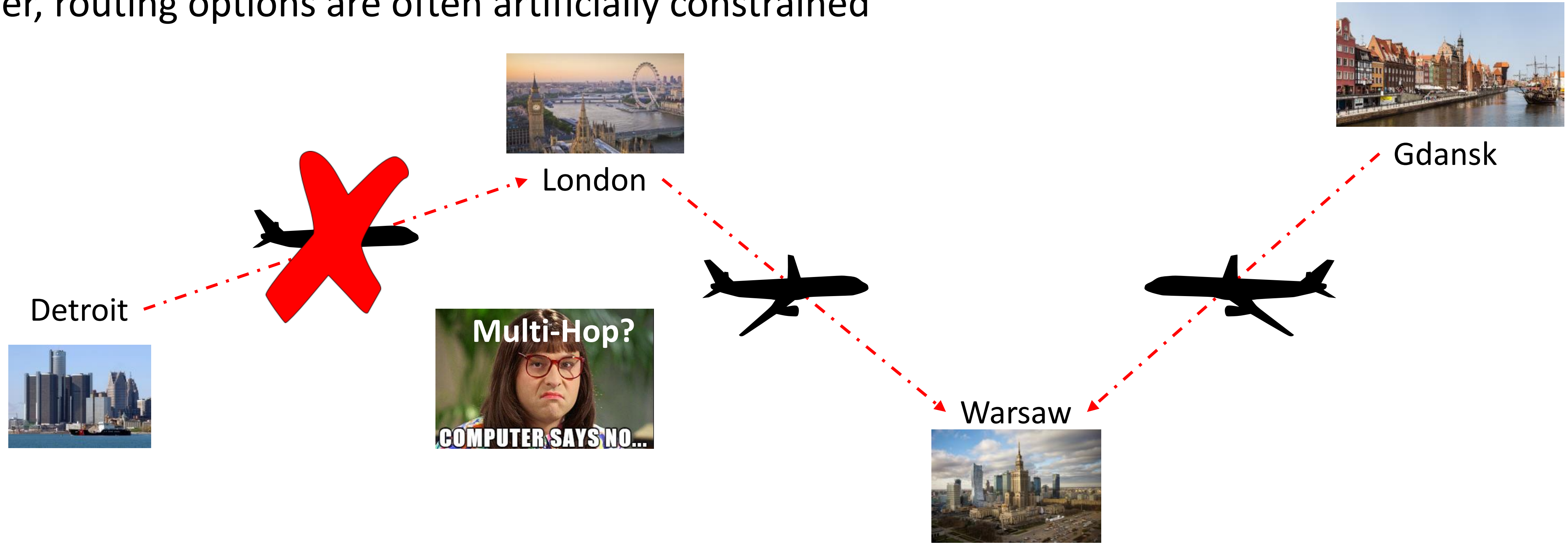
Routing Policy Restrictions

- However, routing options are often artificially constrained



Routing Policy Restrictions

- However, routing options are often artificially constrained



Detroit

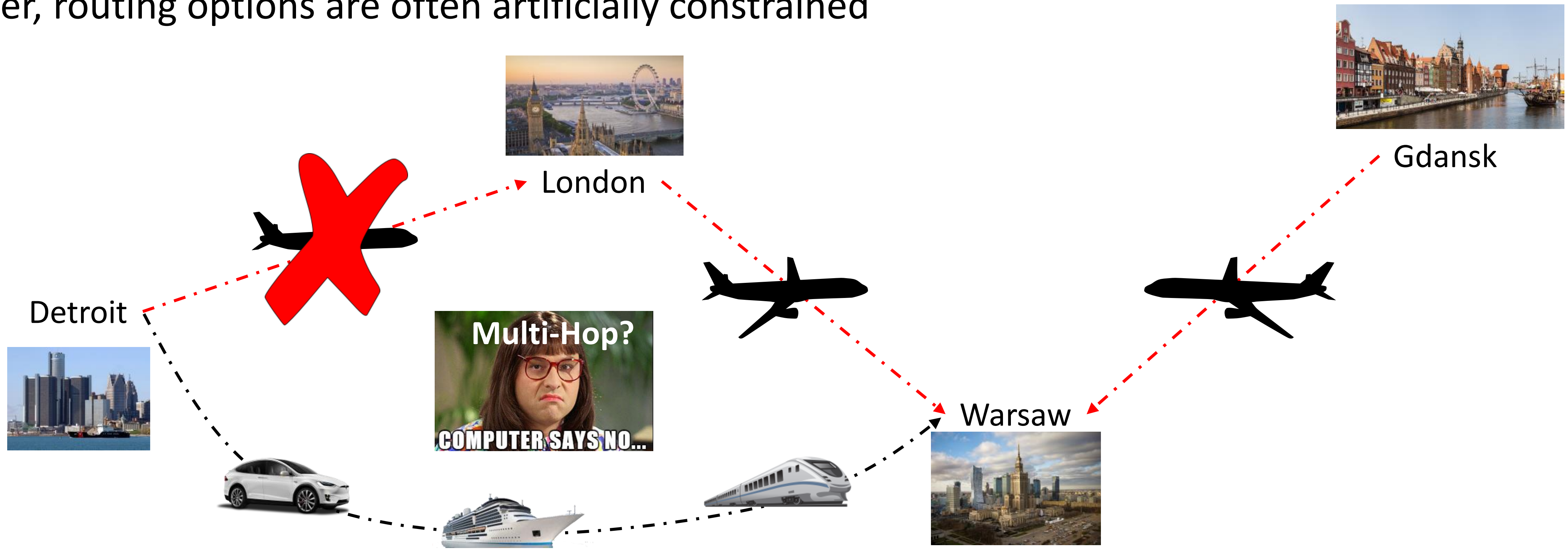
London

Warsaw

Gdansk

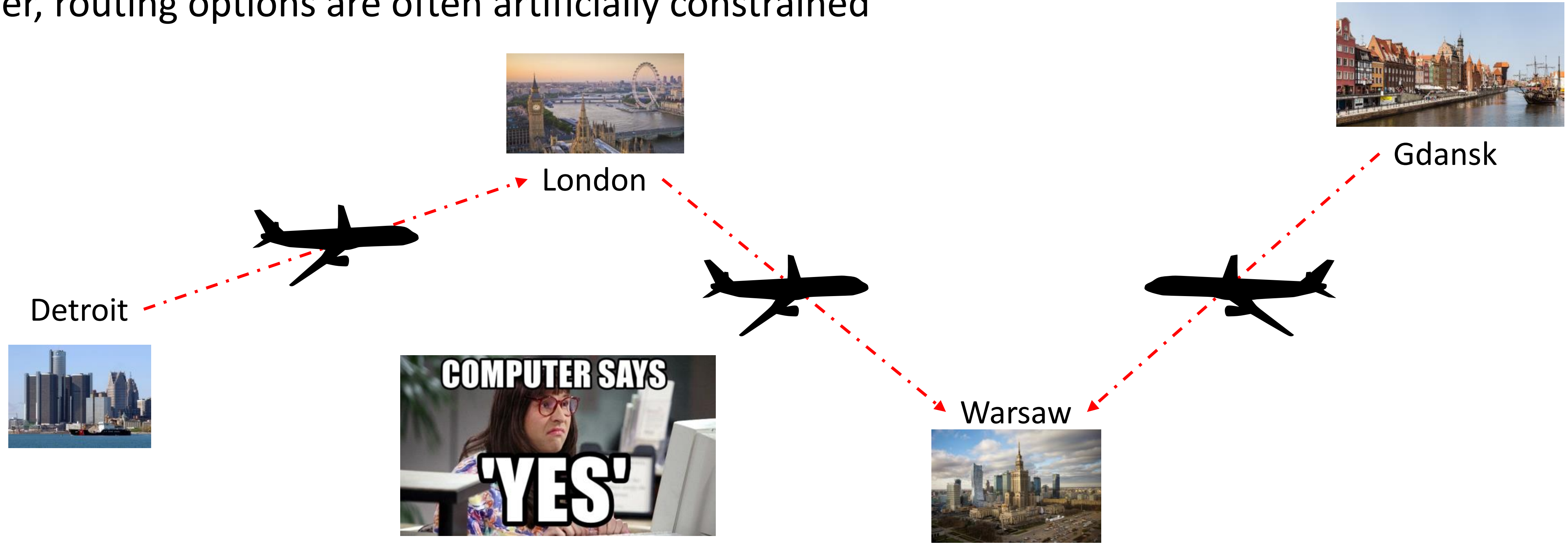
Routing Policy Restrictions

- However, routing options are often artificially constrained



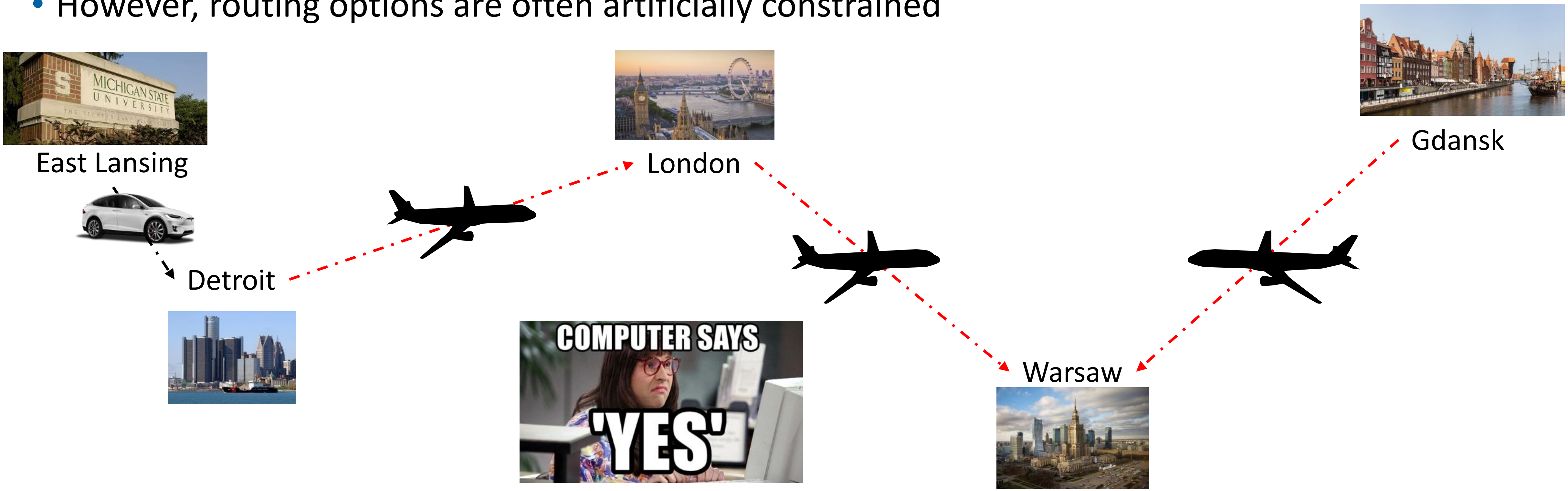
Routing Policy Restrictions

- However, routing options are often artificially constrained



Routing Policy Restrictions

- However, routing options are often artificially constrained



East Lansing

Detroit

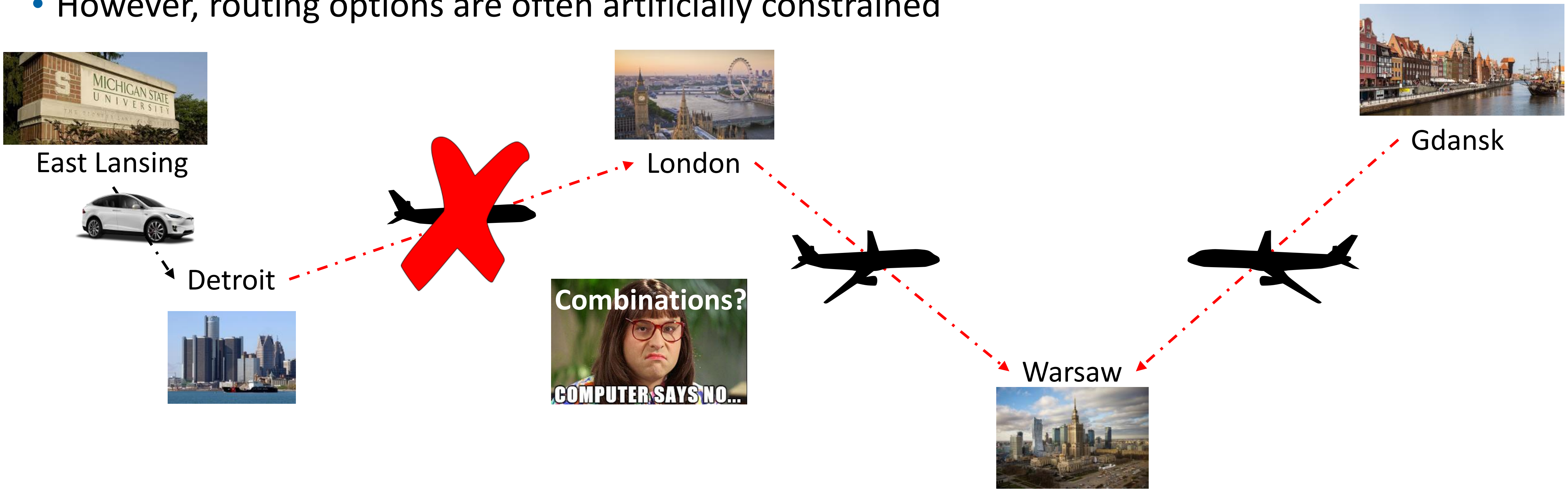
London

Warsaw

Gdansk

Routing Policy Restrictions

- However, routing options are often artificially constrained



East Lansing

Detroit

London

Warsaw

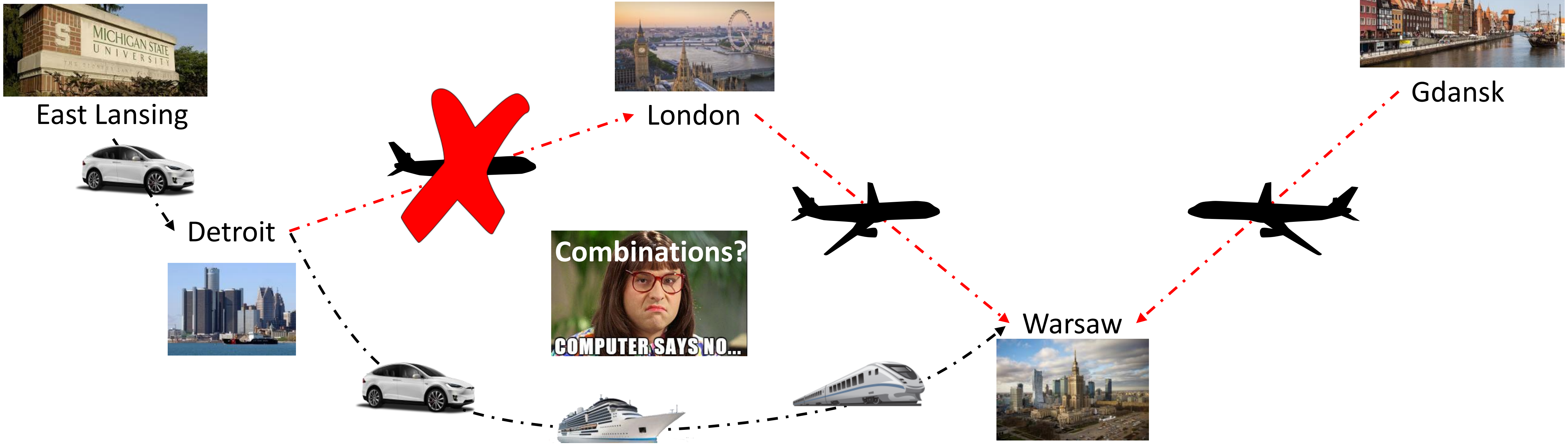
Gdansk

Our goals:

- Multi-hop routing
- Non-segregated
  - Mix static and reconfigurable

Routing Policy Restrictions

- However, routing options are often artificially constrained

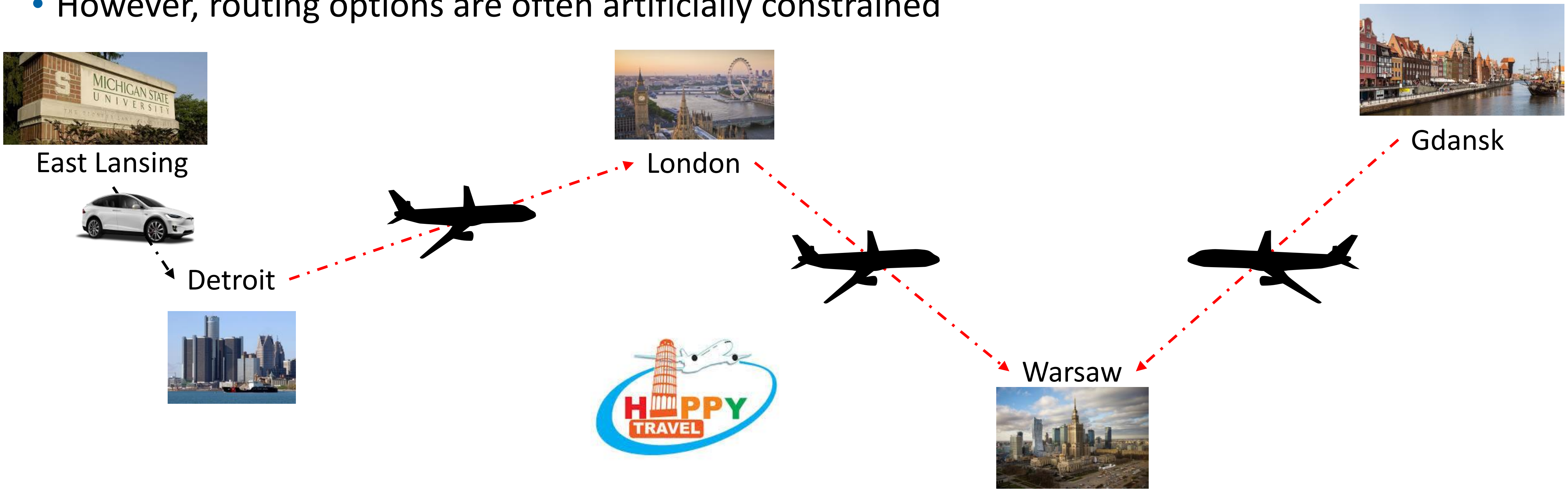


Our goals:

- Multi-hop routing
- Non-segregated
  - Mix static and reconfigurable

Routing Policy Restrictions

- However, routing options are often artificially constrained

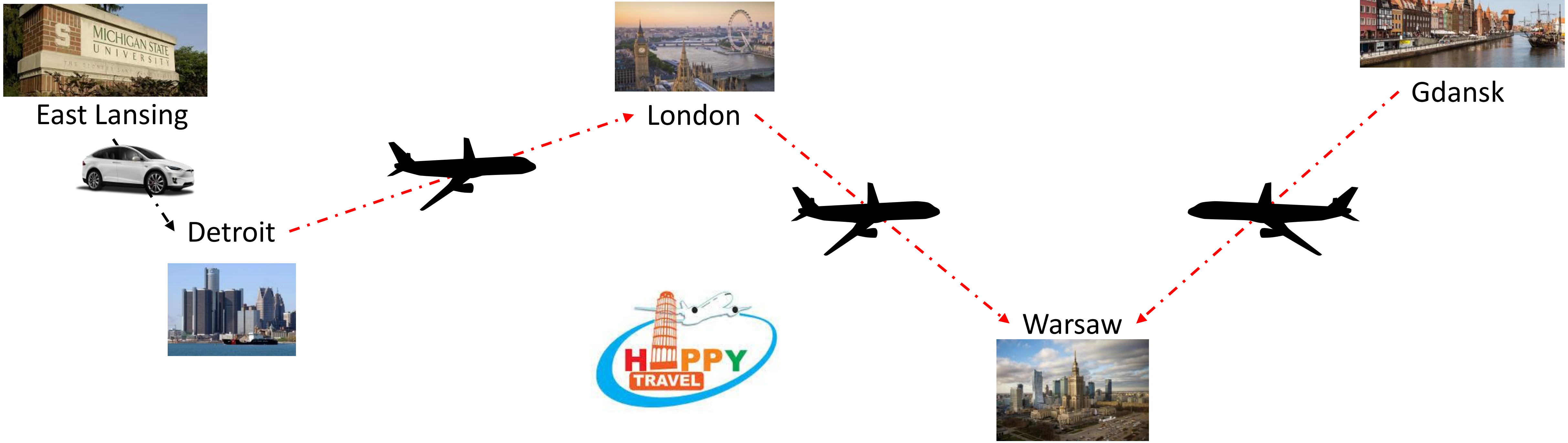


However:
- Currently not well understood ☹

Our goals:
- Multi-hop routing
- Non-segregated
  - Mix static and reconfigurable

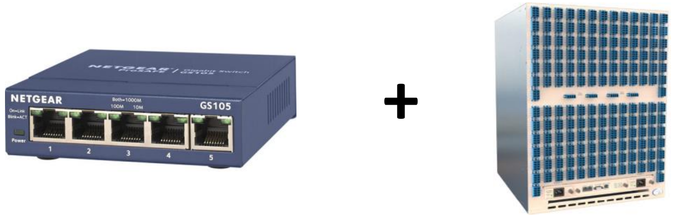
Routing Policy Restrictions

- However, routing options are often artificially constrained



Brief Model and First Overview

- Consider **Hybrid Networks**
  - Static topology + reconfigurable switches



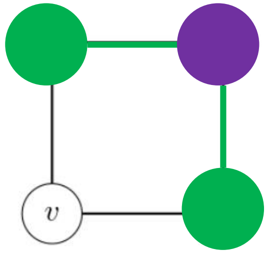
- Objective for given communication pattern:
  - Optimize for **short routes** (sum of weighted path lengths)

- Some first things we can show:
  - Already in simple general settings: **NP-hard to be optimal**
  - For **single-hop** reconfigurable XOR static topology: max. **matching algorithms optimal**
    - (even for a reconfigurable switch permitting *k* connections per node)

NP-hard to approximate better than $\Omega(\log |V|)$ (Feige'98)

Also: NP-Hard to *Approximate*

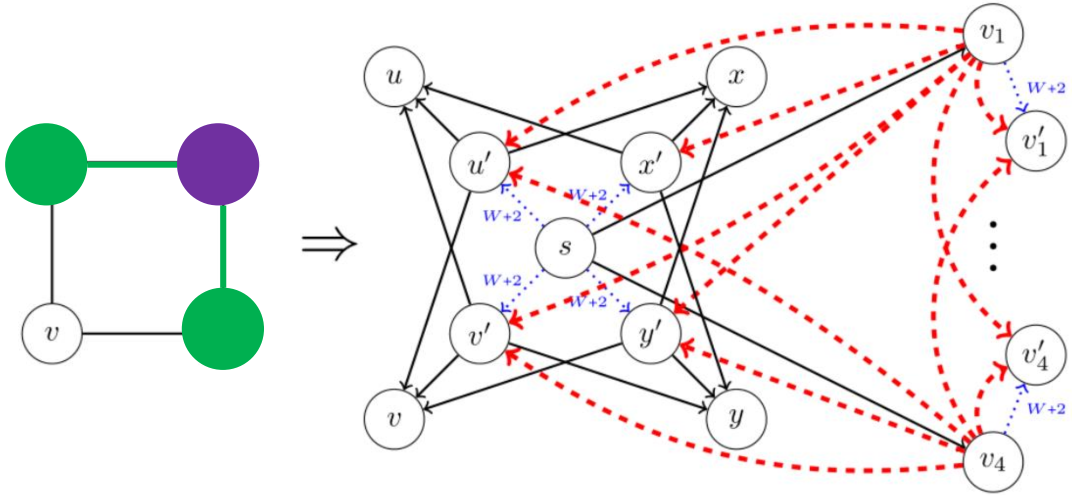
- We perform a reduction from *Dominating Set*
  - Find small node set $D \subseteq V$ s.t. every node is neighbored (*dominated*) by D



NP-hard to approximate better than $\Omega(\log|V|)$ (Feige'98)

Also: NP-Hard to *Approximate*

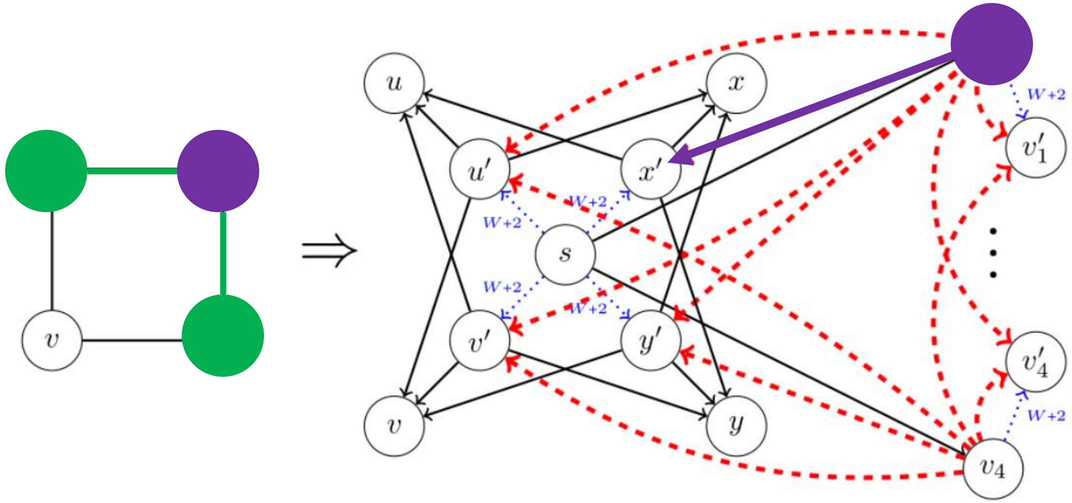
- We perform a reduction from *Dominating Set*
  - Find small node set $D \subseteq V$ s.t. every node is neighbored (*dominated*) by D



NP-hard to approximate better than $\Omega(\log|V|)$ (Feige'98)

Also: NP-Hard to *Approximate*

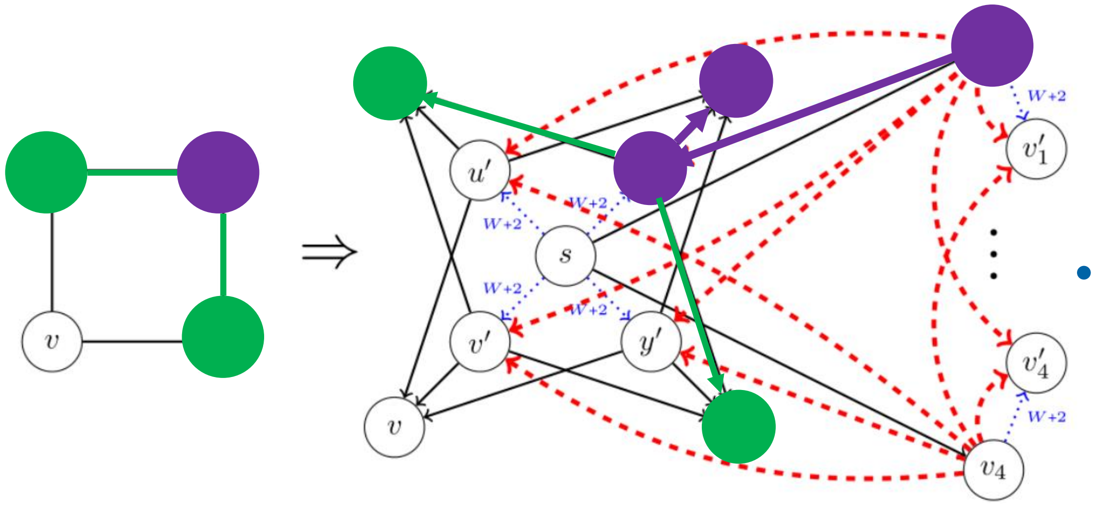
- We perform a reduction from *Dominating Set*
  - Find small node set $D \subseteq V$ s.t. every node is neighbored (*dominated*) by D



NP-hard to approximate better than $\Omega(\log|V|)$ (Feige'98)

Also: NP-Hard to *Approximate*

- We perform a reduction from *Dominating Set*
  - Find small node set $D \subseteq V$ s.t. every node is neighbored (*dominated*) by D



Approximation bounds carry over

And commonly used in many papers

General Reconfigurable Algorithms?

- We know: Segregated single-hop: Matching algorithms are a perfect fit
  - How to extend to non-segregated *paths*?

East Lansing
Detroit
London
Warsaw
Gdansk
HAPPY TRAVEL

- Observation: Shortest path traverses each reconfigurable switch only once*
  - Allows us to extend *Dijkstra*'s algorithm

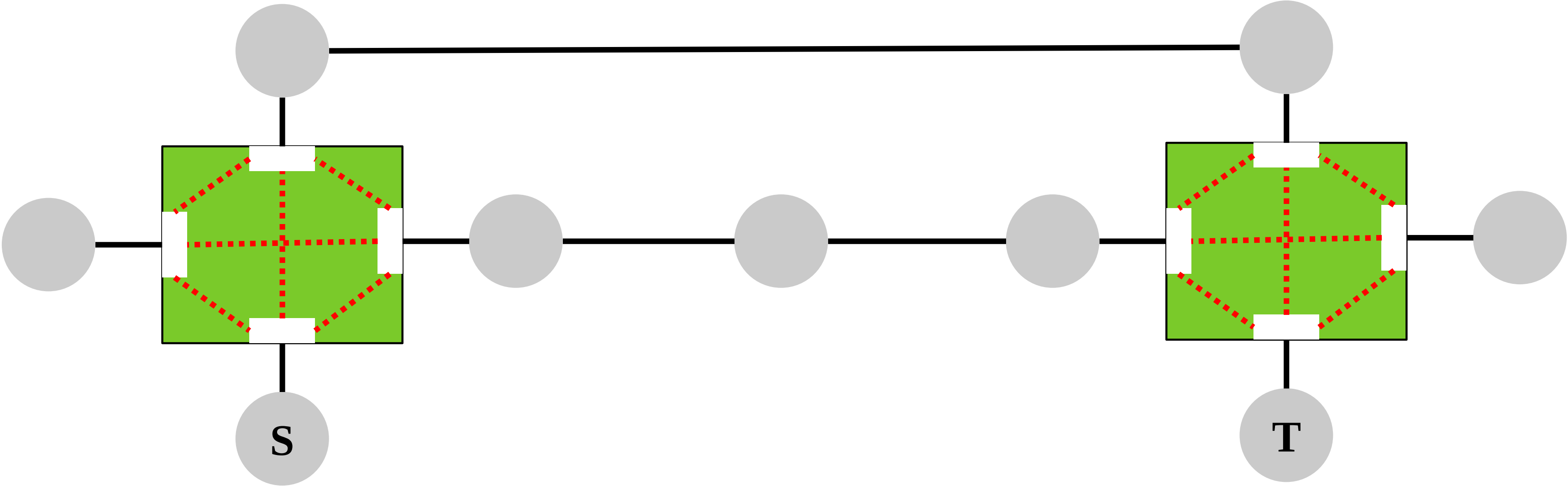
*if triangle-inequality holds inside reconfigurable switches

Reconfigurable Dijkstra (*S*-*T*-Path)

1) Add all still possible reconfigurable links as static links
2) Run standard Dijkstra from source *S*
3) Add newly used links on shortest path to *T* to the matchings

Reconfigurable Dijkstra (*S*-*T*-Path)

1) Add all still possible reconfigurable links as static links
2) Run standard Dijkstra from source *S*
3) Add newly used links on shortest path to *T* to the matchings

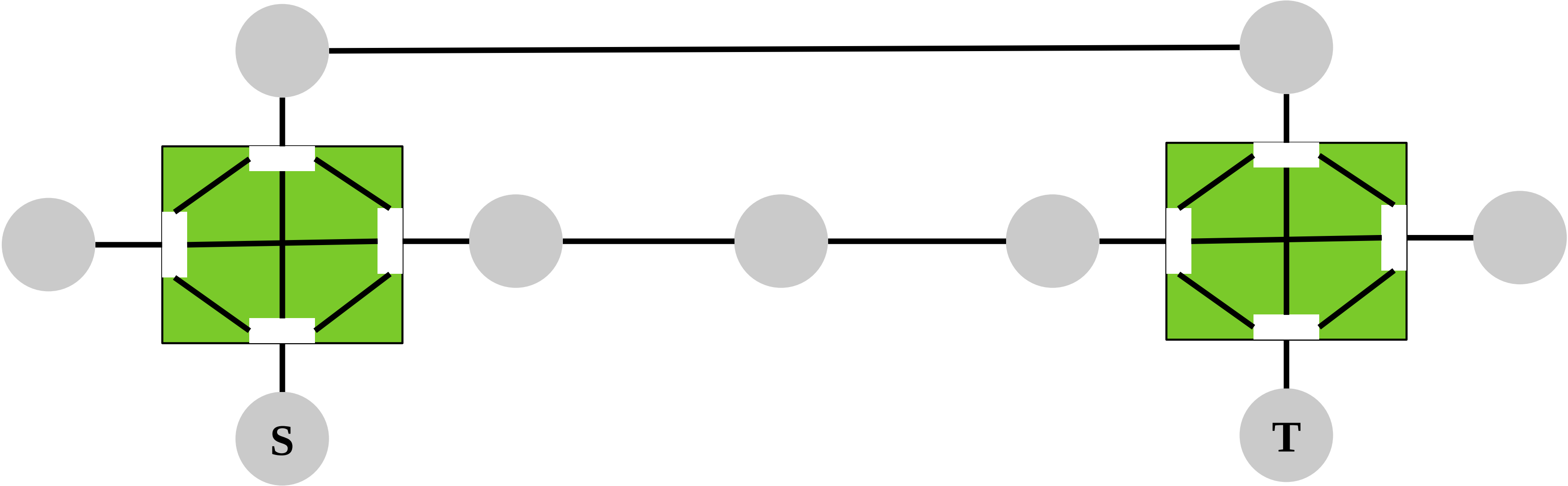


Reconfigurable Dijkstra (*S*-*T*-Path)

1) **Add all still possible reconfigurable links as static links**

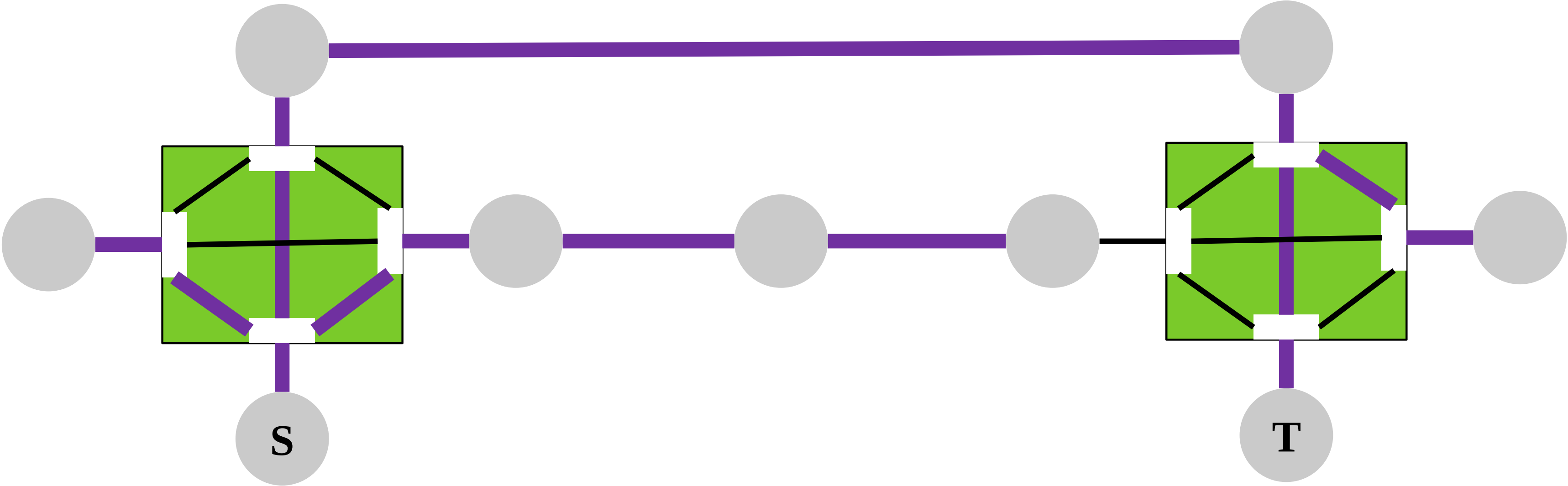
2) Run standard Dijkstra from source *S*

3) Add newly used links on shortest path to *T* to the matchings



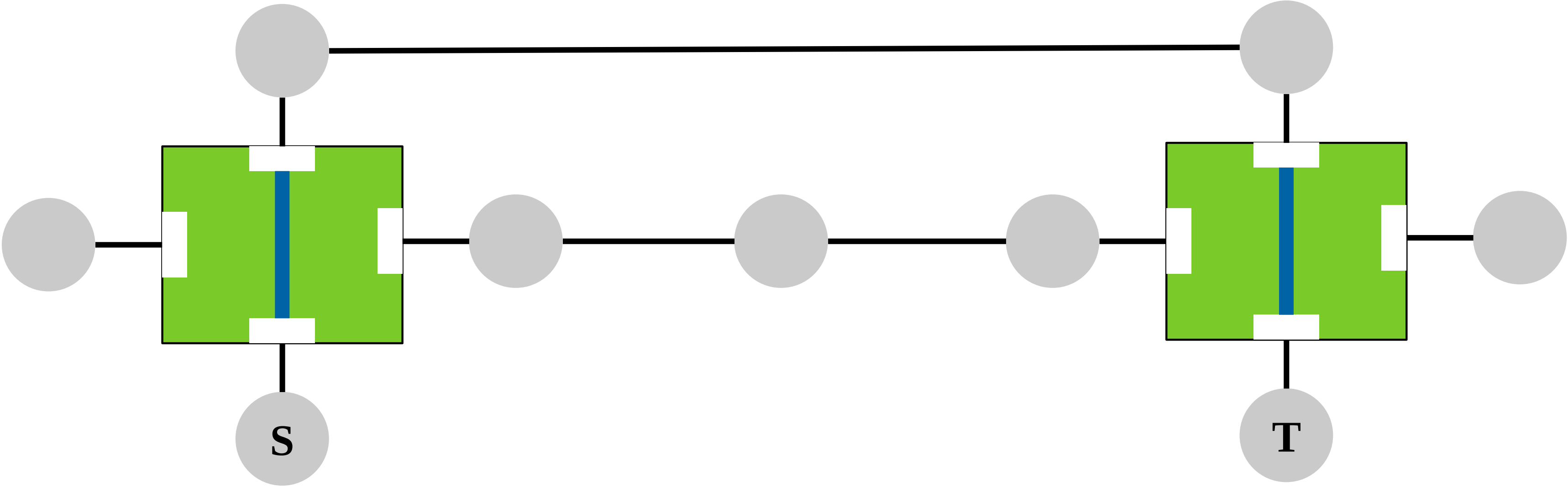
Reconfigurable Dijkstra (*S*-*T*-Path)

1) Add all still possible reconfigurable links as static links
2) **Run standard Dijkstra from source *S***
3) Add newly used links on shortest path to *T* to the matchings



Reconfigurable Dijkstra (*S*-*T*-Path)

1) Add all still possible reconfigurable links as static links
2) Run standard Dijkstra from source *S*
3) **Add newly used links on shortest path to *T* to the matchings**



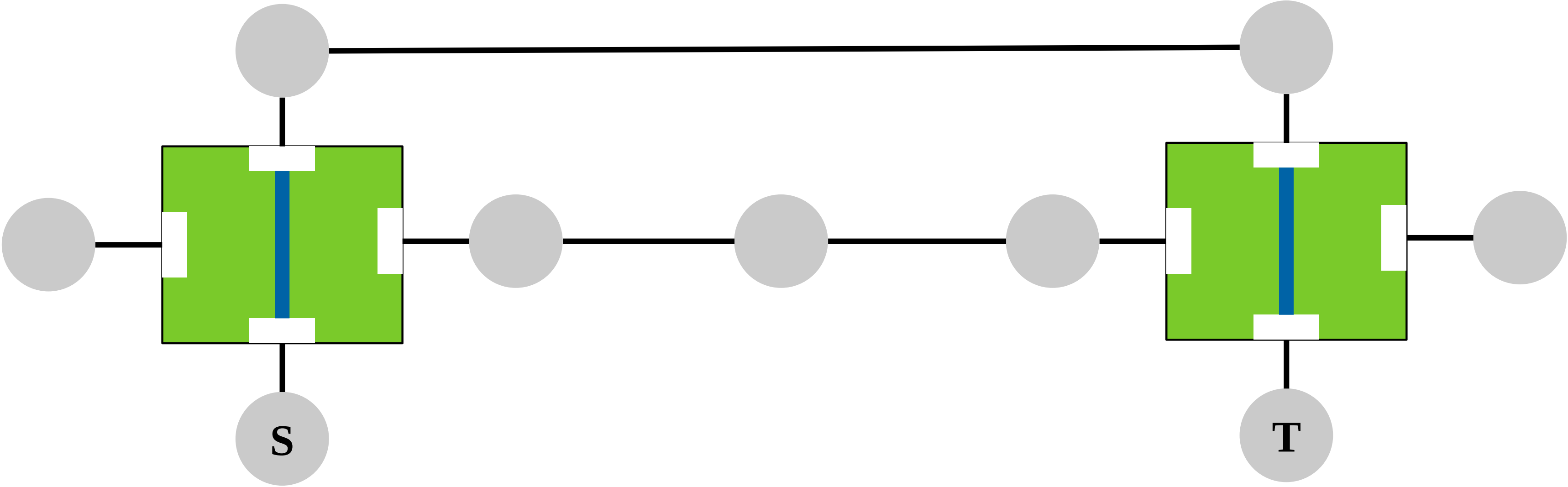
Reconfigurable Dijkstra (*S*-*T*-Path) • • ●

Also works if some matching links already exist

1) Add all still possible reconfigurable links as static links

2) Run standard Dijkstra from source *S*

3) **Add newly used links on shortest path to *T* to the matchings**



Use Reconfigurable Dijkstra (***RD***) as a Building Block to Add Matching Links

DemandFirst

1) Sort demands by size
2) Run ***RD*** *on list*

Evaluate impact of RD on *all* demands?

GainDemand

1) Run ***RD*** for each demand
2) Sort by improvements for all
3) Run ***RD*** *on list*

Why evaluate only *once* at beginning?

GainUpdate

1) Run ***GainDemand***, but re-evaluate after each insertion of links

Why not link-by-link?

GreedyLinks

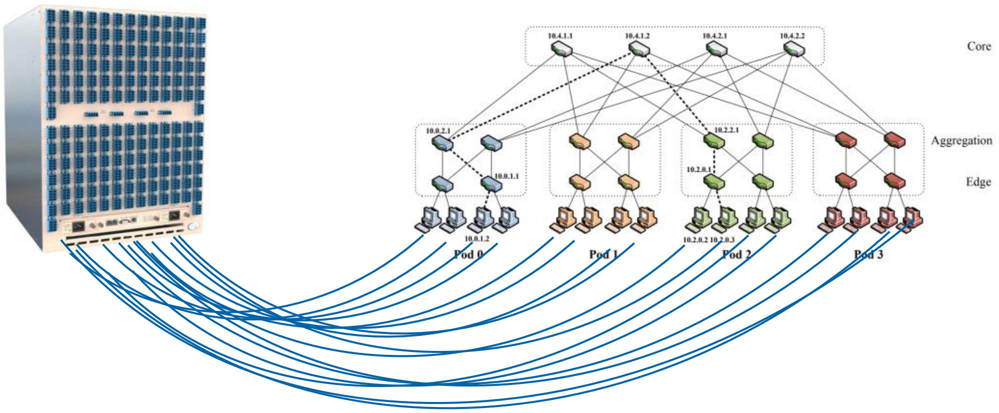
1) Pick link that benefits all demands the most
2) Repeat until no more links possible

Simulations

- Standard topology:
  - Static: Clos/Tree-like (depth 3)
  - Reconfigurable: Connected to all leaves
- Traffic data
  - From recent facebook data set
  - Aggregated to different #nodes/times
- Algorithms:
  - State of the art: Maximum Matching, just static
  - Our: Demand First, GainDemand/Update, GreedyLinks
  - Also: Optimal ILP (small #servers)

From: calient.net

From: Al-Fares et al. 2008



Formulation in paper

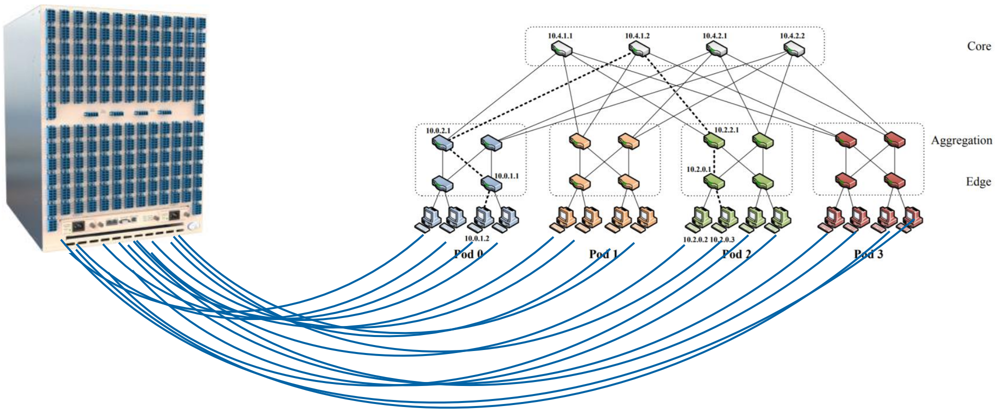
Simulations

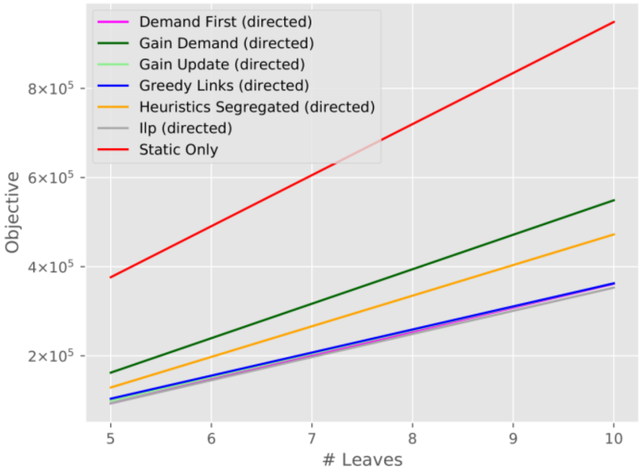
- Standard topology:
  - Static: Clos/Tree-like (depth 3)
  - Reconfigurable: Connected to all leaves
- Traffic data
  - From recent facebook data set
  - Aggregated to different #nodes/times
- Algorithms:
  - State of the art: Maximum Matching, just static
  - Our: Demand First, GainDemand/Update, GreedyLinks
  - Also: Optimal ILP (small #servers)

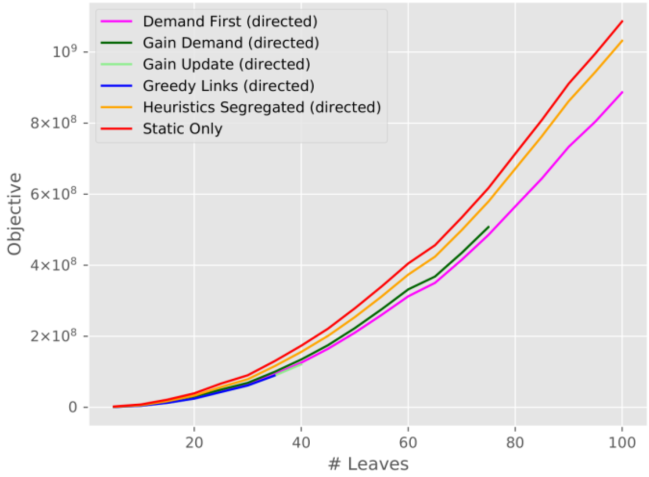
Formulation in paper

From: calient.net

From: Al-Fares et al. 2008



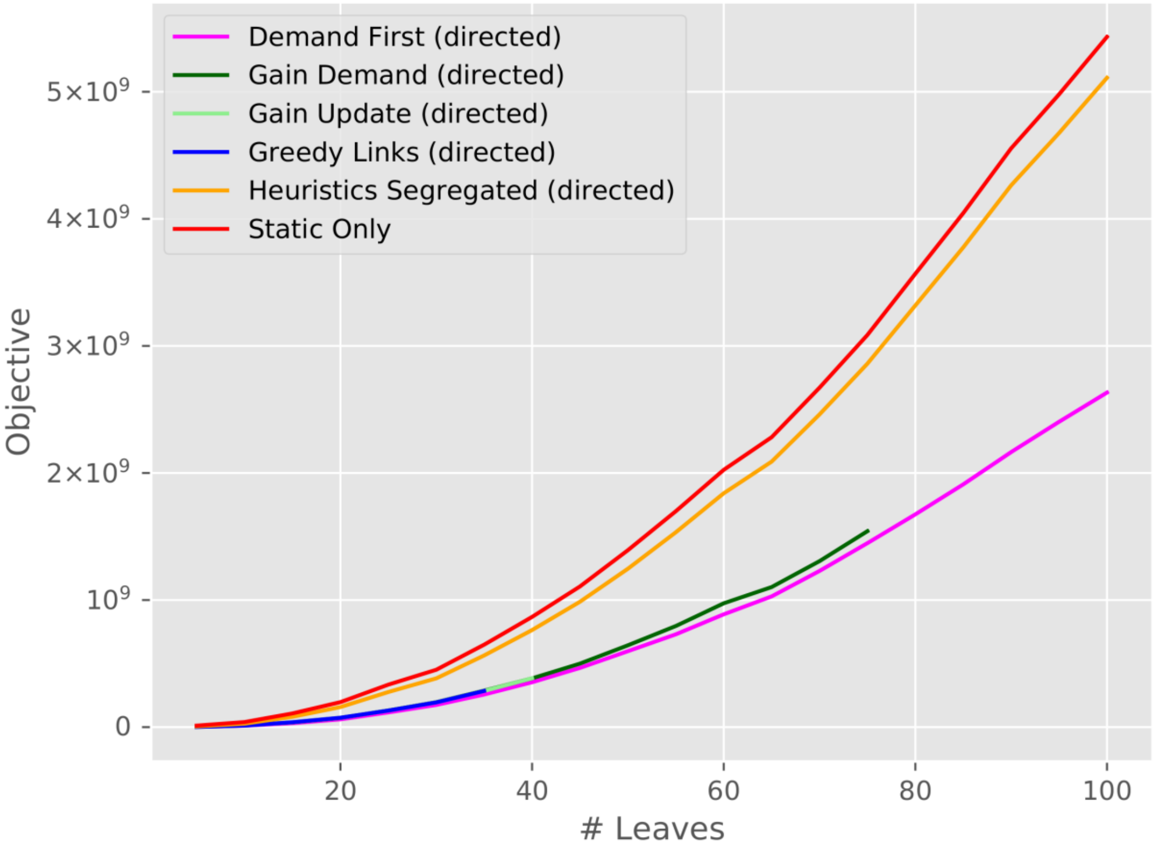


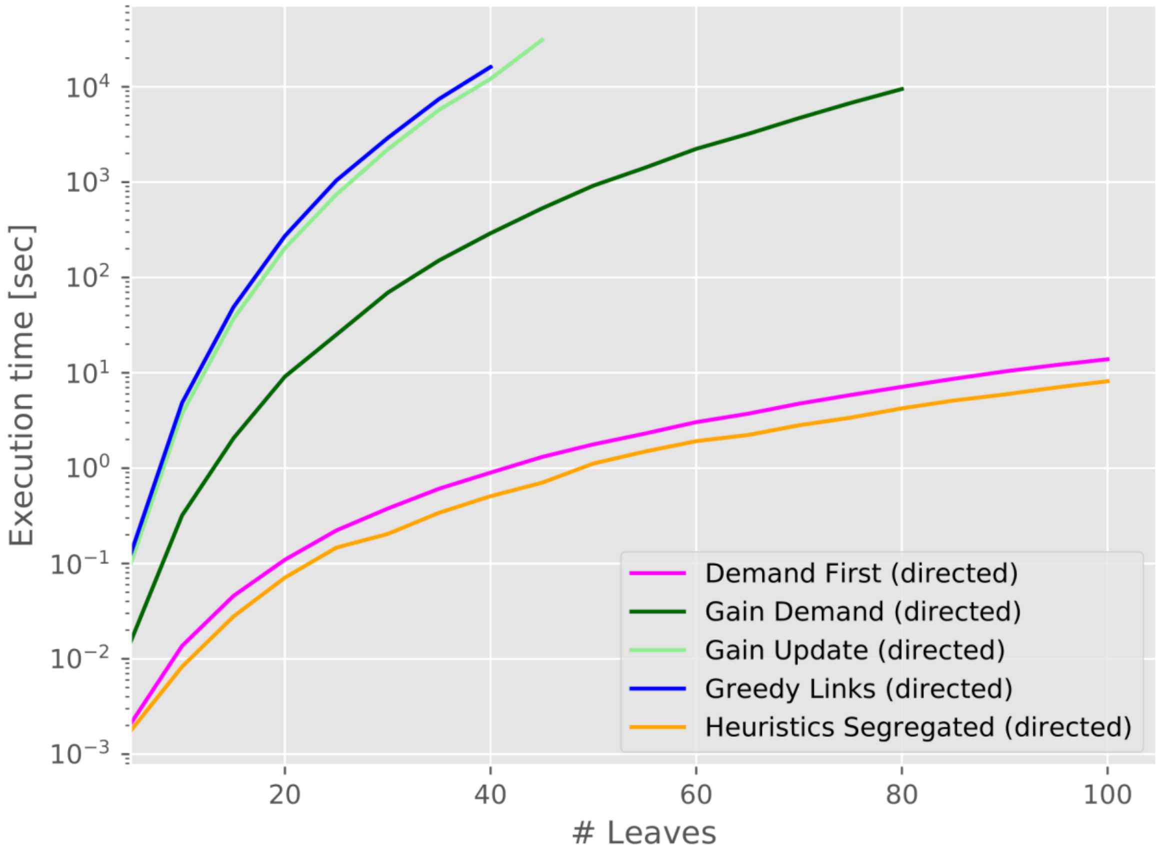


weight ratio: 1:1, time window: 10

Want to compare your own ideas?
Our simulator is publicly available ☺

Performance and Runtime

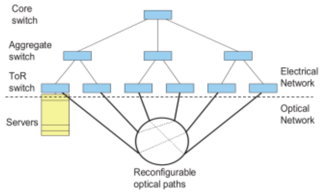




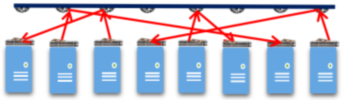
weight ratio: 1:5, time window: 100

Summary

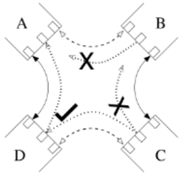
- We studied **reconfigurable data centers** w.r.t. short routes



c-Through

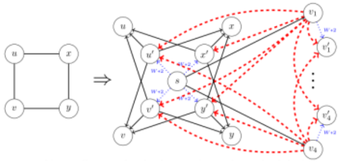


ProjecToR

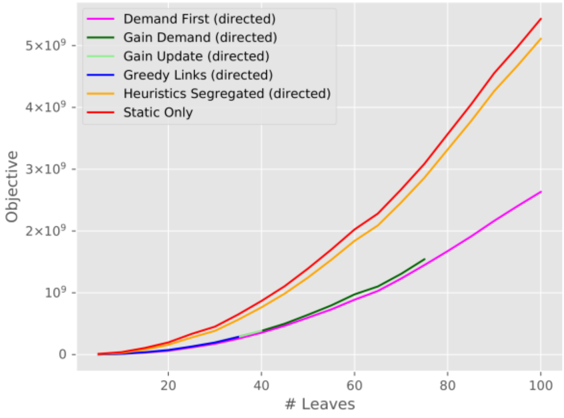


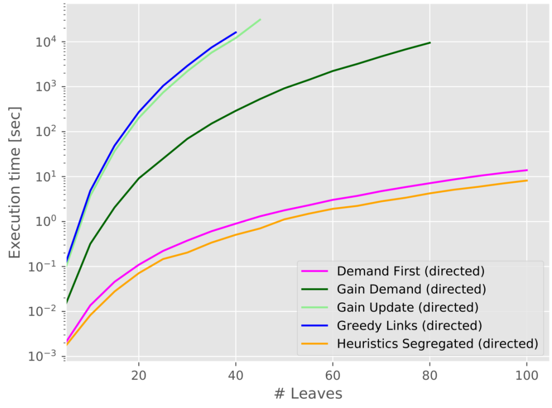
Proteus

- **NP-hard** to approximate well…. ☹

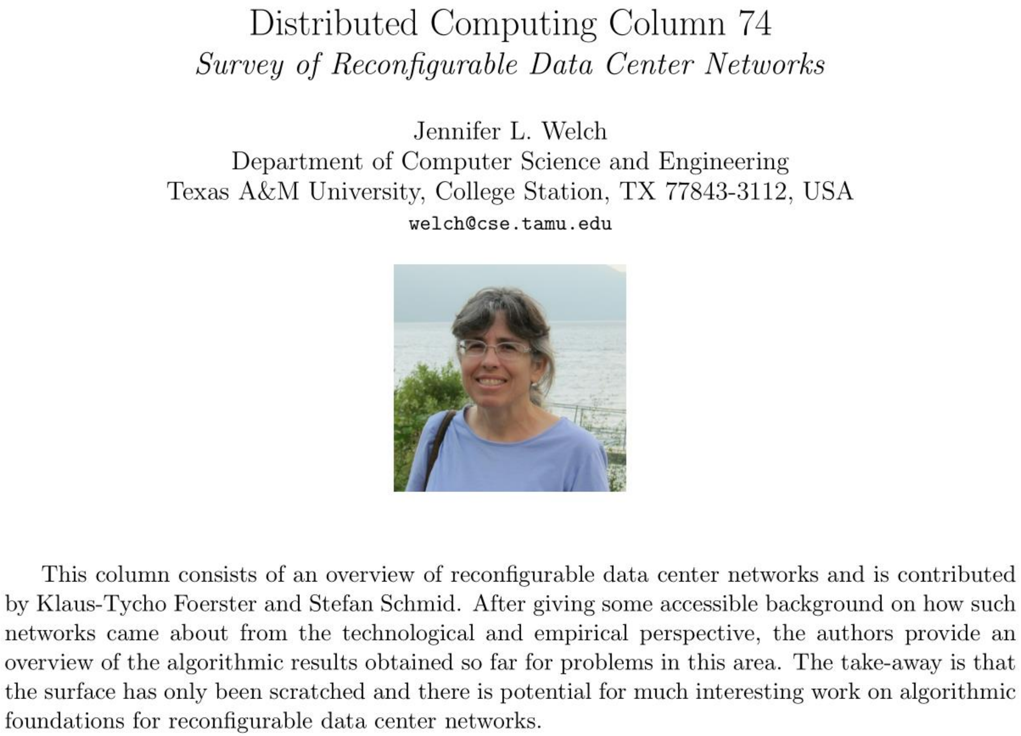


- But: Our algorithms are efficient in practice ☺
  - **Improve** the performance of the **state-of-the art**
  - Roughly **similar runtimes**
  - **Not restricted** to specific technologies





More Background: Next SIGACT News

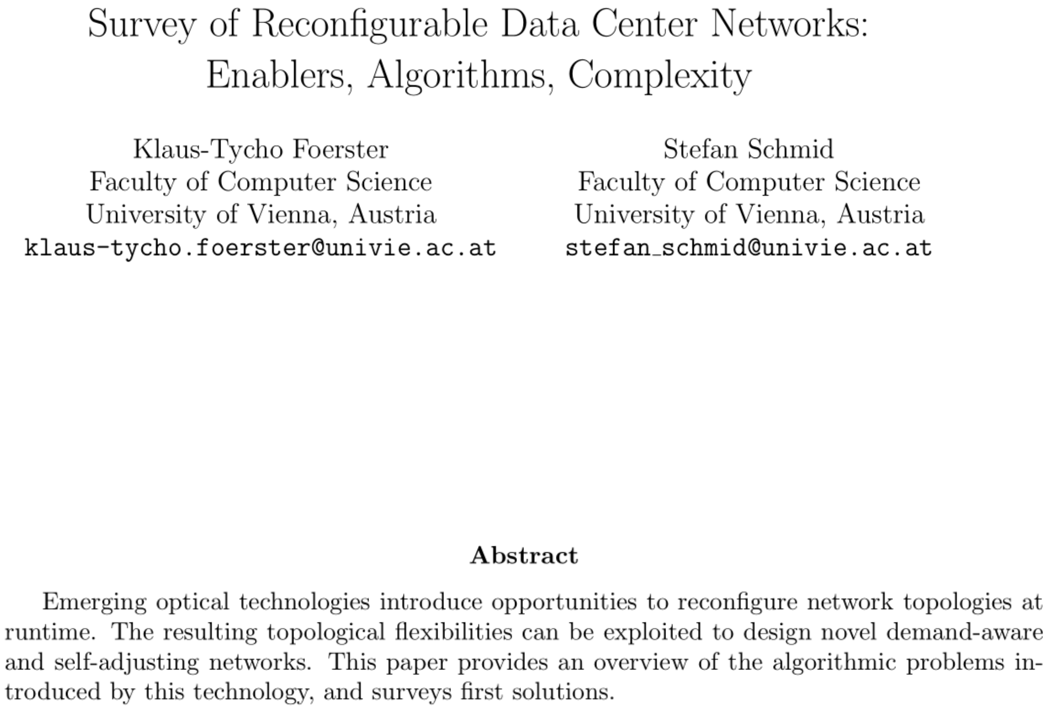


Distributed Computing Column 74

Survey of Reconfigurable Data Center Networks

Jennifer L. Welch
Department of Computer Science and Engineering
Texas A&M University, College Station, TX 77843-3112, USA
welch@cse.tamu.edu

This column consists of an overview of reconfigurable data center networks and is contributed by Klaus-Tycho Foerster and Stefan Schmid. After giving some accessible background on how such networks came about from the technological and empirical perspective, the authors provide an overview of the algorithmic results obtained so far for problems in this area. The take-away is that the surface has only been scratched and there is potential for much interesting work on algorithmic foundations for reconfigurable data center networks.



Survey of Reconfigurable Data Center Networks: Enablers, Algorithms, Complexity

Klaus-Tycho Foerster
Faculty of Computer Science
University of Vienna, Austria
klaus-tycho.foerster@univie.ac.at

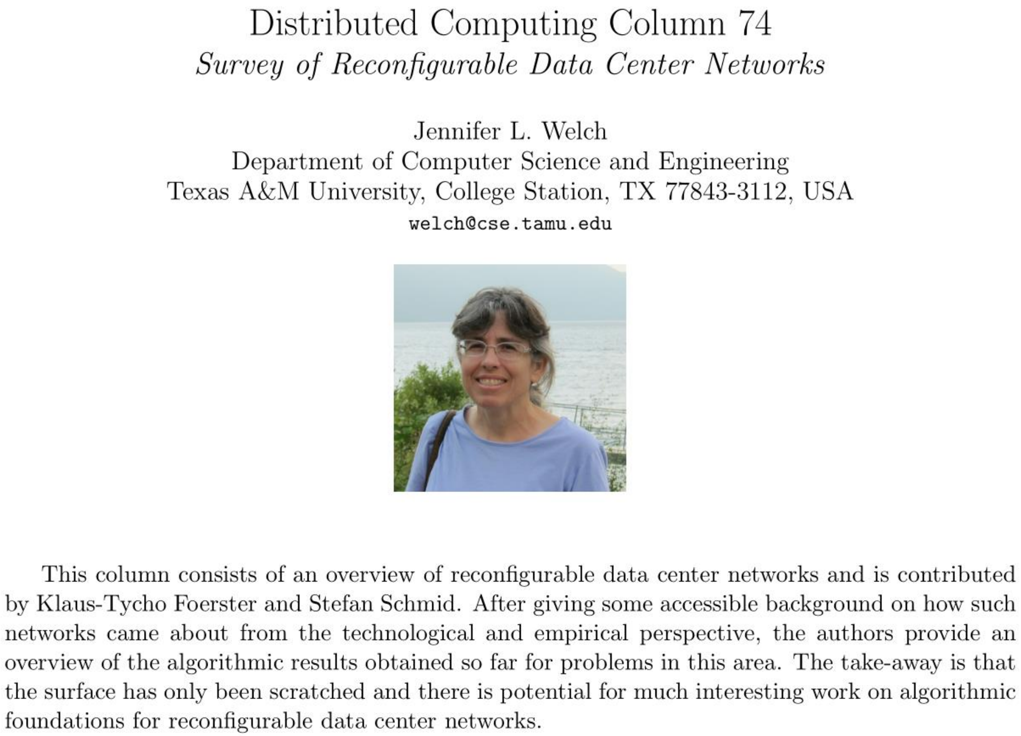
Stefan Schmid
Faculty of Computer Science
University of Vienna, Austria
stefan_schmid@univie.ac.at

Abstract

Emerging optical technologies introduce opportunities to reconfigure network topologies at runtime. The resulting topological flexibilities can be exploited to design novel demand-aware and self-adjusting networks. This paper provides an overview of the algorithmic problems introduced by this technology, and surveys first solutions.

A preprint of our survey is available at: foerster.me/survey19.pdf
The talk slides are available at: foerster.me/ifip19.pdf
Our source code is publicly available (see the paper)

More Background: Next SIGACT News

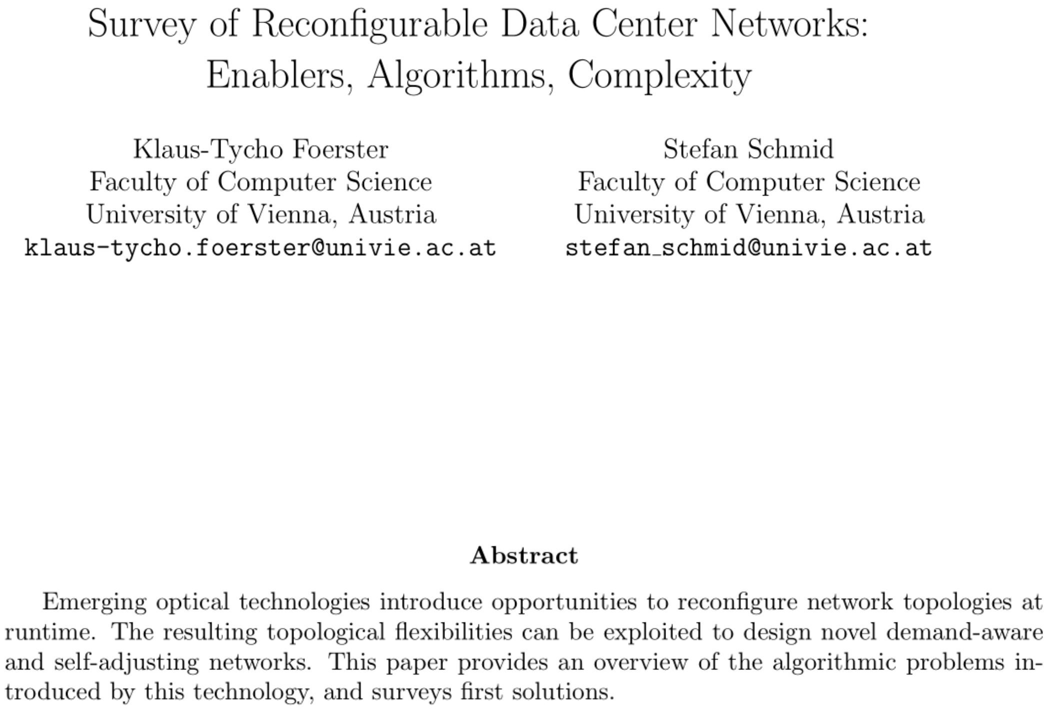


Distributed Computing Column 74

Survey of Reconfigurable Data Center Networks

Jennifer L. Welch
Department of Computer Science and Engineering
Texas A&M University, College Station, TX 77843-3112, USA
welch@cse.tamu.edu

This column consists of an overview of reconfigurable data center networks and is contributed by Klaus-Tycho Foerster and Stefan Schmid. After giving some accessible background on how such networks came about from the technological and empirical perspective, the authors provide an overview of the algorithmic results obtained so far for problems in this area. The take-away is that the surface has only been scratched and there is potential for much interesting work on algorithmic foundations for reconfigurable data center networks.



Survey of Reconfigurable Data Center Networks: Enablers, Algorithms, Complexity

Klaus-Tycho Foerster
Faculty of Computer Science
University of Vienna, Austria
klaus-tycho.foerster@univie.ac.at

Stefan Schmid
Faculty of Computer Science
University of Vienna, Austria
stefan_schmid@univie.ac.at

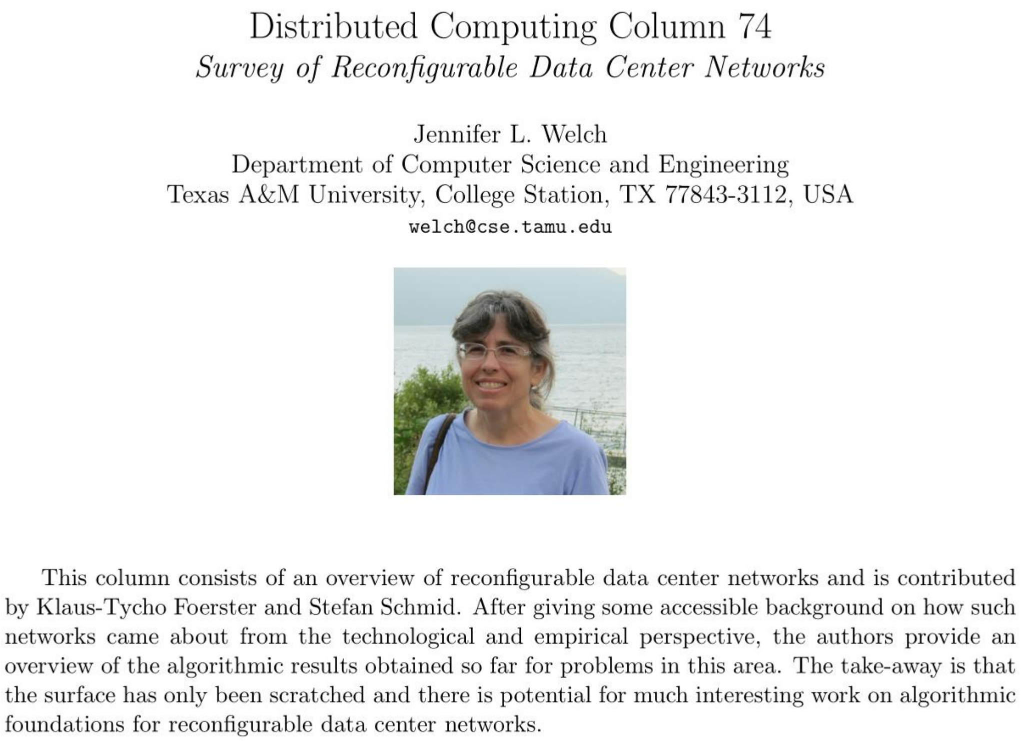
Abstract

Emerging optical technologies introduce opportunities to reconfigure network topologies at runtime. The resulting topological flexibilities can be exploited to design novel demand-aware and self-adjusting networks. This paper provides an overview of the algorithmic problems introduced by this technology, and surveys first solutions.

Thank you! ☺

A preprint of our survey is available at: foerster.me/survey19.pdf
The talk slides are available at: foerster.me/ifip19.pdf
Our source code is publicly available (see the paper)

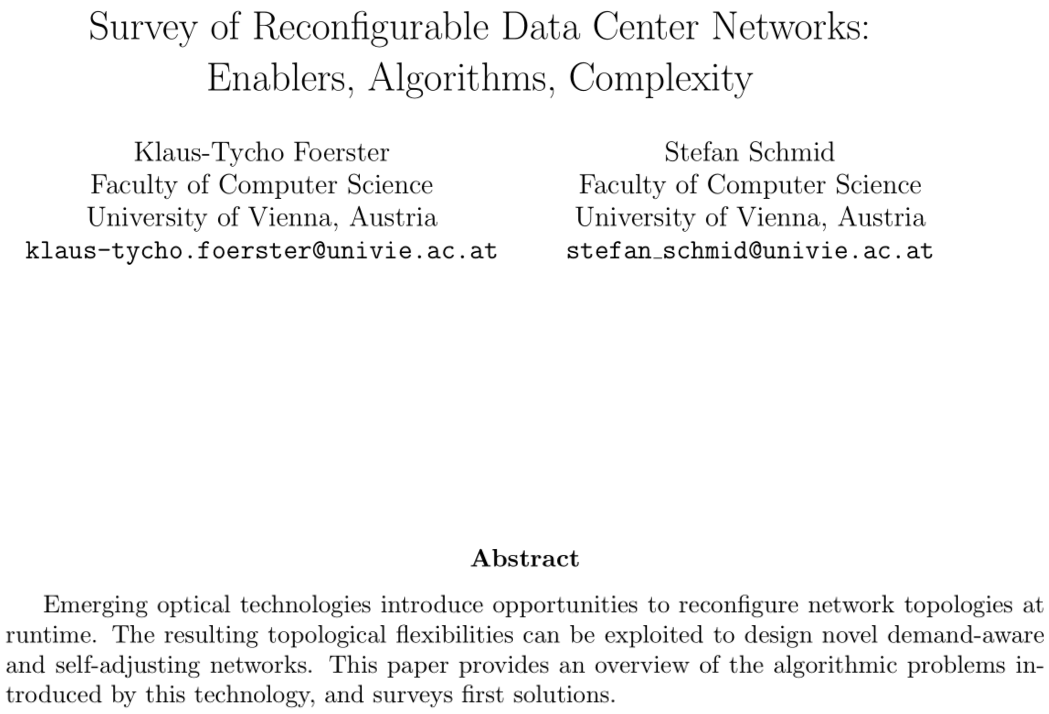
More Background: Next SIGACT News



Distributed Computing Column 74
Survey of Reconfigurable Data Center Networks

Jennifer L. Welch
Department of Computer Science and Engineering
Texas A&M University, College Station, TX 77843-3112, USA
welch@cse.tamu.edu

This column consists of an overview of reconfigurable data center networks and is contributed by Klaus-Tycho Foerster and Stefan Schmid. After giving some accessible background on how such networks came about from the technological and empirical perspective, the authors provide an overview of the algorithmic results obtained so far for problems in this area. The take-away is that the surface has only been scratched and there is potential for much interesting work on algorithmic foundations for reconfigurable data center networks.



Survey of Reconfigurable Data Center Networks:
Enablers, Algorithms, Complexity

Klaus-Tycho Foerster
Faculty of Computer Science
University of Vienna, Austria
klaus-tycho.foerster@univie.ac.at

Stefan Schmid
Faculty of Computer Science
University of Vienna, Austria
stefan_schmid@univie.ac.at

Abstract

Emerging optical technologies introduce opportunities to reconfigure network topologies at runtime. The resulting topological flexibilities can be exploited to design novel demand-aware and self-adjusting networks. This paper provides an overview of the algorithmic problems introduced by this technology, and surveys first solutions.

universität wien

Thank you! ☺

A preprint of the survey is available at: foerster.me/survey19.pdf
The talk slides are available at: foerster.me/ifip19.pdf
Our source code is publicly available (see the paper)

Efficient Non-Segregated Routing for Reconfigurable Demand-Aware Networks

Thomas Fenz, Klaus-Tycho Foerster, Stefan Schmid, and Anaïs Villedieu

References

- K.-T. Foerster, M. Ghobadi, and S. Schmid, “Characterizing the algorithmic complexity of reconfigurable data center architectures,” in ANCS. IEEE/ACM, 2018.
- K.-T. Foerster, M. Pacut, and S. Schmid, “On the complexity of non-segregated routing in reconfigurable data center architectures,” ACM SIGCOMM Computer Communication Review (CCR), 2019.
- J. H. Zeng, “Data sharing on traffic pattern inside facebook’s data-center network,” https://research.fb.com/data-sharing-on-traffic-pattern-inside-facebooks-datacenter-network/, Jan. 2017.
- facebook, “Facebook network analytics data sharing,”https://www.facebook.com/groups/1144031739005495/, 2018.
- Y. Xia, X. S. Sun, S. Dzinamarira, D. Wu, X. S. Huang, and T. S. E. Ng, “A tale of two topologies: Exploring convertible data center network architectures with flat-tree,” in SIGCOMM. ACM, 2017
- M. Al-Fares, A. Loukissas, and A. Vahdat, “A scalable, commodity data center network architecture,” in SIGCOMM. ACM, 2008.
- M. Ghobadi, R. Mahajan, A. Phanishayee, N. R. Devanur, J. Kulkarni, G. Ranade, P. Blanche, H. Rastegarfar, M. Glick, and D. C. Kilper, “Projector: Agile reconfigurable data center interconnect,” in SIGCOMM. ACM, 2016.
- N. Farrington, G. Porter, S. Radhakrishnan, H. H. Bazzaz, V. Subramanya, Y. Fainman, G. Papen, and A. Vahdat, “Helios: a hybrid electrical/optical switch architecture for modular data centers,” in SIGCOMM. ACM, 2010.
- N. H. Azimi, Z. A. Qazi, H. Gupta, V. Sekar, S. R. Das, J. P. Longtin, H. Shah, and A. Tanwer, “Firefly: a reconfigurable wireless data center fabric using free-space optics,” in SIGCOMM. ACM, 2014
- E. W. Dijkstra, “A note on two problems in connexion with graphs,” Numerische Mathematik, vol. 1, no. 1, pp. 269–271, Dec 1959.
- K. Chen et al., “OSA: an optical switching architecture for data center networks with unprecedented flexibility,” IEEE/ACM Trans.Netw., vol. 22, no. 2, pp. 498–511, 2014.
- W. M. Mellette, R. McGuinness, A. Roy, A. Forencich, G. Papen, A. C. Snoeren, and G. Porter, “Rotornet: A scalable, low-complexity, opticaldatacenter network,” in SIGCOMM. ACM, 2017.