UNIVERSITEIT GENT

FACULTY OF ENGINEERING AND ARCHITECTURE

Dimensioning (optical) networks for cloud computing

Chris Develder, *et al.*

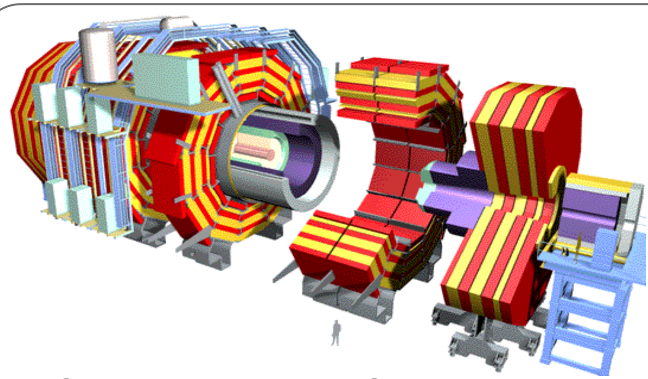
Ghent University – iMinds
Dept. of Information Technology – IBCN

iMinds FUTURE INTERNET DEPT.

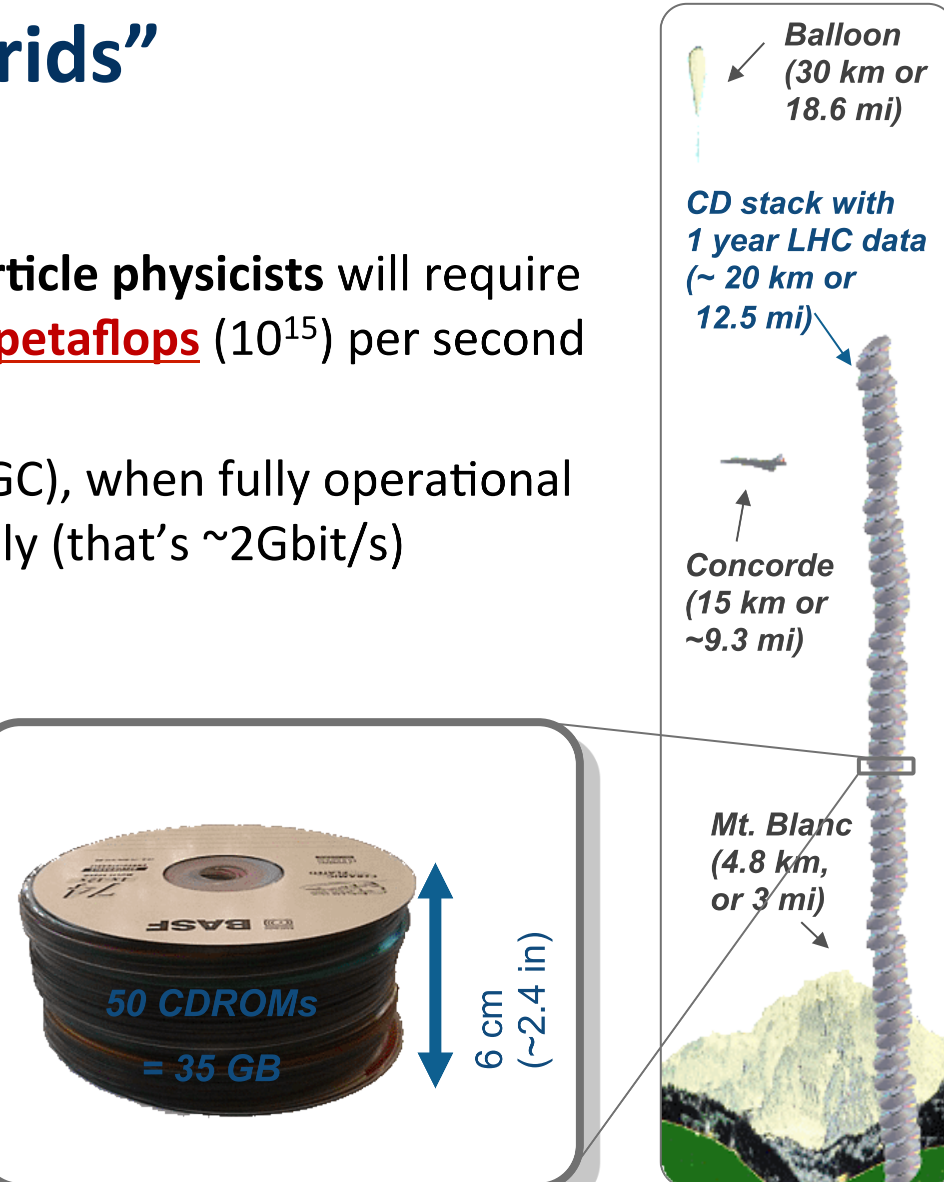
INTEC

Background: “Optical grids”

- eScience:
  - By 2015 it is estimated that **particle physicists** will require exabytes (10^{18}) of storage and **petaflops** (10^{15}) per second of computation
  - CERN’s LHC Computing Grid (LGC), when fully operational generates **15 petabytes** annually (that’s ~2Gbit/s)



LHC = Large Hadron Collidor: particle accellerator



“Optical grids” for consumer services?

E.g., **video editing**:
2Mpx/frame for HDTV, suppose effect requires 10 flops/px/frame, then evaluating 10 options for 10s clip is **50 Gflops** (today’s high performance PC: <5 Gflops/s)

Online gaming:
e.g. Final Fantasy XI: 1.500.000 gamers

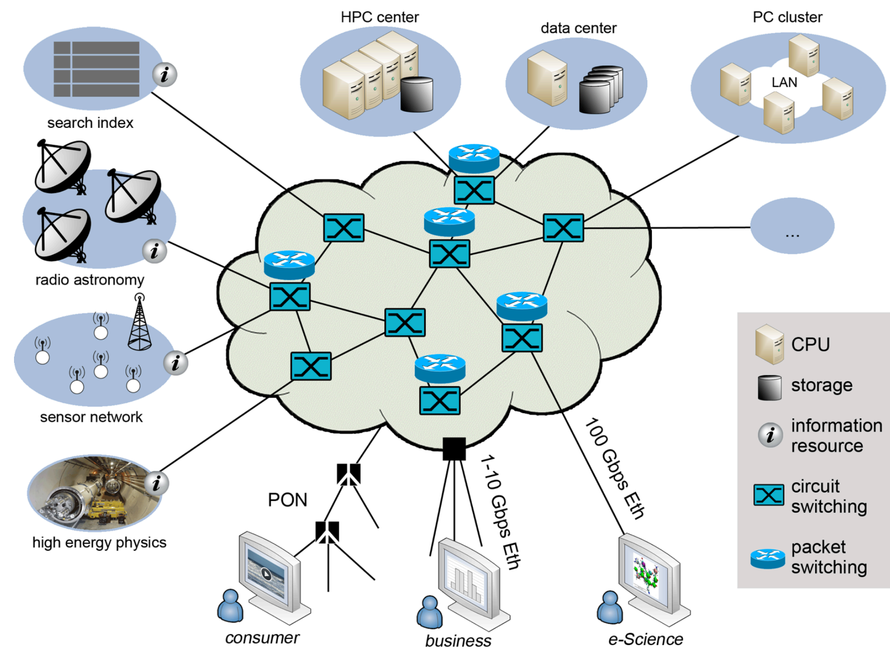
*Virtual reality: rendering of 3*10^8 polygons/s → 10^4 GFlops*

Multimedia editing

Today: towards optical grid / cloud computing

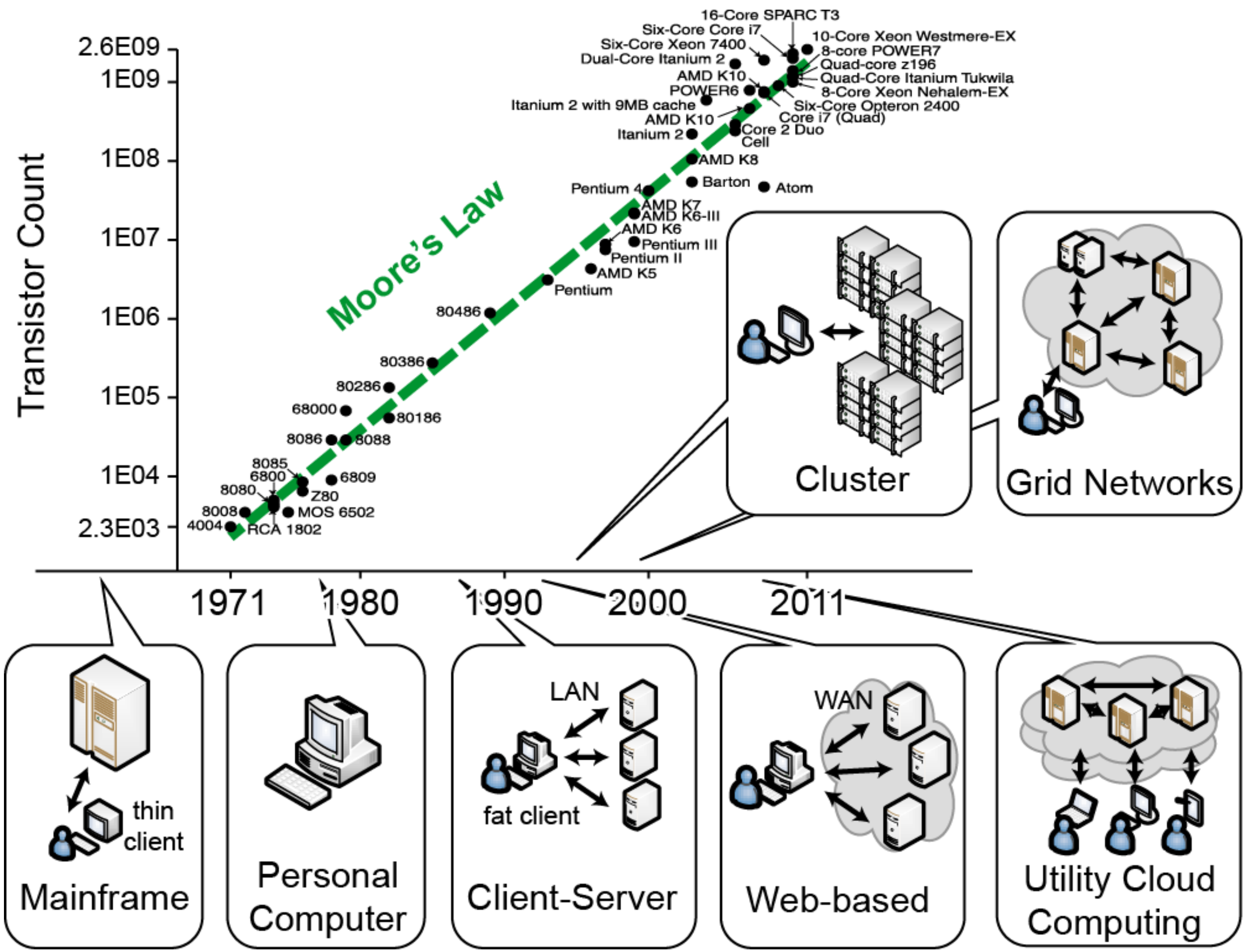
Optical networks crucial for increasingly demanding cloud services, e.g.,

- Computing:
  - High energy physics
  - Amazon EC2, Microsoft Azure
- Online storage:
  - Dropbox, Google Drive, etc.
- Collaboration tools:
  - MSOffice 365, Google Doc
- Video streaming:
  - Netflix, YouTube



C. Develder, et al., *"Optical networks for grid and cloud computing applications"*, Proc. IEEE, Vol. 100, No. 5, May 2012, pp. 1149-1167.

A historical perspective …



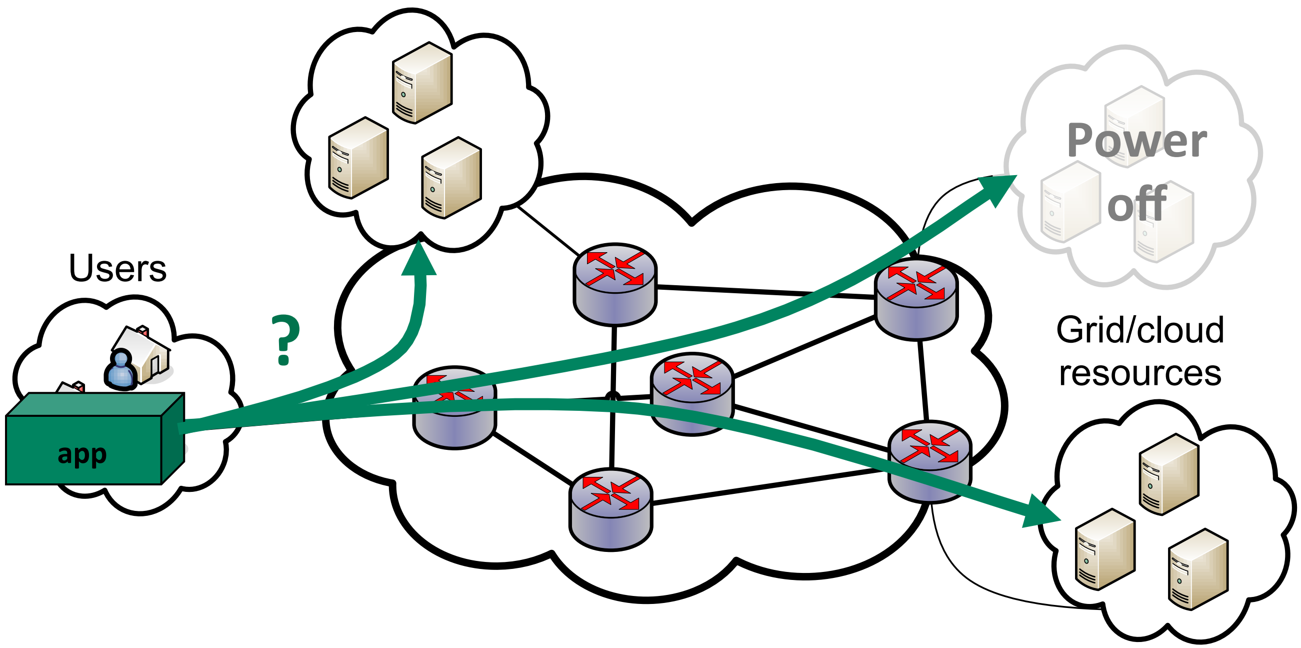
Outline

1. Introduction
2. Network dimensioning for clouds: What's different?
3. An iterative network + server dimensioning approach
4. Exploiting anycast for resilience purposes
5. The next step: accounting for inter-DC synchronization
6. Wrap-up

Dimensioning for clouds: What's different?

Anycast

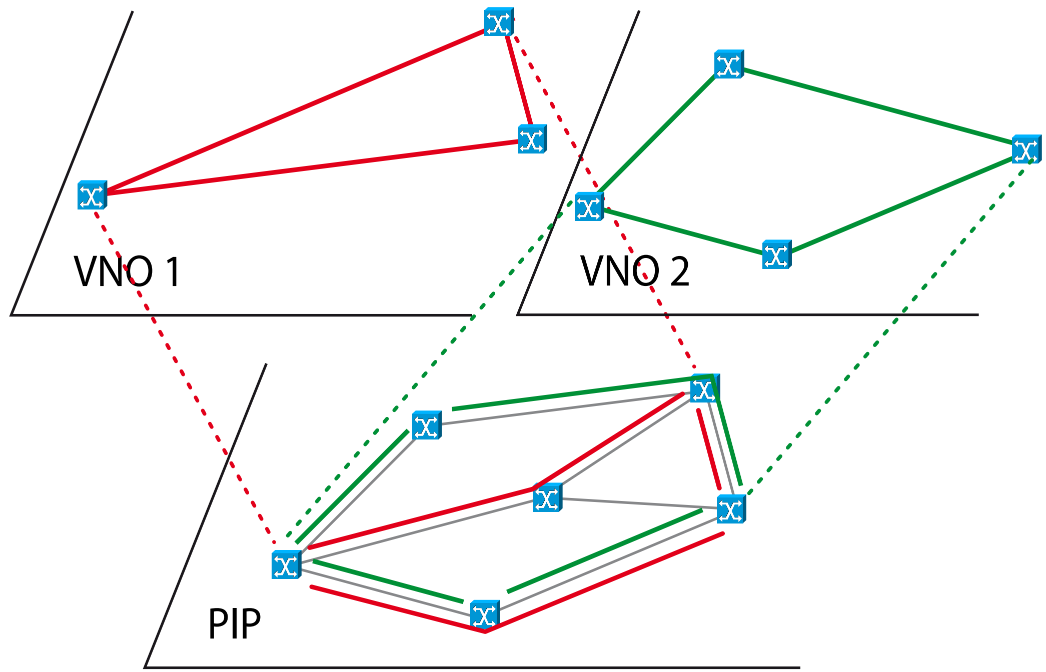
- Users do (in general) **NOT** care where applications are served
  - E.g., virtual machines in IaaS can be instantiated anywhere
  - E.g., bag-of-tasks grid jobs can be run at any server



J. Buysse, K. Georgakilas, A. Tzanakaki, M. De Leenheer, B. Dhoedt and C. Develder, *"Energy-efficient resource provisioning algorithms for optical clouds"*, IEEE/OSA J. Opt. Commun. Netw., Vol. 5, No. 3, Mar 2013, pp. 226-239. doi:10.1364/JOCN.5.000226.

Network virtualization

Physical network is logically partitioned in isolated virtual networks



- **Virtual Network Operators (VNO)** operate logically separated networks
- **Physical Infrastructure Providers (PIP)** have full control over infrastructure (fibers, OXCs)

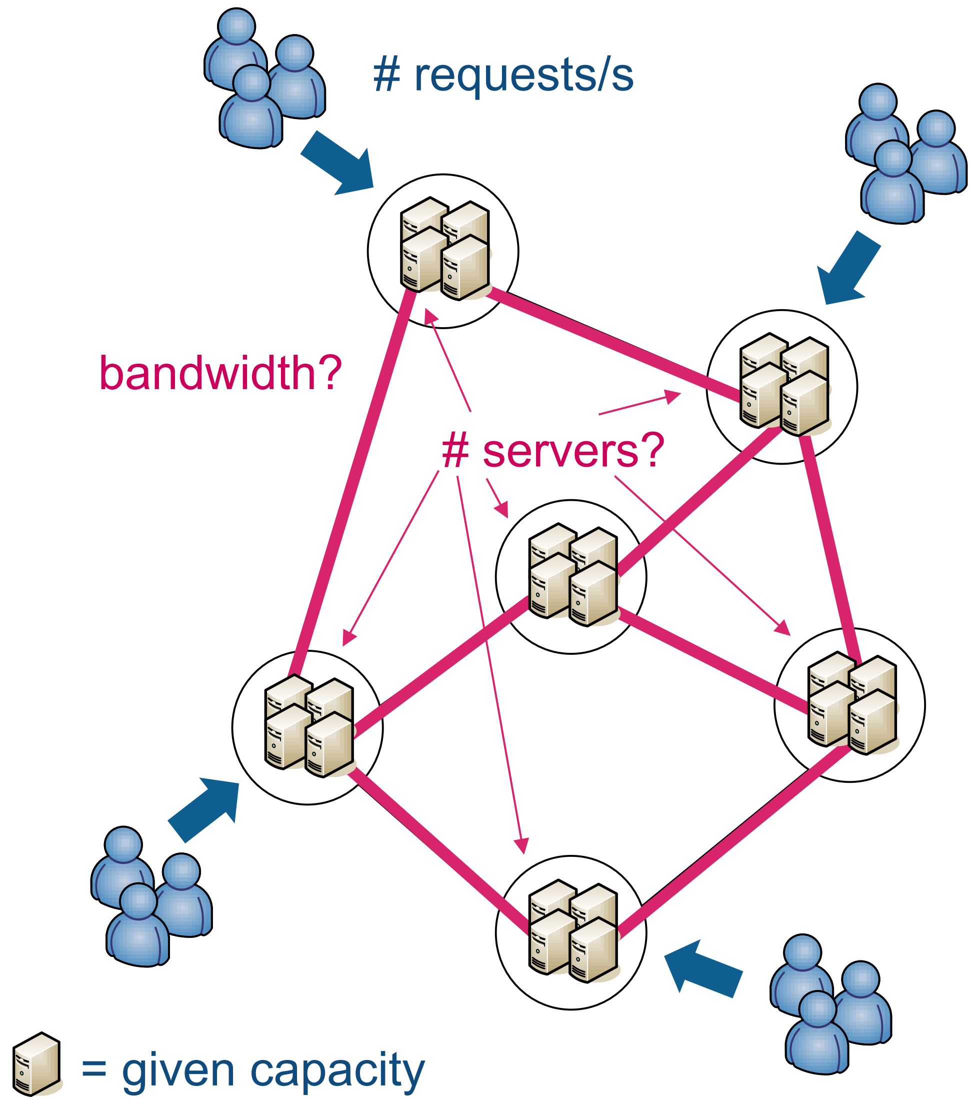
J.A. García-Espín, et al., *"Logical Infrastructure Composition Layer: the GEYSERS holistic approach for infrastructure virtualisation"*, in Proc. TERENA Networking Conference (TNC 2012), Reykjavík, Iceland, 21-24 May 2012.

An iterative network + server dimensioning approach and the impact of scheduling

C. Develder, B. Mukherjee, B. Dhoedt and P. Demeester, *"On dimensioning optical Grids and the impact of scheduling"*, Photonic Netw. Commun., Vol. 17, No. 3, Jun. 2009, pp. 255-265. doi:10.1007/s11107-008-0160-z

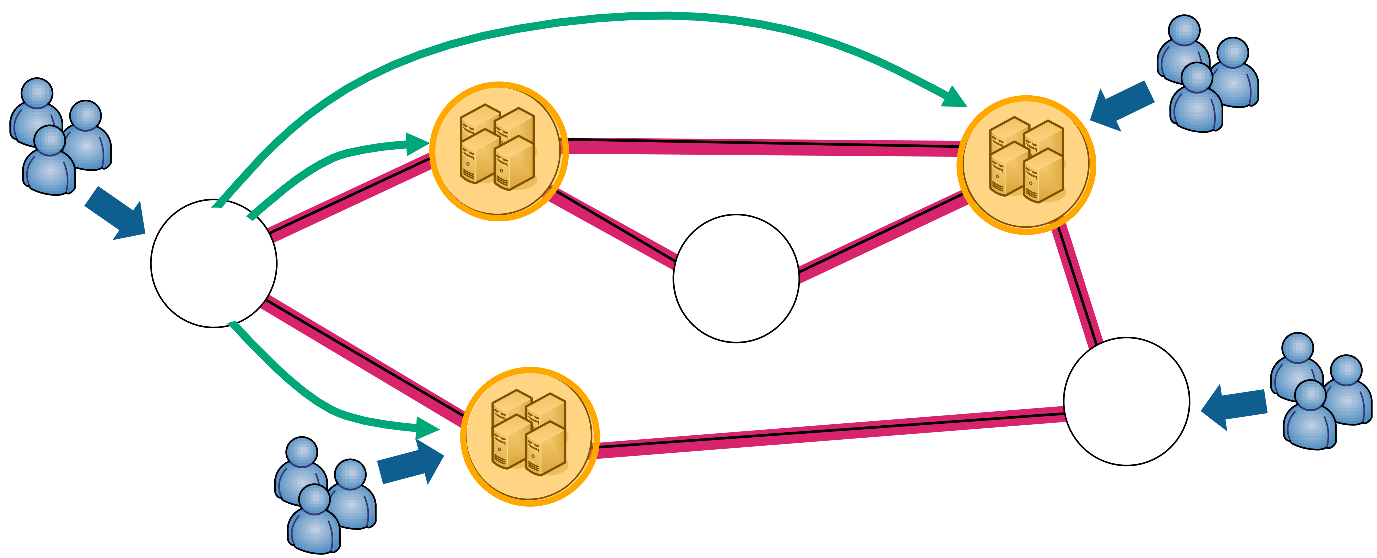
Problem Statement

- Given:
  - Network topology
  - Request arrival process
  - Requested processing capacity
  - Target loss rate
- Find
  - Locations of servers,
  - Amount of servers,
  - Amount of link bandwidth
- While
  - Meeting max. loss
  - Minimizing network capacity



Solution

- Phased approach
  1. Determine K server locations (approx., ILP)
  2. Determine server capacity (analytical, ErlangB)
  3. Determine inter-site bandwidths (simulation)
  4. Dimension link bandwidths (= number of wavelengths)



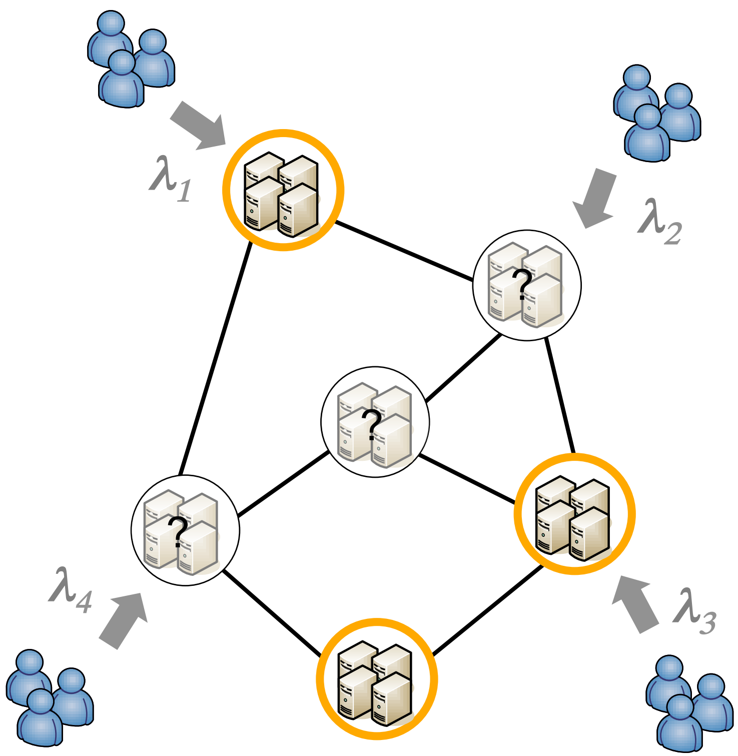
Step 1: Server locations

- Given:
  - Job arrivals at each site
  - Each source S site sends all its requests to a single destination D *(simplifying assumption!)*
  - Shortest path routing is used
- Find:
  - K server locations, minimizing total amount of used link bandwidth

→ ≈ K-means clustering problem

Step 1: Finding the K "best" locations

- Binary variables:
  - t_v = 1 iff site v is server location
  - $f_{vv'}$ = 1 iff request from source v is directed to v'



- Constants:
  - $h_{vv'}$ = cost for sending 1 unit request from source v to server site v'
  - Δ_v = number of unit requests from source v

$$\min \sum_{v} \sum_{v'} \Delta_v \cdot h_{vv'} \cdot f_{vv'}$$

subject to

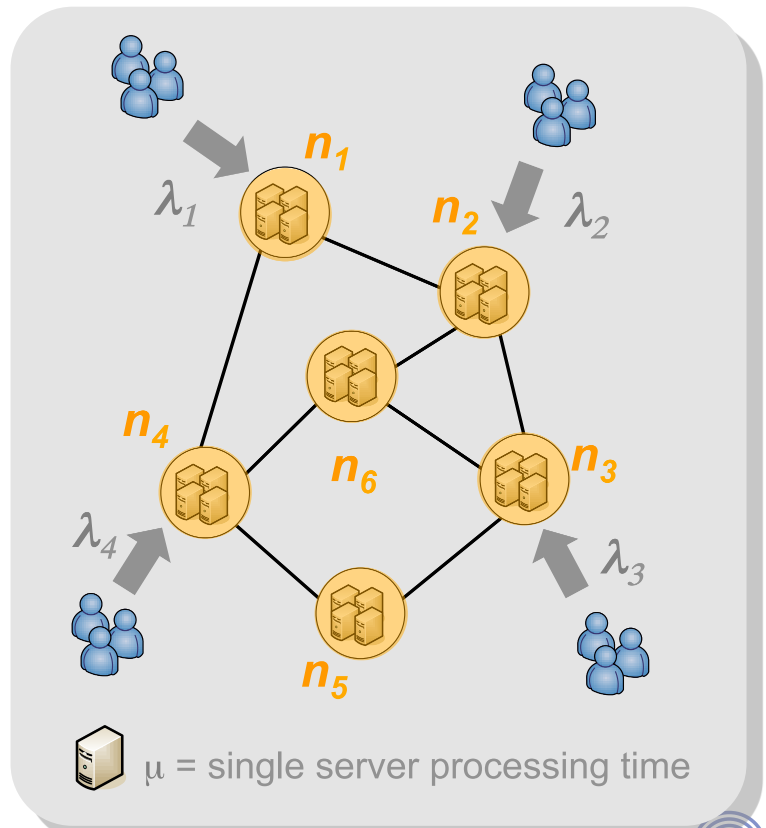
$$\begin{cases} \sum_{v} t_v = K \\ \sum_{v'} f_{vv'} = 1 \quad \forall v \\ f_{vv'} \le t_v \quad \forall v, v' \end{cases}$$

Step 2: Server capacities

- Find number of servers *n*:
  - $L = ErlangB(n,\lambda,\mu) = \frac{(\lambda/\mu)^n/n!}{\sum_{k=0}^{n}(\lambda/\mu)^k/k!}$

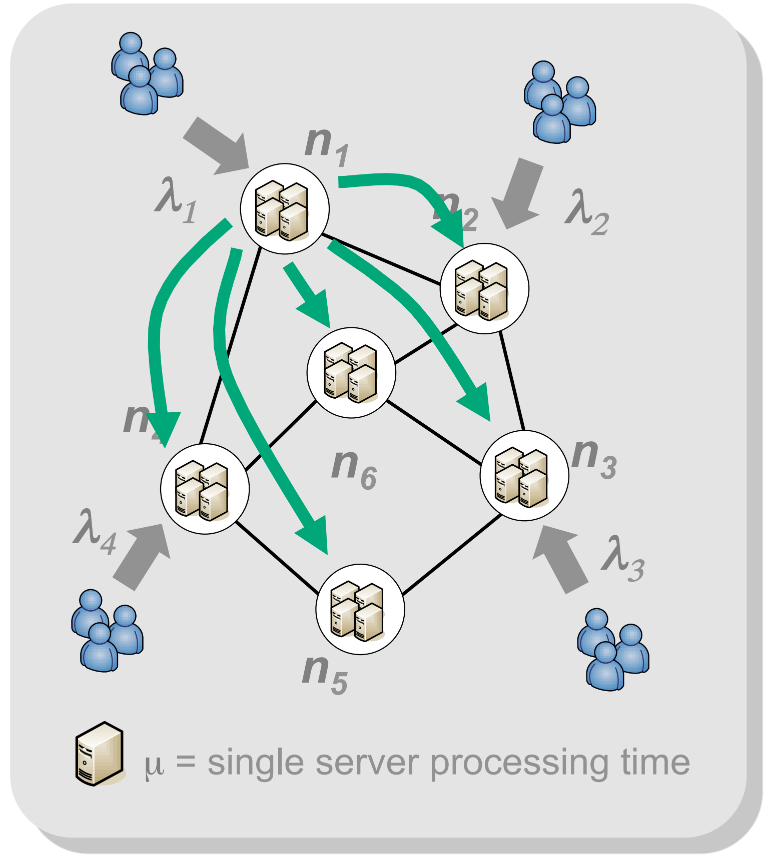
L = target loss
λ = total arrival rate (all N sites)
μ = single server processing time
n = total number of servers

- Distribution among server sites: 3 alternatives
  - ***unif***: uniformly distributed among all sites: $n_i = n/N$
  - ***prop***: proportional to local arrival rate: $n_i = \lambda_i/(N\cdot\lambda)$
  - ***lloss***: try to achieve the same (local) loss rate at each site: $n_i \sim n'_i$ with $L = ErlangB(n'_i, \lambda, \mu)$



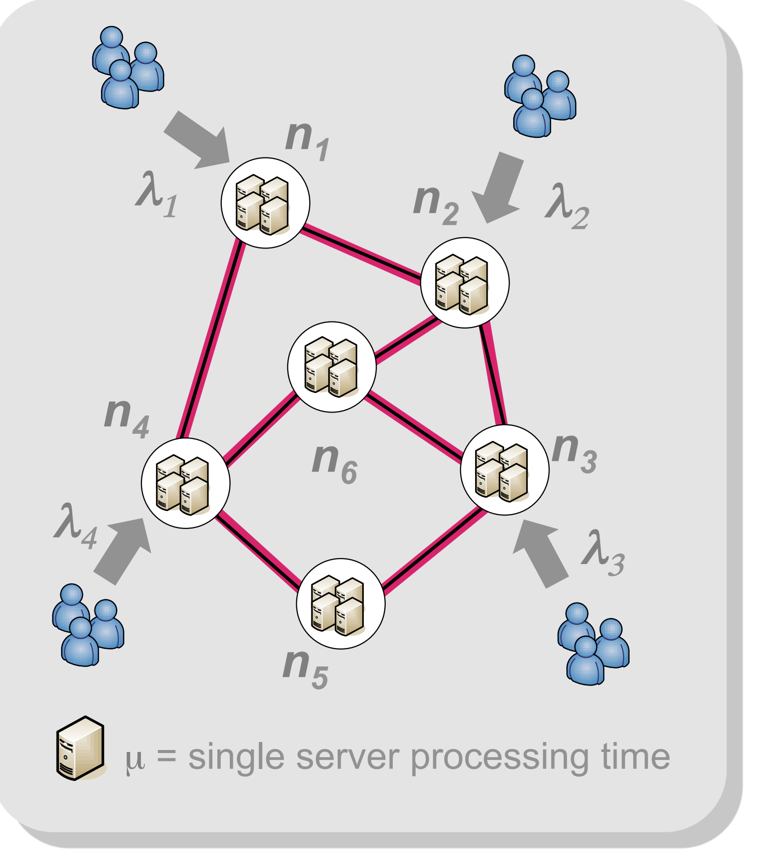
Step 3: Inter-site bandwidth

- Given:
  - Request arrivals
  - Site server capacities
- Find
  - Bandwidth exchanged between sites (i.e., amount of requests)
  - Scheduling alternatives: <u>*alwayst try local*</u>, if busy then
    - **SP**: shortest path (i.e., closest free server)
    - **rand**: randomly pick a free site
    - **mostfree**: choose site with most free servers



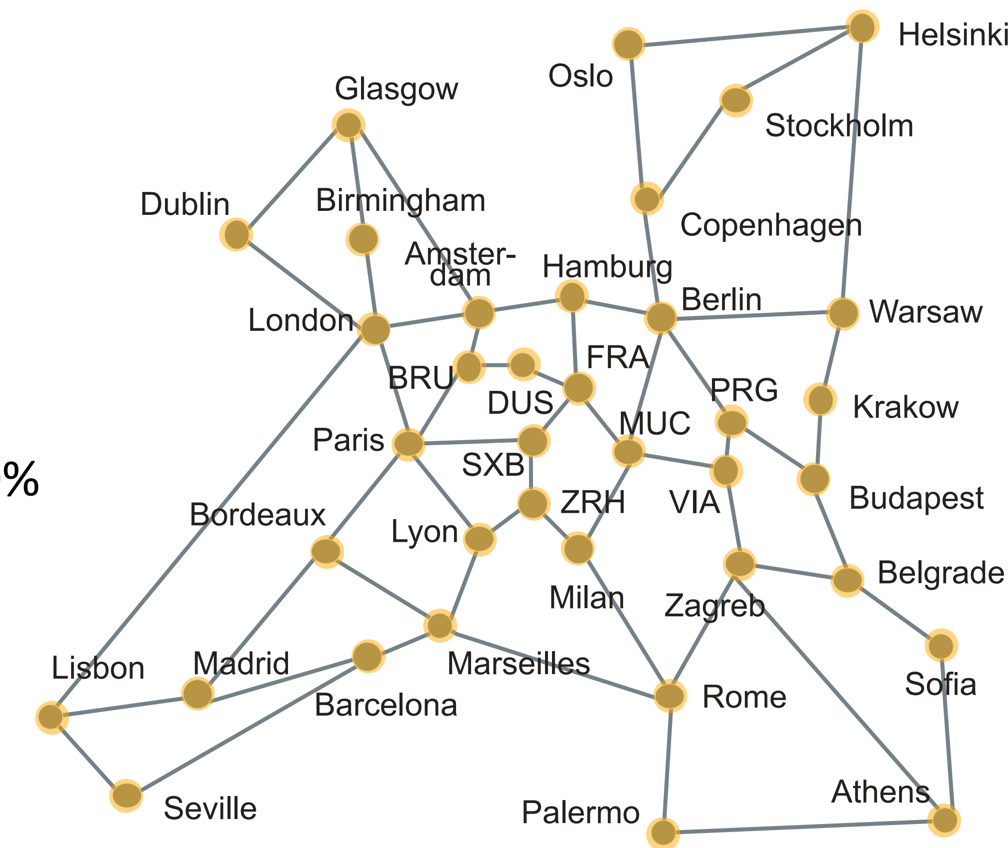
Step 4: Link dimensions

- Given:
  - Inter-site request arrivals (from step 2)
  - Shortest path routing (assumption)
- Find
  - Link bandwidth (amount of wavelengths)

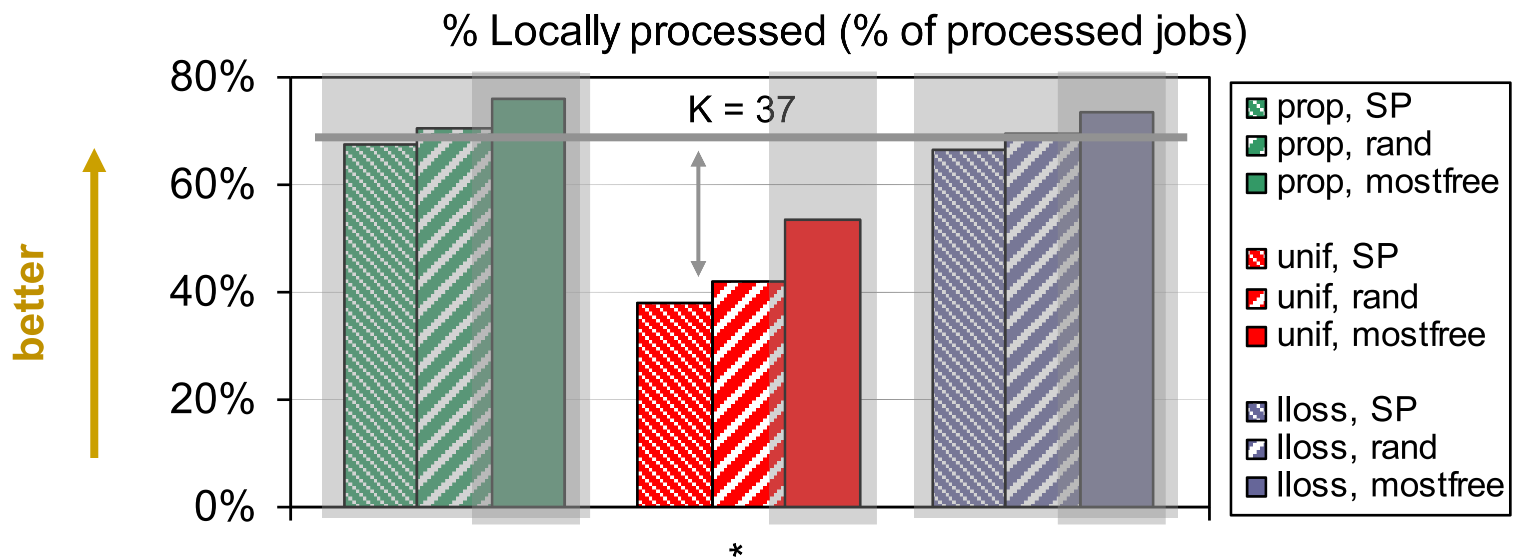


Case study

- Topology:
  - EU topology
  - 37 nodes
  - 57 links
- Request arrivals λ_i:
  - Random
  - 30% uniformly in [1,15], 70% uniformly in [30,60]
- Server capacity:
  - 1 request / time unit
- ErlangB:
  - Max. 5% loss
    $\Rightarrow$ 799 servers



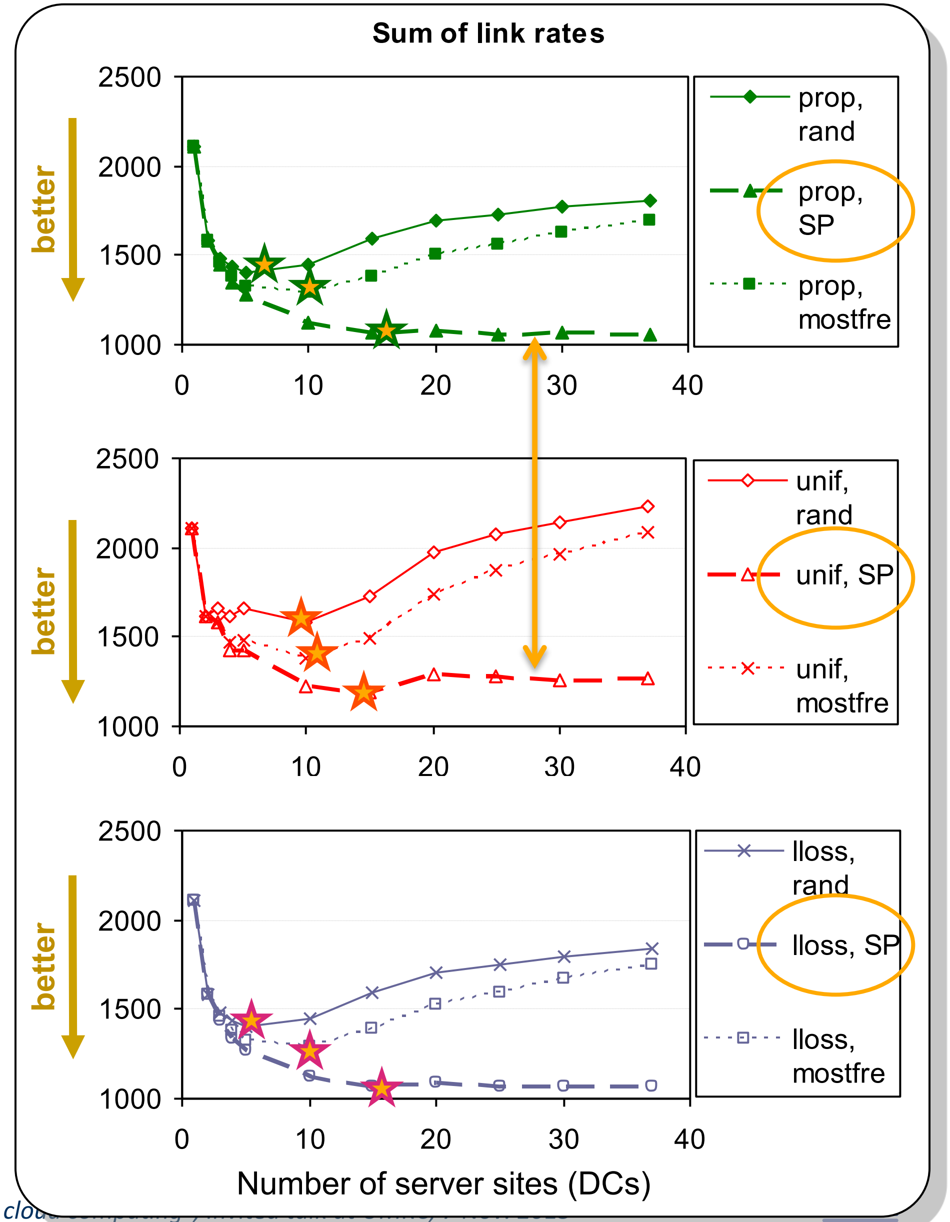
Case study results: 'Local' processing rate



- Server distribution:
  - ***unif***: uniformly distributed
  - ***prop***: ~ local arrival rate
  - ***lloss***: ~ same (local) loss rate
- Scheduling: local first, if busy then…
  - ***SP***: shortest path
  - ***rand***: randomly pick a free site
  - ***mostfree***: site with most free servers
- Conclusions:
  - ***mostfree*** achieves highest local processing
  - Intelligent server placement (prop, lloss) achieves higher local processing

Case study results: Link bandwidths

- Influence of *# server sites*:
  - There is an “optimal” value, depending on the scheduling algorithm & server distribution
- Influence of *scheduling*:
  - SP scheduling obviously leads to lowest total link bandwidth
- Influence of *server distribution*:
  - Non-uniform server distribution (prop, lloss) leads to significant BW reduction



C. Develder, "Dimensioning (optical) networks for cloud computing"

Conclusions wrt dimensioning

- Proposal of dimensioning approach
  - Sequential approach (first server locations & dimensions, then network)
  - Combination of analytics and simulation

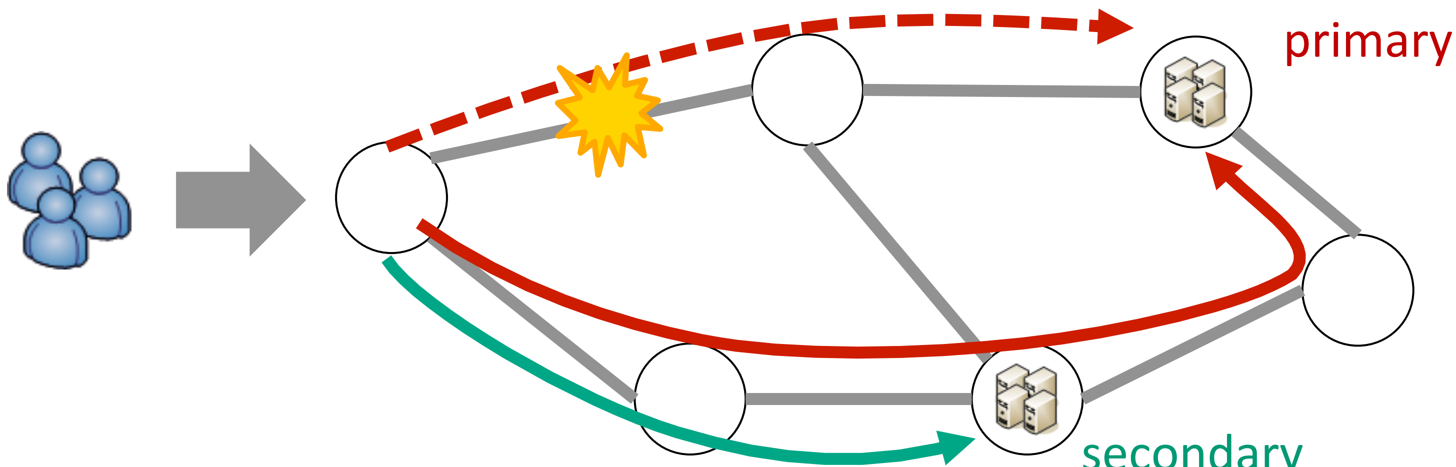
- Comparison of site dimensioning and scheduling alternatives
  - Dimensioning: intelligent server placement allows higher local processing
  - Scheduling: maximizing “local” processing may come at link bandwidth price

Exploiting anycast for resilience

C. Develder, J. Buysse, B. Dhoedt and B. Jaumard, *"Joint dimensioning of server and network infrastructure for resilient optical grids/clouds"*, IEEE/ACM Trans. Netw., Vol. PP, Oct. 2013, pp. 1-16. doi:10.1109/TNET.2013.2283924

Exploiting relocation

- Dimension optical grid/cloud so that it is resilient against failures
- Exploit anycast principle: allow rerouting to other destinations



J. Buysse, M. De Leenheer, B. Dhoedt and C. Develder, *"Providing resiliency for optical grids by exploiting relocation: A dimensioning study based on ILP"*, Comput. Commun., Vol. 34, No. 12, Aug. 2011.

A. Shaikh, J. Buysse, B. Jaumard and C. Develder, *"Anycast routing for survivable optical grids: scalable solution methods and the impact of relocation"*, IEEE/OSA J. Opt. Commun. Netw., Vol. 3, No. 9, Sep. 2011.

Problem statement

Given

- Topology (sources, candidate data center locations, OXCs)
- Demand (for given sources)
- **Survivability** requirements (e.g., link and/or node failures)

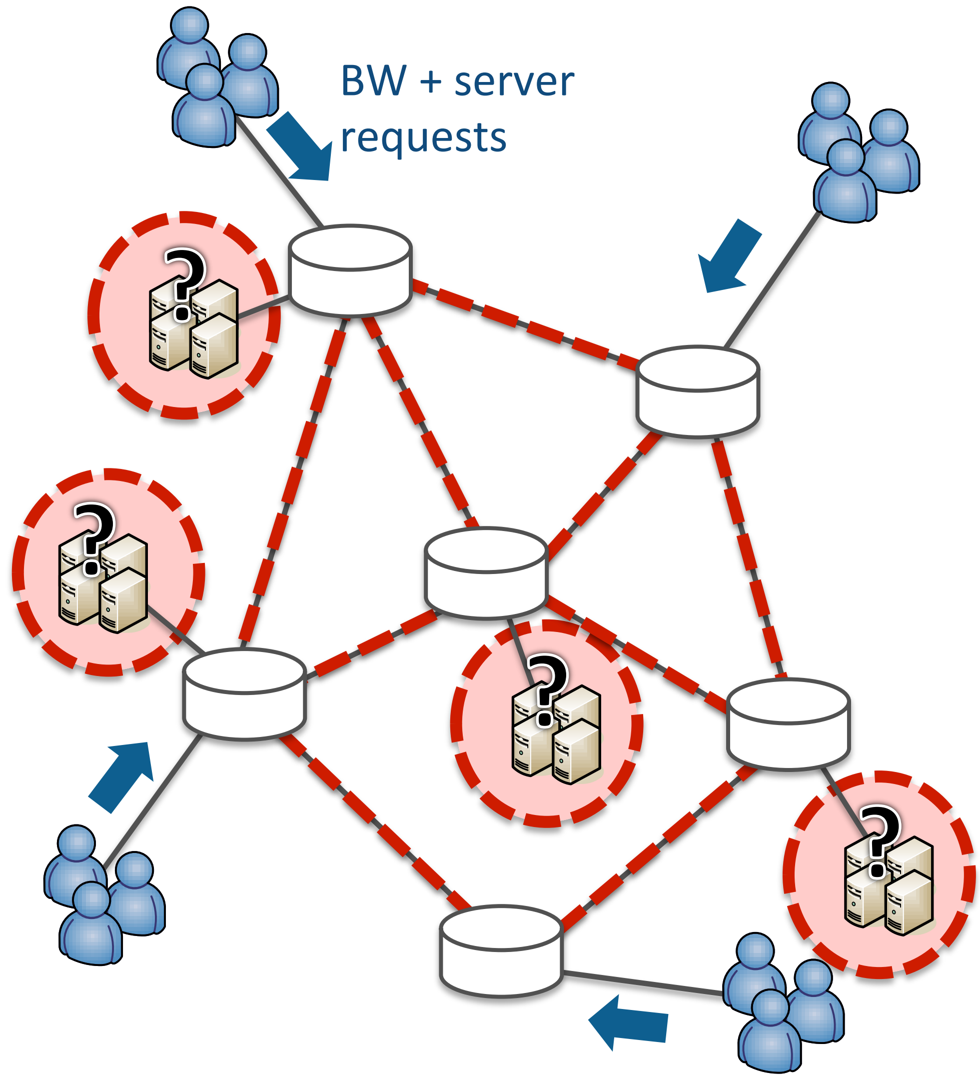
Shared protection

Find

- K locations (chosen from candidate data center locations)
- Destination sites and routes
- Network and server capacity

Such that

- Network and server resources are minimized

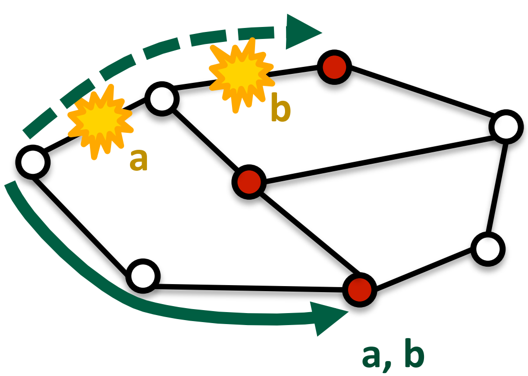


Solution approach

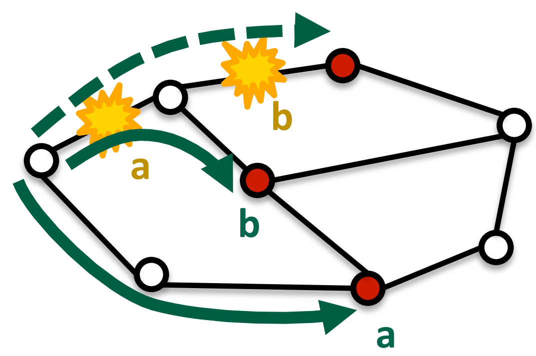
Step 1: Find the K best data center locations

Step 2: Find the primary/secondary destinations + paths towards them

Failure-Independent (FID) rerouting
=> Column generation

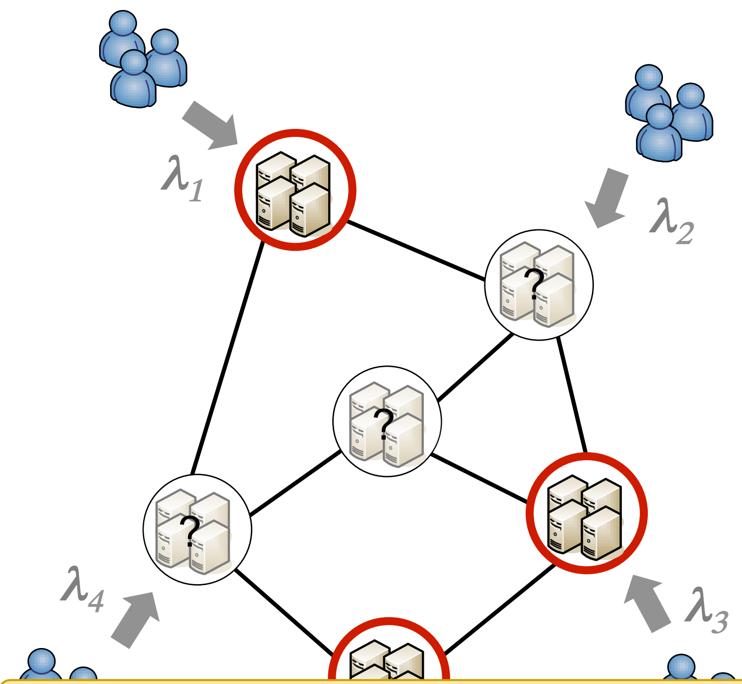


Failure-Dependent (FD) rerouting
=> Single ILP



Step 1: Finding the K "best" locations

- Binary variables:
  - t_v = 1 iff site v is server location
  - $f_{vv'}$ = 1 iff request from source v is directed to v'



- Constants:
  - $h_{vv'}$ = cost for sending 1 unit request from source v to server site v'
  - Δ_v = number of unit requests from source v

$$\min \sum_{v} \sum_{v'} \Delta_v \cdot h_{vv'} \cdot f_{vv'}$$

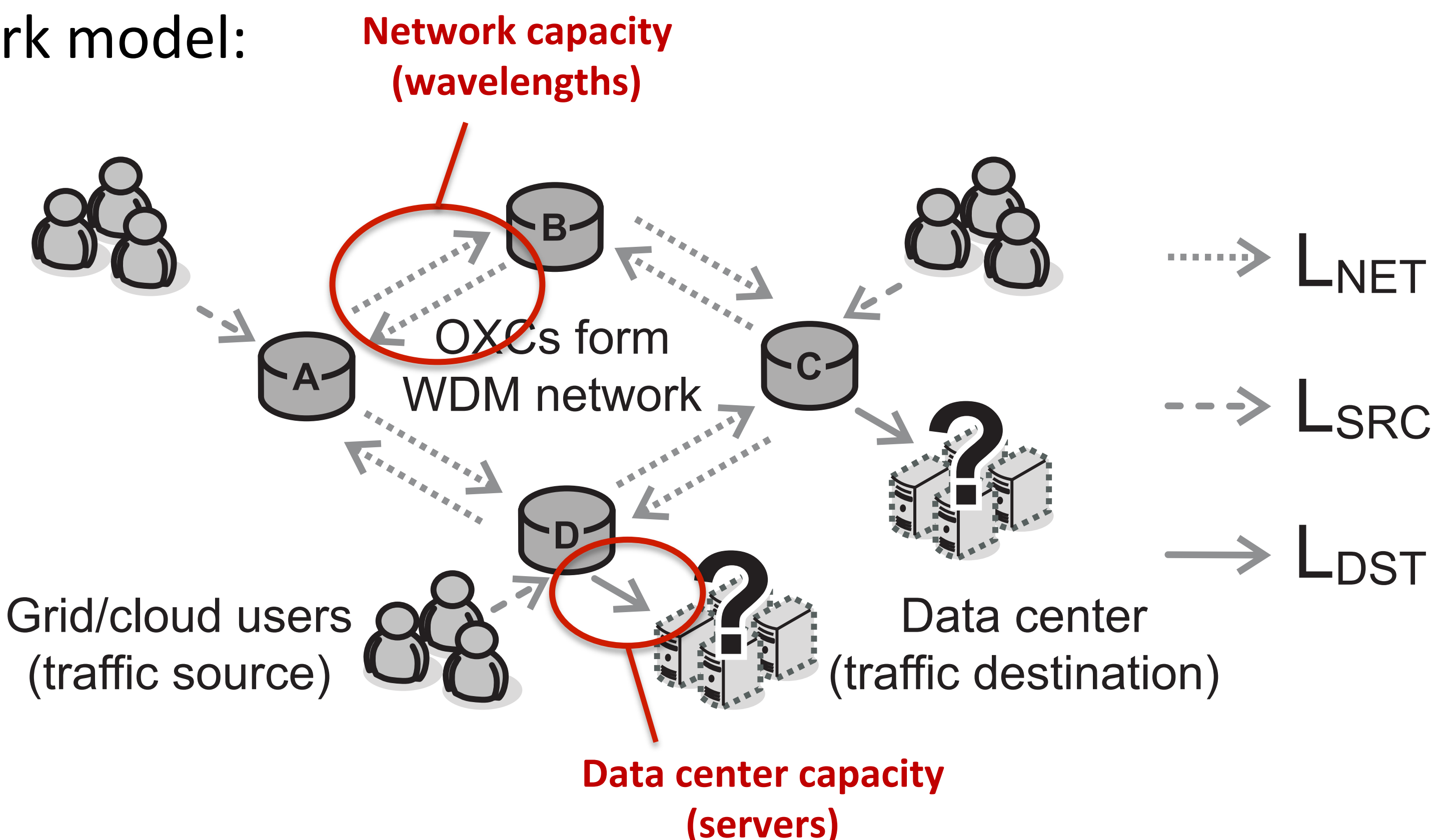
subject to

$$\begin{cases} \sum_{v} t_v = K \\ \sum_{v'} f_{vv'} = 1 \quad \forall v \\ f_{vv'} \le t_v \quad \forall v, v' \end{cases}$$

C. Develder, B. Mukherjee, B. Dhoedt and P. Demeester, *"On dimensioning optical Grids and the impact of scheduling"*, Photonic Netw. Commun., Vol. 17, No. 3, Jun. 2009

Step 2: Find destinations and routes towards them

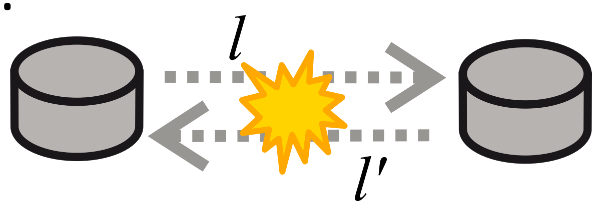
- Network model:



- w_l = "capacity" of link l
- Capacity = wavelengths for NET links, servers for DST links!

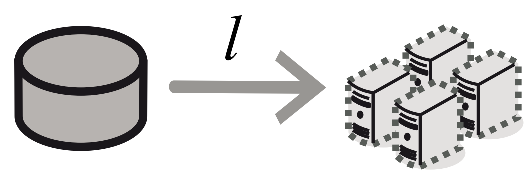
Step 2: Find destinations and routes towards them

- Failure: modeled as SRLG = set of links that simultaneously fail
- Single link failure:



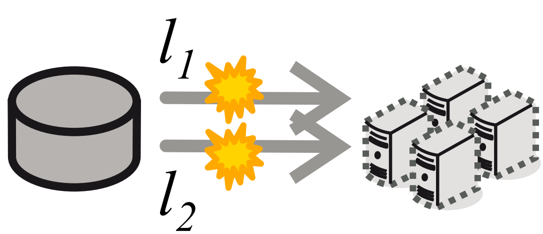
$$s = \{l, l'\}$$

- Single server failure: *1:N* protection [= add 1 for case single one out of N fails]
  - No relocation:
    - Let x = number of servers under working conditions
    - Then we need $\lceil (1+1/N) \cdot x \rceil$ servers



$$w_l \geq \rho_l \cdot x$$
$$\rho_l = 1 + 1/N$$

  - Relocation: consider $(1+N)$ parallel links, at most 1 fails



$$s_1 = \{l_1\}$$
$$s_2 = \{l_2\}$$

Step 2: Find destinations and routes towards them

- Failure dependent (FD) rerouting => Single ILP
- Variables:
  - p_{vls} : number of unit demands with source v that cross link l under failure s
  - w_l : capacity on link l
- Objective:

Network capacity (wavelengths)

Data center capacity (servers)

$$\min \left(\sum_{\ell \in L_{\text{NET}}} w_\ell + \alpha \sum_{\ell \in L_{\text{DST}}} w_\ell \right)$$

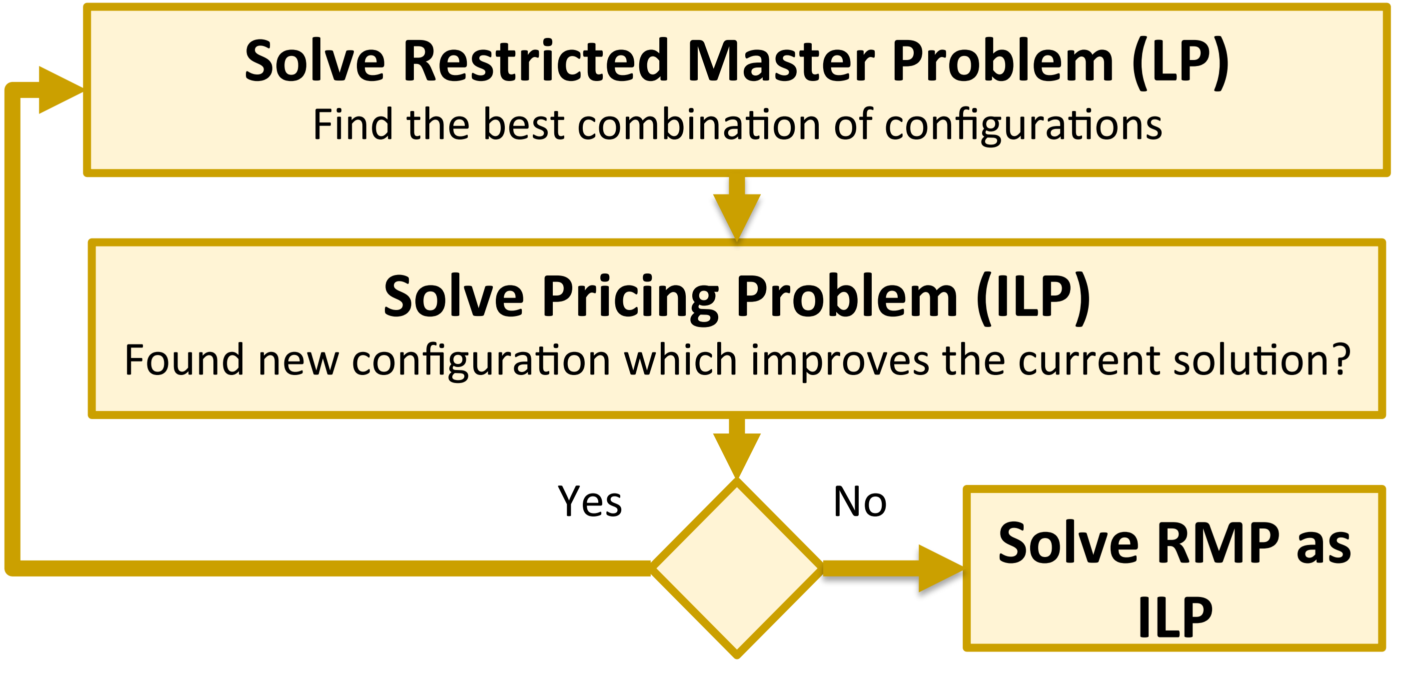
- Constraints:
  - p_{vls} : flow constraints + don't use failing links when protecting against s
  - w_l : count capacity

1 for network link
1+1/N for server link, in case of relocation

$$w_\ell \geq \rho_\ell \cdot \sum_{v \in V_{\text{SRC}}} p_{v\ell s} \qquad \forall s \in S$$

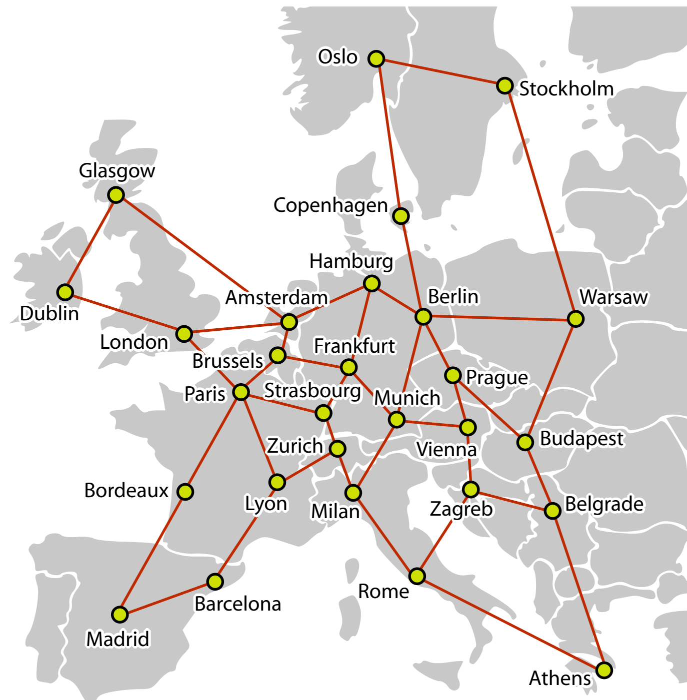
Step 2: Find destinations and routes towards them

- **Failure-independent (FID) rerouting** => Column generation:
  - Assume: given "configurations" = combination of working and backup paths
  - Restricted Master Problem (RMP) finds best combination of configurations
  - Pricing Problem (PP) finds new configuration that can reduce cost



Case study set-up

- Topology
  - European network
  - 28 nodes and 41 bidirectional links



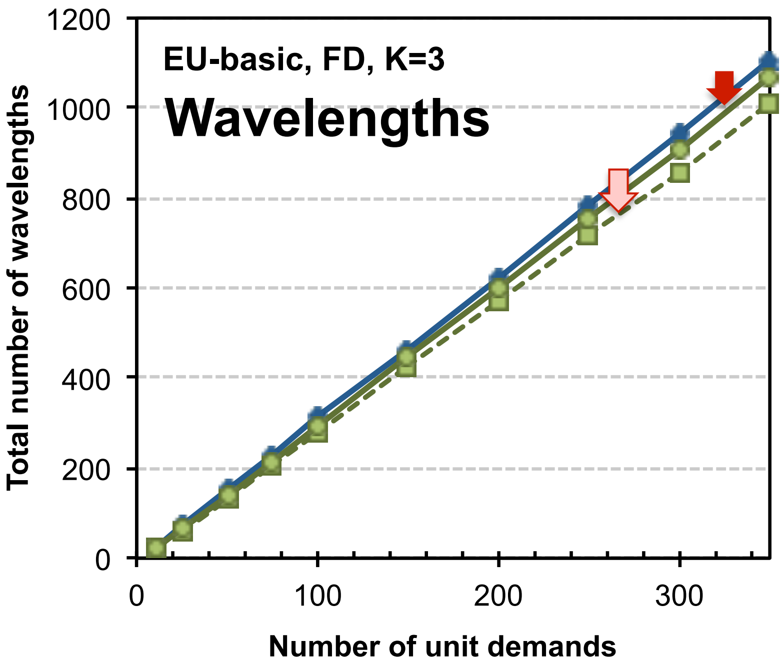
- Demand
  - Randomly generated requests (10-350)
  - 10 instances for each number of requests

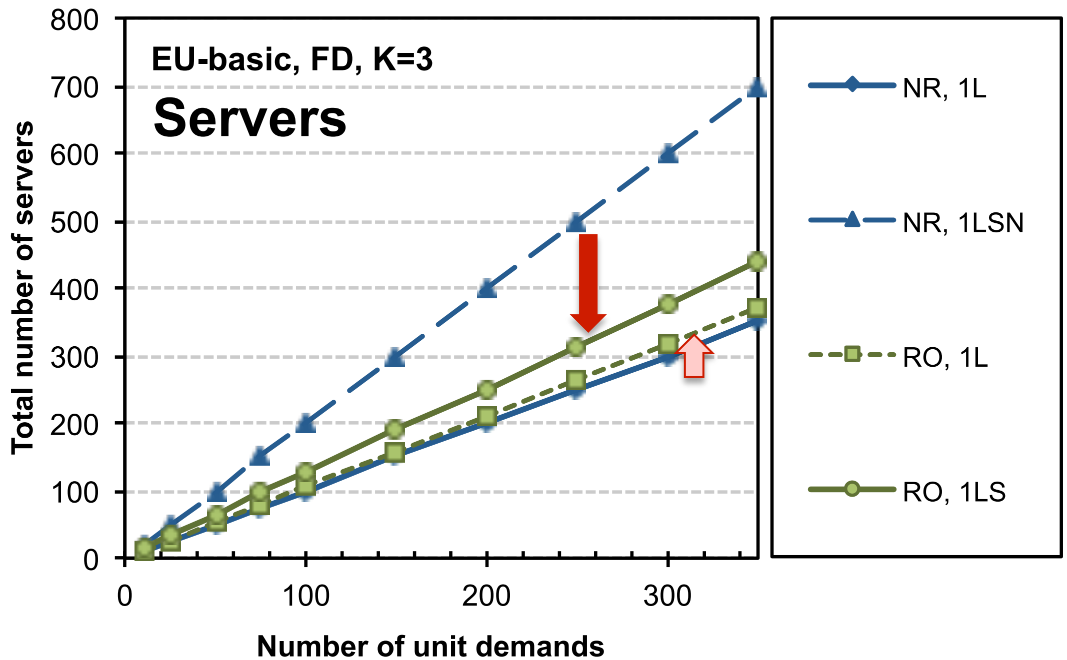
- **Four scenarios:**

	No relocation		Exploiting relocation
Single link failures:	***1L, NoReloc***	⇔	***1L, Reloc***
Single failures of either link or server:	***1LSN, NoReloc***	⇔	***1LS, Reloc***

The impact of relocation

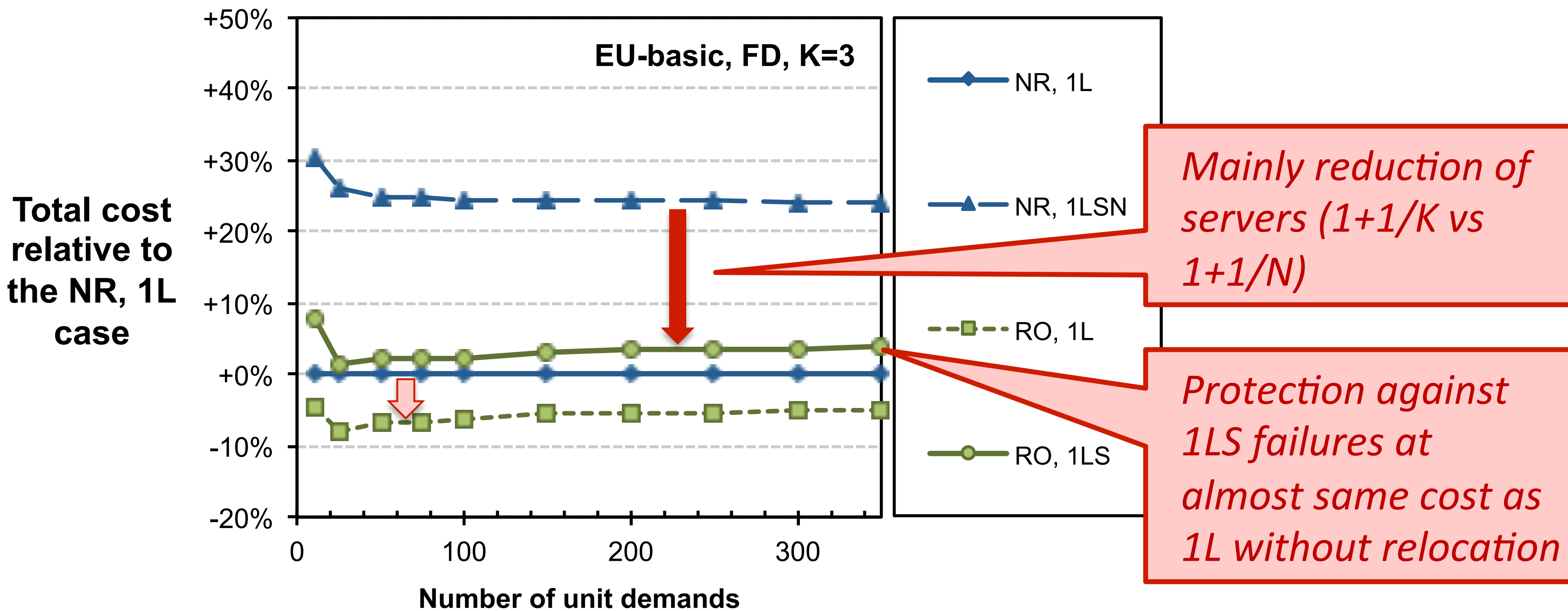
- Single Link failures (1L):
  - Reduction of backup wavelengths
  - Slight increase in server capacity
- Single link/server failure (1LS)
  - Reduction of backup wavelengths
  - Fewer servers than 1:N server protection (N=1)





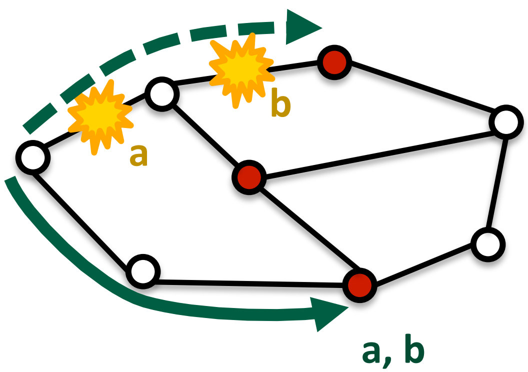
The impact of relocation

- Single Link failures (1L):
  - Reduction of backup wavelengths
  - Slight increase in server capacity
- Single link/server failure (1LS)
  - Reduction of backup wavelengths
  - Fewer servers than 1:N server protection (N=1)

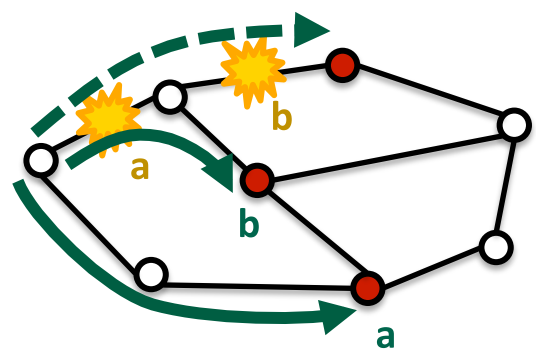


Failure dependent rerouting? (FD vs FID)

Failure-Independent (FID) rerouting

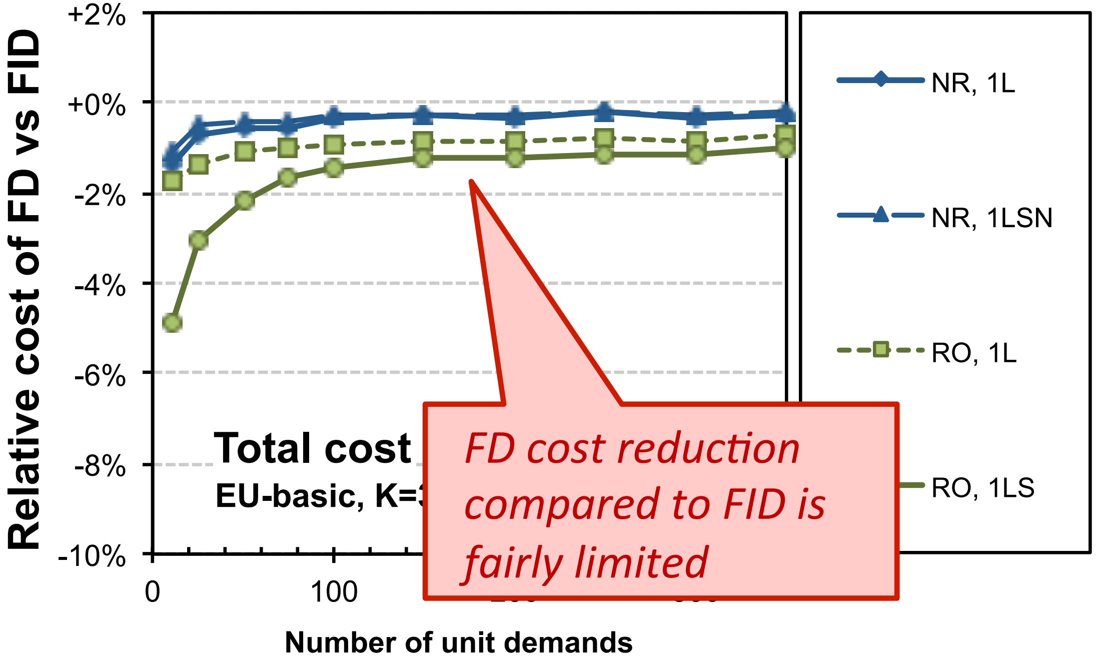


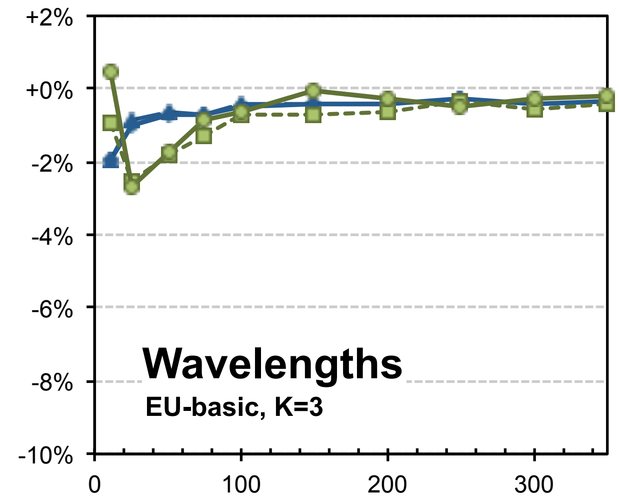
Failure-Dependent (FD) rerouting

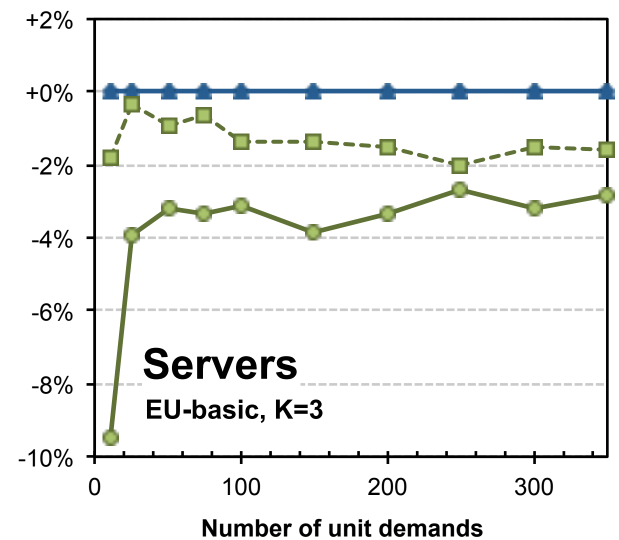


Failure dependent rerouting? (FD vs FID)

- FD is best, obviously
- Yet, difference is limited (few %)
  – at least for small K (= number of server sites)



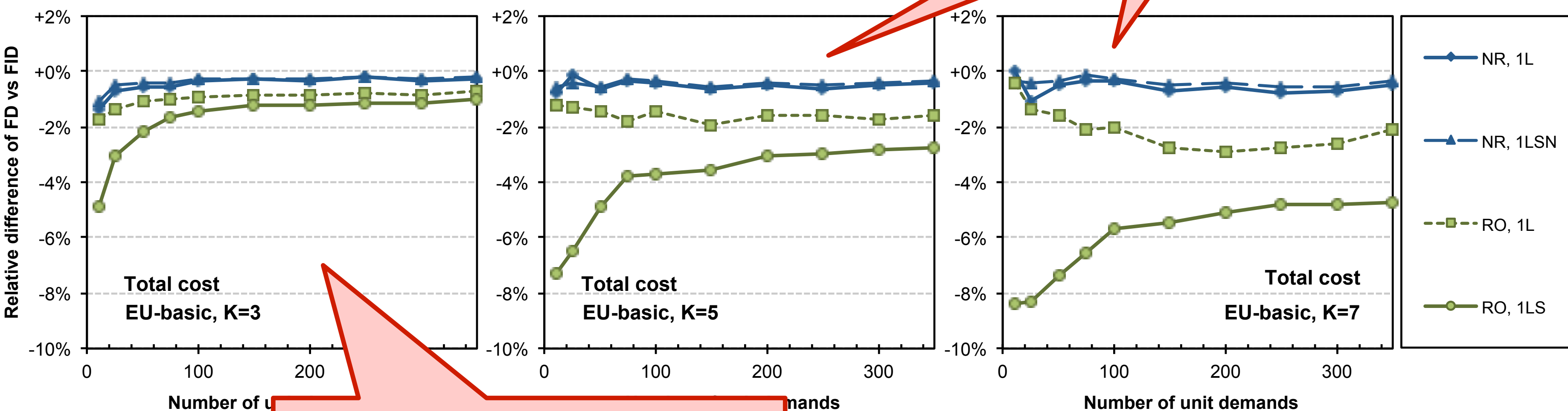




Failure dependent rerouting? (FD vs FID)

- FD is best, obviously
- Yet, difference is limited (few %)
  – at least for small K (= number of server sites)

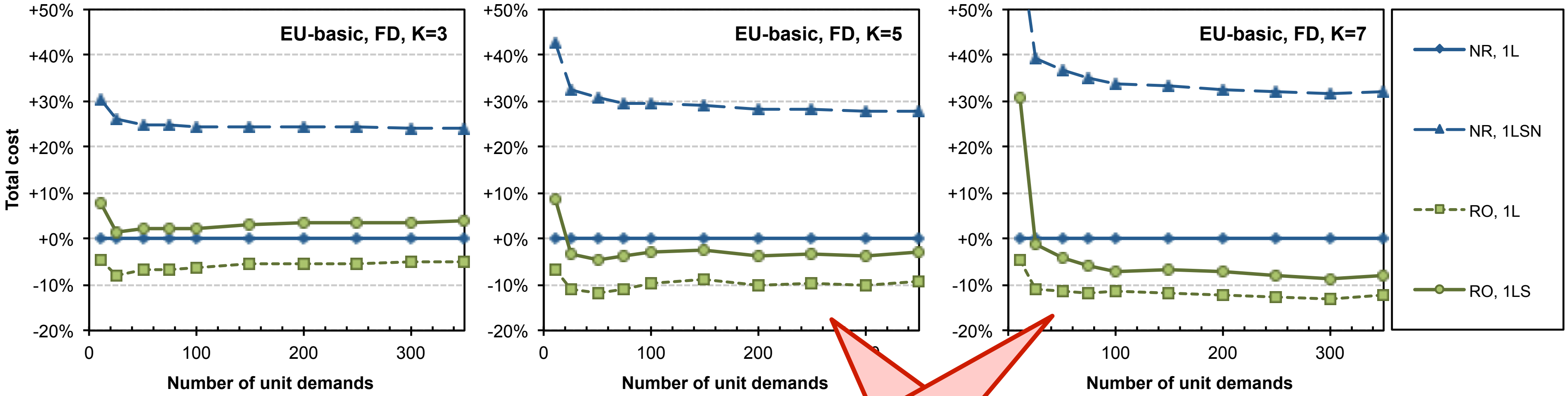
FD advantage increases for larger number of server sites!



FD cost reduction larger for case of relocation (esp. 1LS)

Influence of K on benefit of relocation?

- K ➚
  - Wavelength reduction more pronounced
  - Lower extra cost to provide single server failure protection



Relocation advantage increases for larger number of server sites!

Conclusions

- Dimensioning algorithm for resilient optical grids
  - Exploit relocation for resiliency
  - Compact ILP for finding K best locations
  - ILP (w/ column generation) for server & network dimensions
  - Generic model based on SRLG concept

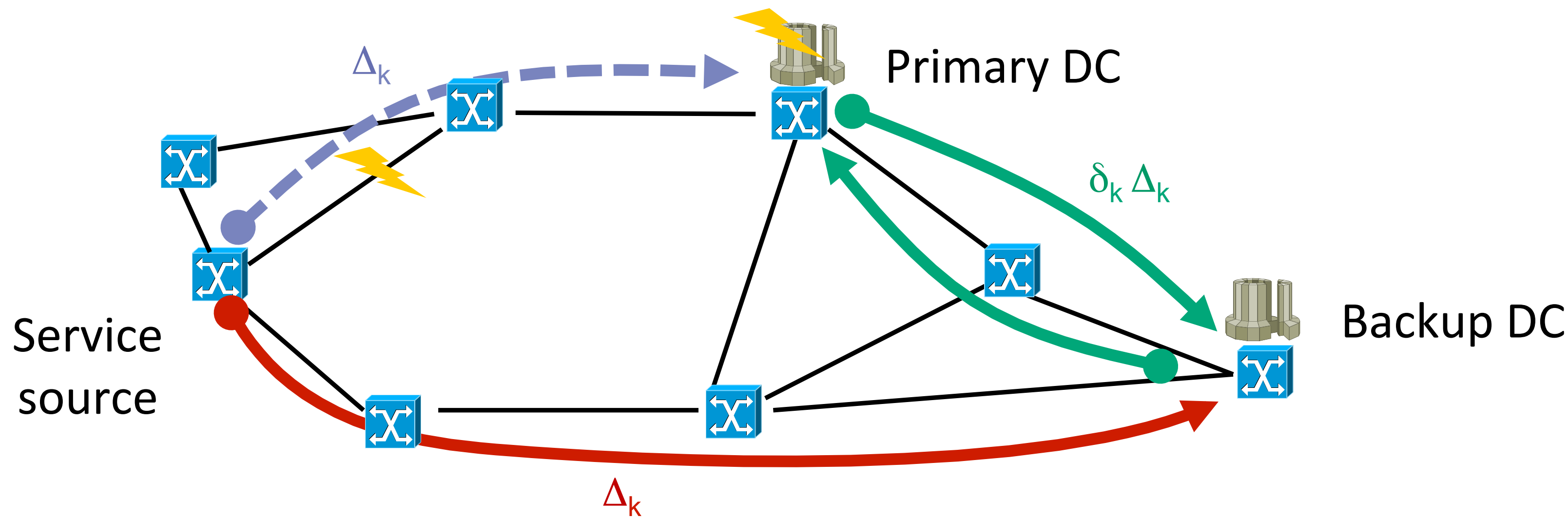
- Case study on EU network topology [10-350 unit demands]
  - Relocation offers cost advantage of up to ca. 10% to protect against single network link failures
  - Total cost to protect against 1LS with relocation
    ≈ Cost to protect against 1L only, without relocation
  - Relocation advantage more substantial for larger number of server sites
  - Failure-dependent rerouting advantage if we use relocation (couple of %)

The next step: virtualization & accounting for server synchronization

M. Bui, B. Jaumard and C. Develder, *"Anycast end-to-end resilience for cloud services over virtual optical networks (Invited)"*, in Proc. 15th Int. Conf. Transparent Optical Netw. (ICTON 2013), Cartagena, Spain, 23-27 Jun. 2013. doi:10.1109/ICTON.2013.6603032

Protection scheme concept

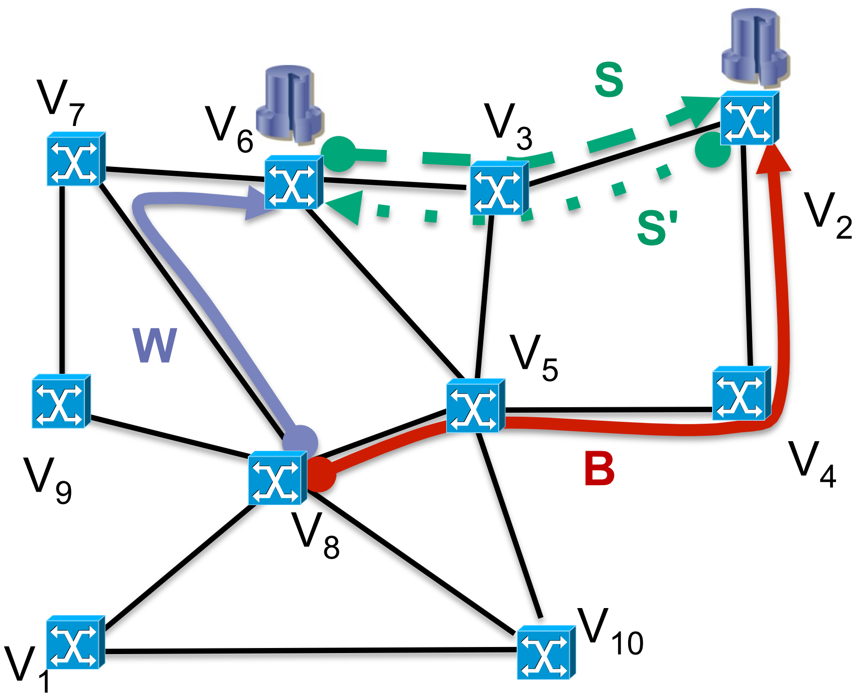
- Covered failures
  - Network links → disjoint **primary** and **backup** paths
  - Data centers → disjoint primary and backup server locations



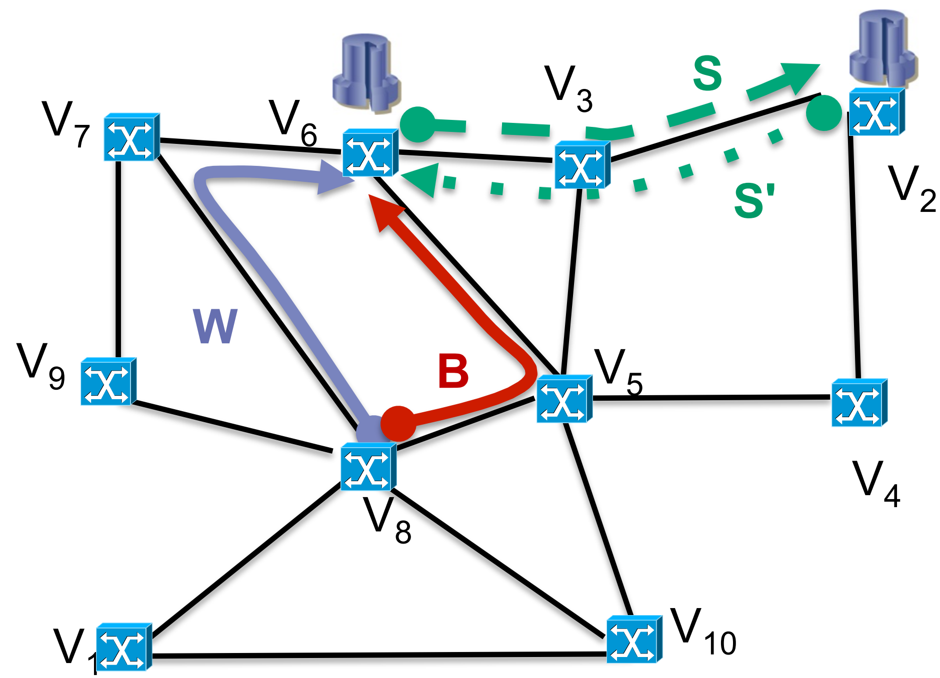
- Note: **synchronization** between data centers for smooth fail-over switching!
  - We assume: sync needs fraction δ of service bandwidth Δ

Two proposed protection schemes

Scheme 1: VNO-resilience



Scheme 2: PIP-resilience



Problem statement

Given

- Cloud network topology: $G = (V, L)$, with V = nodes, L = links
- Locations of the data centers, $V_D \subseteq V$
- Set of service requests, K
  - v_k: source of service
  - Δ_k: bandwidth requirement
  - Services originating from the same source are aggregated

Find

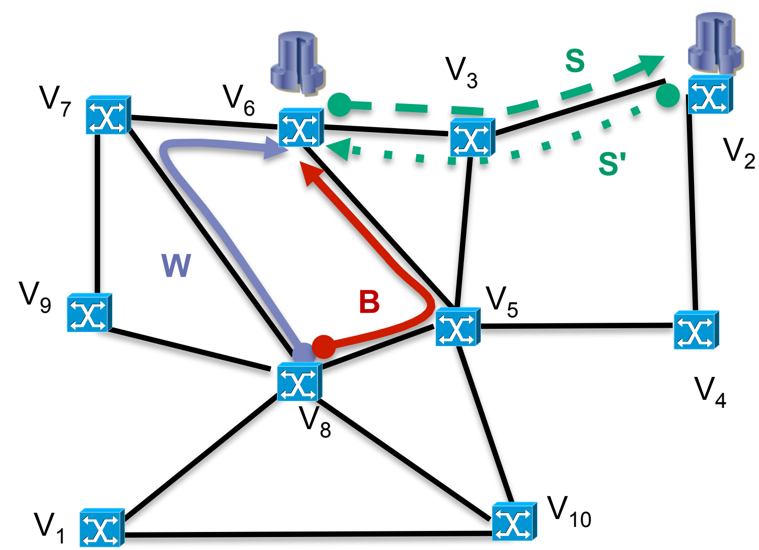
- Choice of primary and backup DC locations for each service
- Primary, backup *and synchronization* paths

Such that total used network bandwidth utilization is minimized

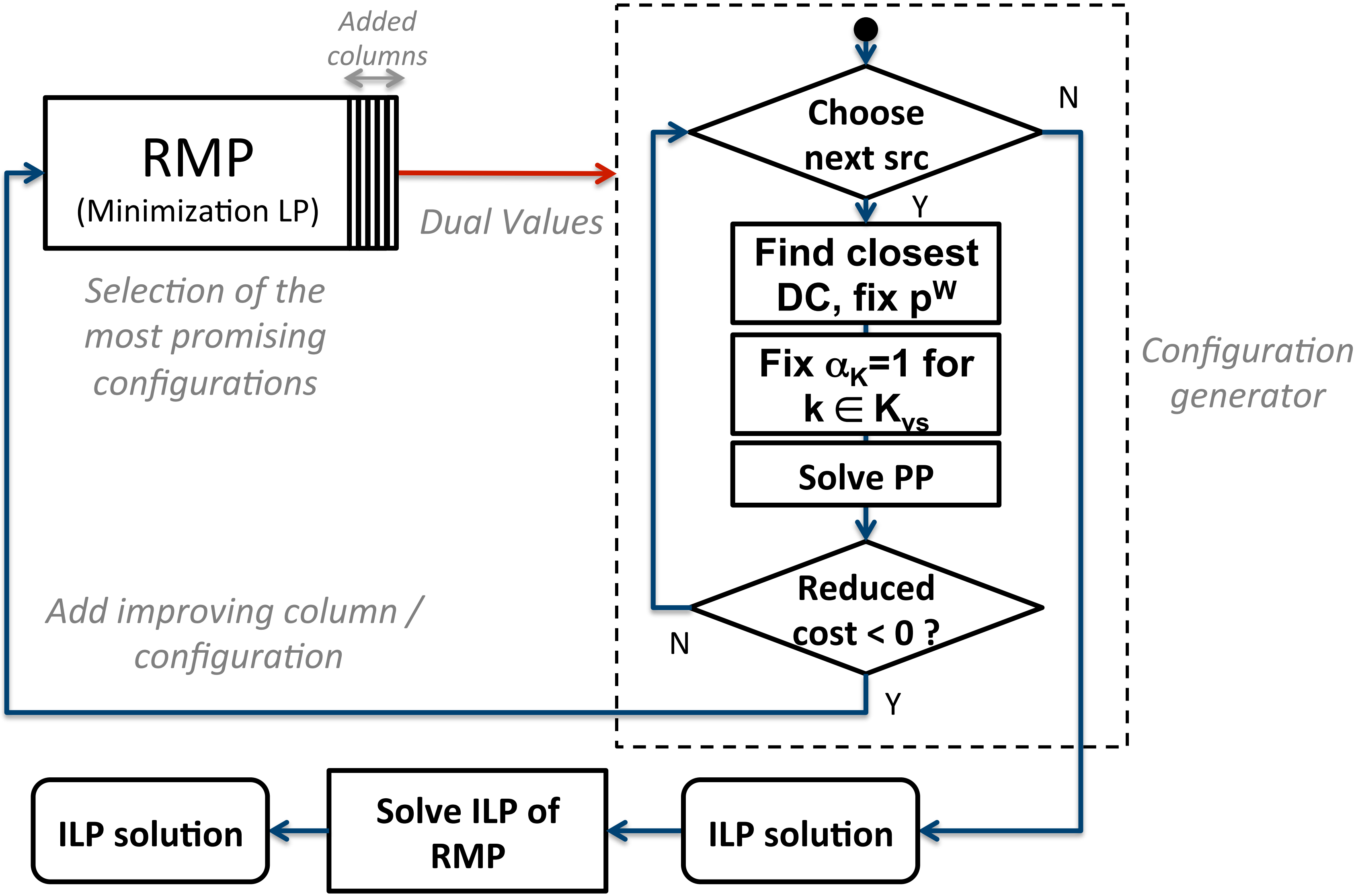
Solution: Column generation model

- Column generation idea:
  - Many different "configurations"
  - Start from a restricted subset of such "configurations"
  - Iteratively find additional configurations that reduce the cost:
    (1) Restricted Master Problem (RMP)
    (2) Pricing Problem (PP) to find new configs

- A configuration =
  - **Working** path
  - **Backup** path
  - 2 **sync** paths, one in each direction, between the primary & backup DCs
  - Set of services protected by the set of 4 paths

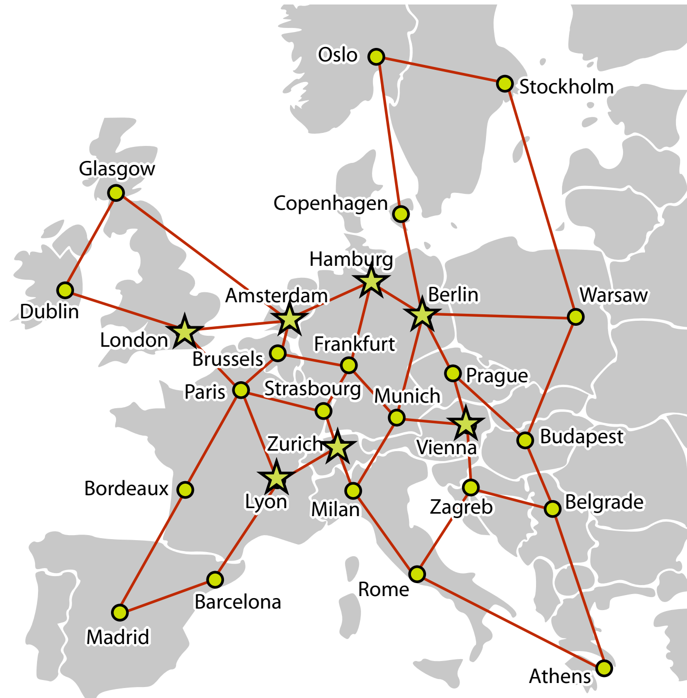


Column generation solution algorithm - Heuristic



Case study

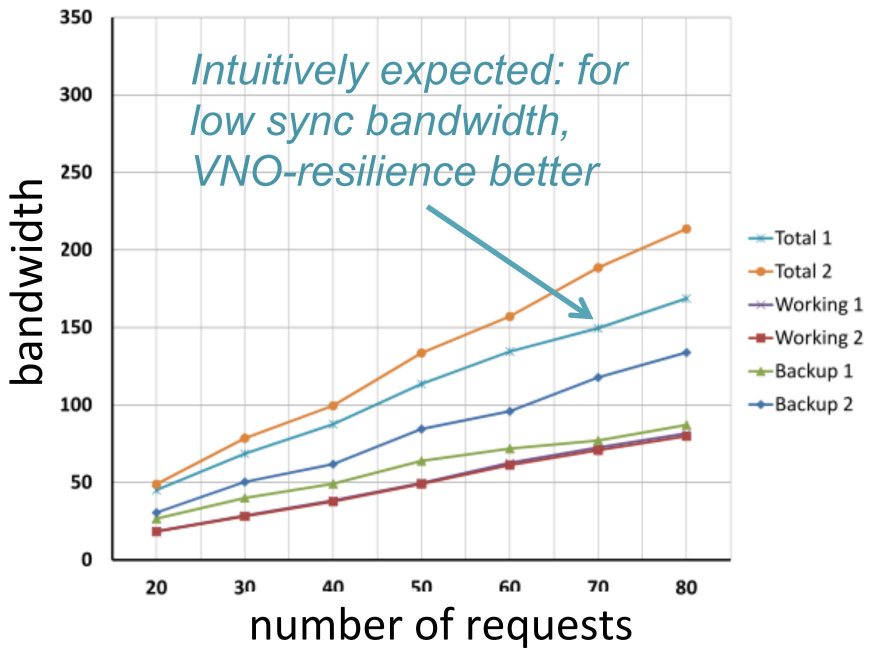
- Topology:
  - European network
  - 28 nodes and 41 bidir links (= 82 directed)
- Two choices of 4 data centers (DCs)
  - *Scattered evenly:*
    Lyon, Berlin, London, Vienna
  - *Pairs of close data centers:*
    Amsterdam, Hamburg, Lyon, Zurich



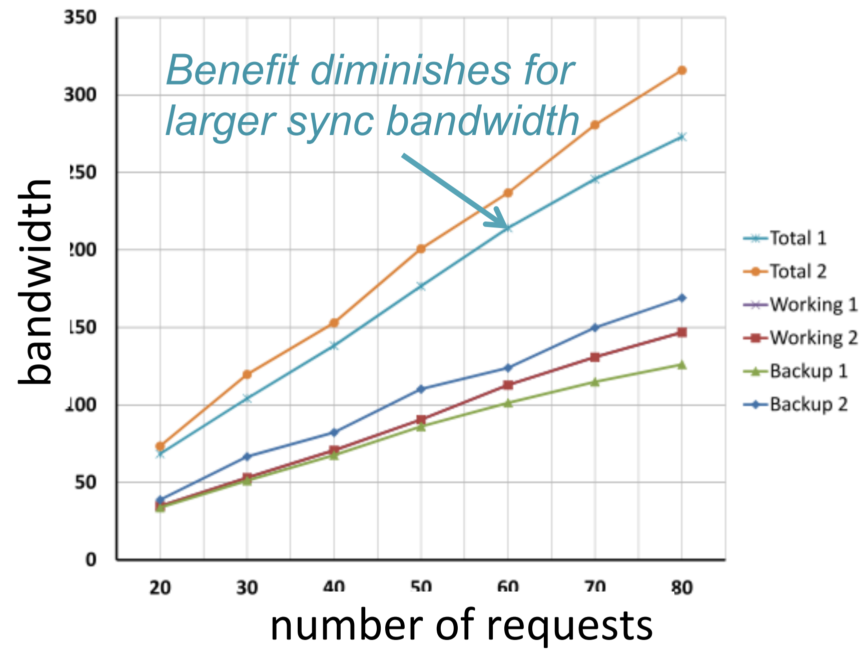
- Synchronization bandwidth fraction: $\delta_k = 0.1$ or 0.9
- Requests generated randomly with bandwidth in [0,1] wavelengths

Results: *evenly distributed DCs*

- DCs in Lyon, Berlin, London, and Vienna
- Model 1: **VNO-resilience**, Model 2: **PIP-resilience**



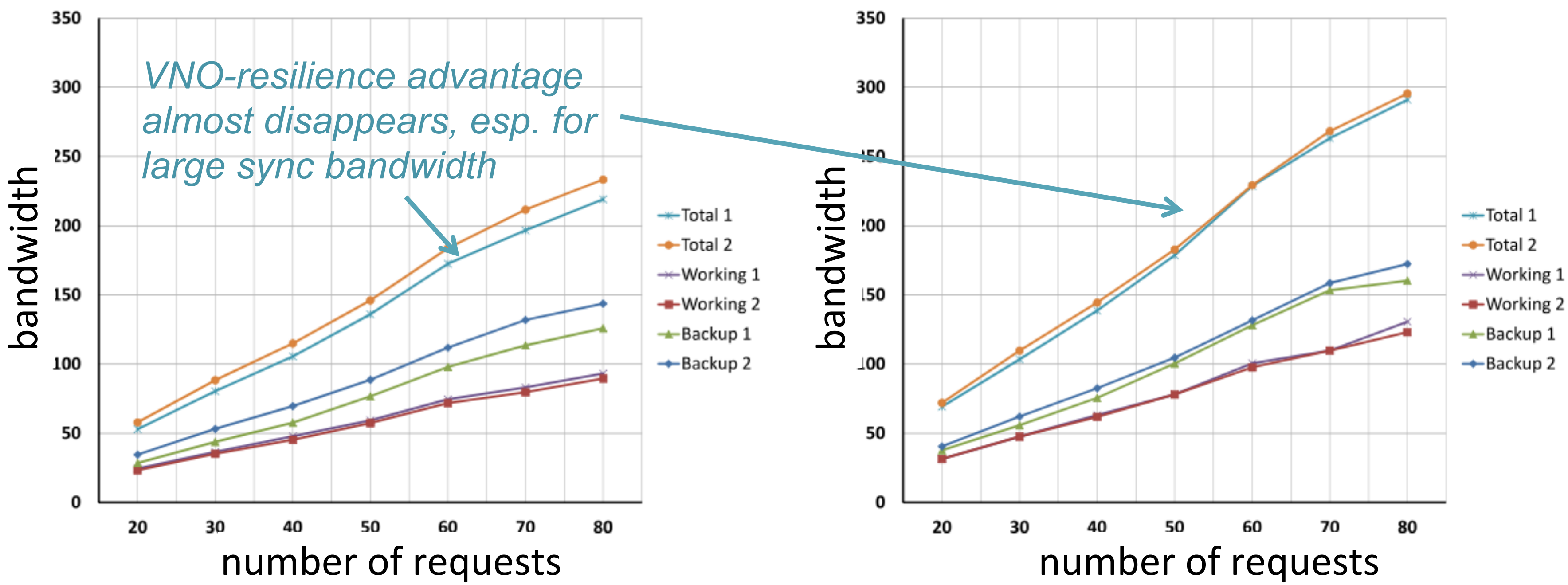
(a) $\delta_k = 0.1$



(b) $\delta_k = 0.9$

Results: ***close DC pairs***

- DCs in Amsterdam, Hamburg, Lyon, Zurich
- Model 1: **VNO-resilience**, Model 2: **PIP-resilience**



(a) $\delta_k = 0.1$ (b) $\delta_k = 0.9$

Conclusions

- Scalable column-generation based method for resilient VNet planning

- Intuition: VNO-resilience has lower physical network requirements than PIP-resilience
- **But…** relative advantage of VNO-resilience may be limited
  - When accounting for synchronization bandwidth between DCs
  - If DCs occur in nearby locations

- Future work:
  - Optimization of choice of DC locations?
  - Incorporate DC capacity constraints (e.g., limit max load)

Wrap-up

Take-away points

- Characteristics of cloud computing:
  - Anycast: User does not greatly care of exact location of servers
  - Virtualization: Cloud service provider may want isolation

- Dimensioning cloud networks:
  - Network + DC: locations of data centers can be optimized
  - Shared protection: exploit anycast through relocation
  - Failure-dependent (FD) vs -independent (FID) routing: limited advantage of FD for small number of data center locations
  - Virtualization: VNO vs PIP resilience: VNO savings can be limited for certain data center location strategies

Thank you … any questions?

Prof. Chris Develder

chris.develder@intec.ugent.be

Ghent University – iMinds