Online Strategies for Intra and Inter Provider Service Migration in Virtual Networks

or/and: How to migrate / allocate resources when you don't know the future?

Dushyant Arora
Marcin Bienkowski
Anja Feldmann
Gregor Schaffrath
Stefan Schmid

Co-authors:




Network virtualization architecture and prototype:

Anja Feldmann, Gregor Schaffrath, Stefan Schmid (T-Labs/TU Berlin)

VNet embeddings

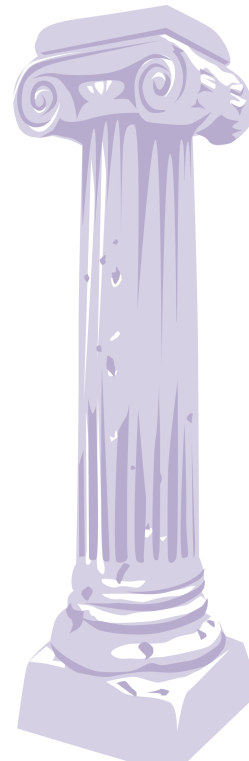
Guy Even and
Moti Medina (Tel Aviv Uni),
Carlo Fürst (TUB)

Service migration

Dushyant Arora (BITS) and
Marcin Bienkowski (Uni
Wroclaw)

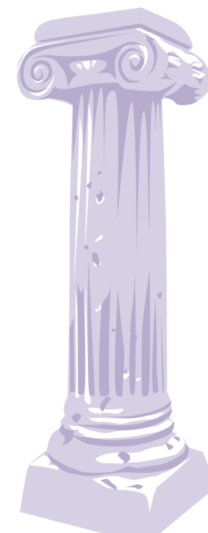
Economics

Arne Ludwig (TUB)



Implementation

Johannes Grassler



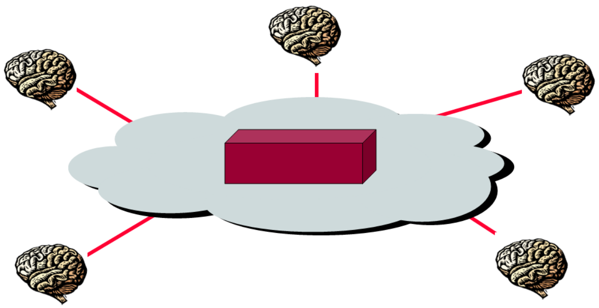
A joint project with Deutsche Telekom, TU berlin and NTT docomo:

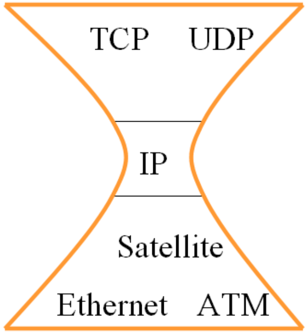
D. Jurca, A. Khan, W. Kellerer, K. Kozu and J. Widmer

Network Virtualization: Motivation

Success of the Internet architecture:

- This morning: continued success of IP protocol!
- nice: supports arbitrary applications „creativity on the edge"!
- even applicable to LANs and telephony





IP hourglass

But still: same ‚dial tone' optimal for everything?

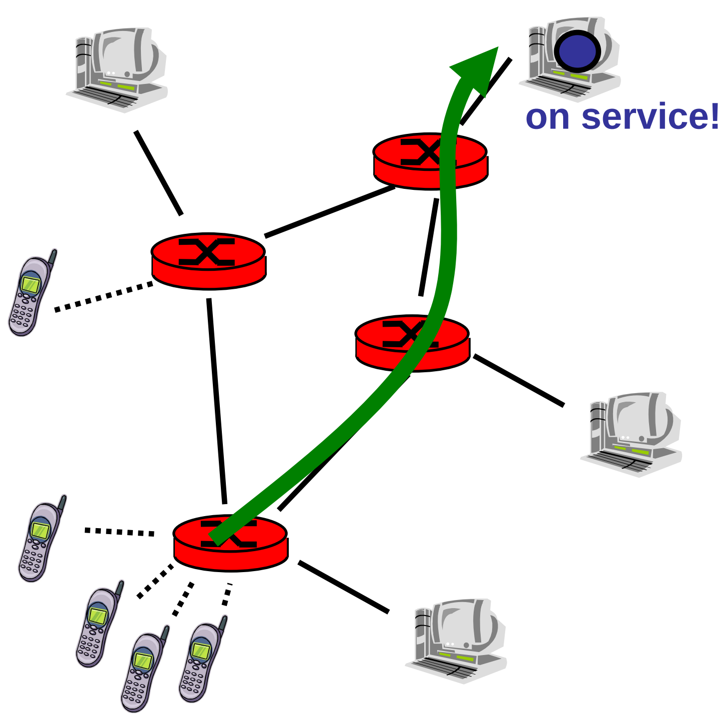
- innovation is only possible at lower and higher layers
- cannot experiment with different network cores (ossification)...
- different applications need different technologies: bulk data transfers vs social networking vs gaming vs live streaming... (distributions news vs social networking?)

Network Virtualization: High-level Concepts

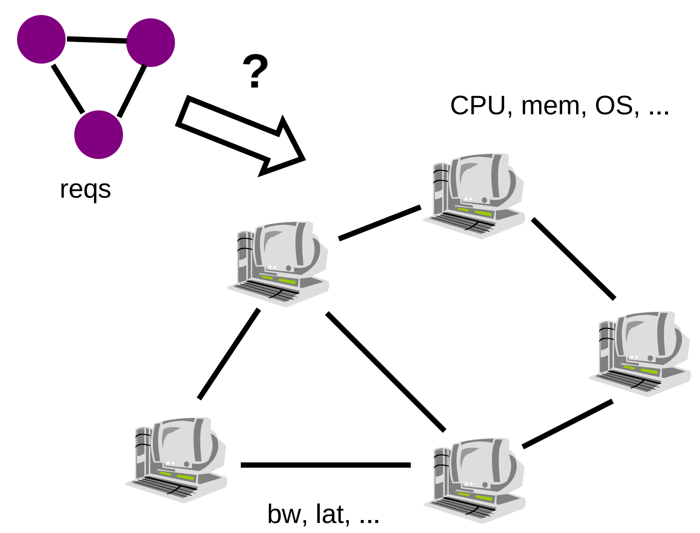
Virtualisation concept: decouples services from physical infrastructure (e.g., OpenFlow)

- Vision: on-demand, QoS, service-tailored VNets (e.g., 9-1-1 VNets, Internet itself), ...
- Also a way „to route money“ (accounting and responsibilities)?

Example 1: A mobile service provider can move services to locations where they are most useful: **QoS**



Example 2: Virtual networks (VNets) can be allocated where the least resources are used, or where most energy can be saved, or...: **flexibility in spec**



Previous Work: New Business Opportunities!

Actors in the Internet today: service providers and ISPs

- ISP: provide access (own infrastructure, rental, or combination), „connectivity service“ (e.g., Telekom, AT&T, ...)
- Service provider: offers services (e.g., Google)
- More roles exist today, often hidden in one company

Envisioned hierarchical business roles

PIP

Physical infrastructure provider (PIP):

owns and manages physical infrastructure („substrate“), supports network virtualization (e.g., GENI: no federation, one PIP only)

VNP

Virtual network provider (VNP):

assembles virtual resources from PIPs into virtual topology, makes negotiations, etc. (e.g., GENI clearinghouse)

VNO

Virtual network operator (VNO):

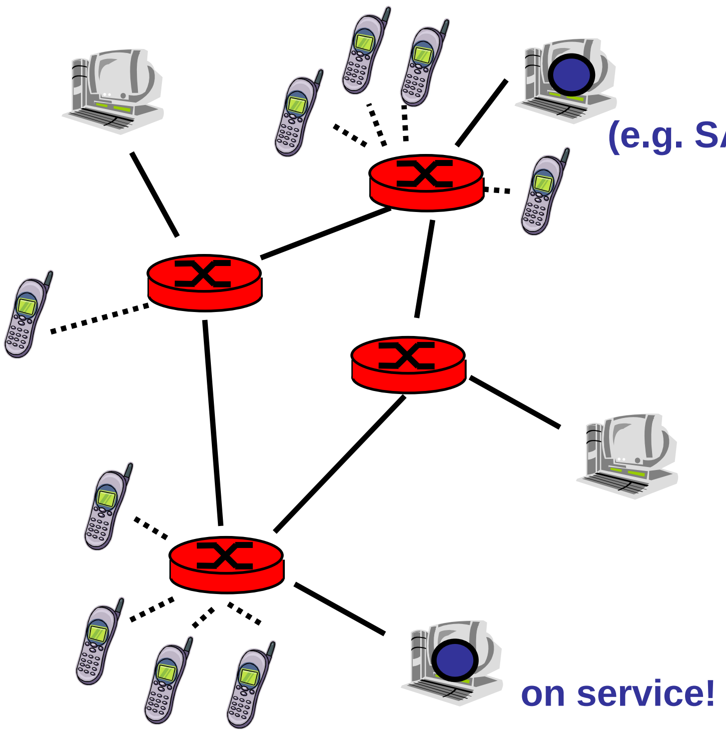
installation and operation of VNet according to SP needs, e.g., triggering cross-PIP migration, etc.

SP

Service provider (SP):

uses VNet to offer services (application or transport service)

This Paper: Online Service Migration for better QoS



When and where to move the service, to maximize QoS and taking migration cost into account?

Access pattern changes, e.g., due to mobility (commuter scenario), due to time-of-day effects (time-zone scenario), etc.

Dealing with Unpredictable Demand?

How to deal with dynamic changes (e.g., mobility of users, arrival of VNets, etc.)?

Online Algorithm

Online algorithms make decisions at time t without any knowledge of inputs / requests at times t'>t.

Competitive Ratio

Competitive ratio r,

r = Cost(ALG) / cost(OPT)

Is the price of not knowing the future!

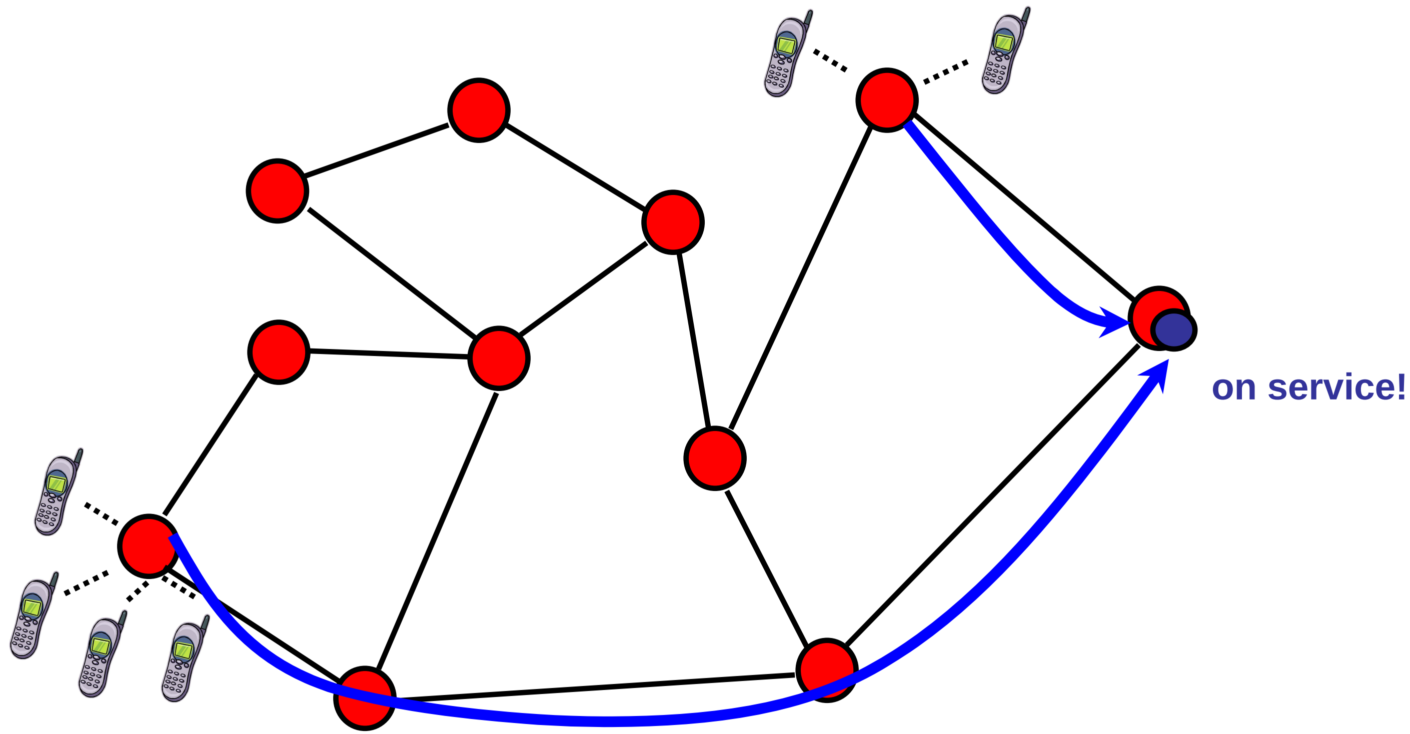
Competitive Analysis

An *r-competitive online algorithm* ALG gives a worst-case performance guarantee: the performance is at most a factor r worse than an optimal offline algorithm OPT!

In virtual networks, many decisions need to be made online: online algorithms and network virtualization are a perfect match! ☺

No need for complex predictions but still good! ☺

Online Service Migration



Assume: one service, migration cost *m* (e.g., service interruption cost), access cost 1 per hop (or sum of link delays along migration path).

When and where to move for *offline algorithm* or *optimal competitive ratio*?

Optimal Offline Algorithm

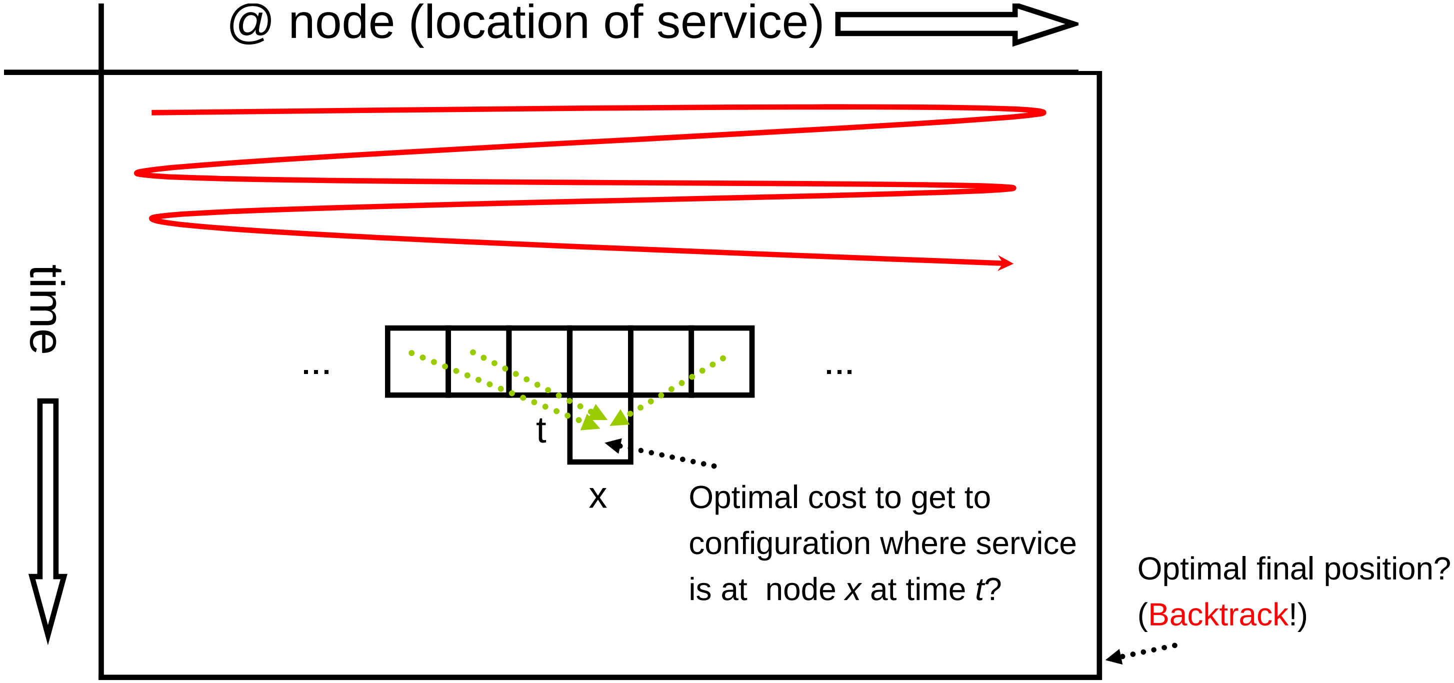
Can be computed using dynamic programming (optimal substructures)!

Filling out a for optimal server configuration (at node *u* at time *t*):

OPT

$$opt[u,t] = \min_{v \in V}\{opt[t-1][v] + MIG(v,u) + ACC(u,t)\}$$

Visualization:



Online Algorithm

Idea: Migrate to center of gravity when access cost at current node is as high as migration cost!

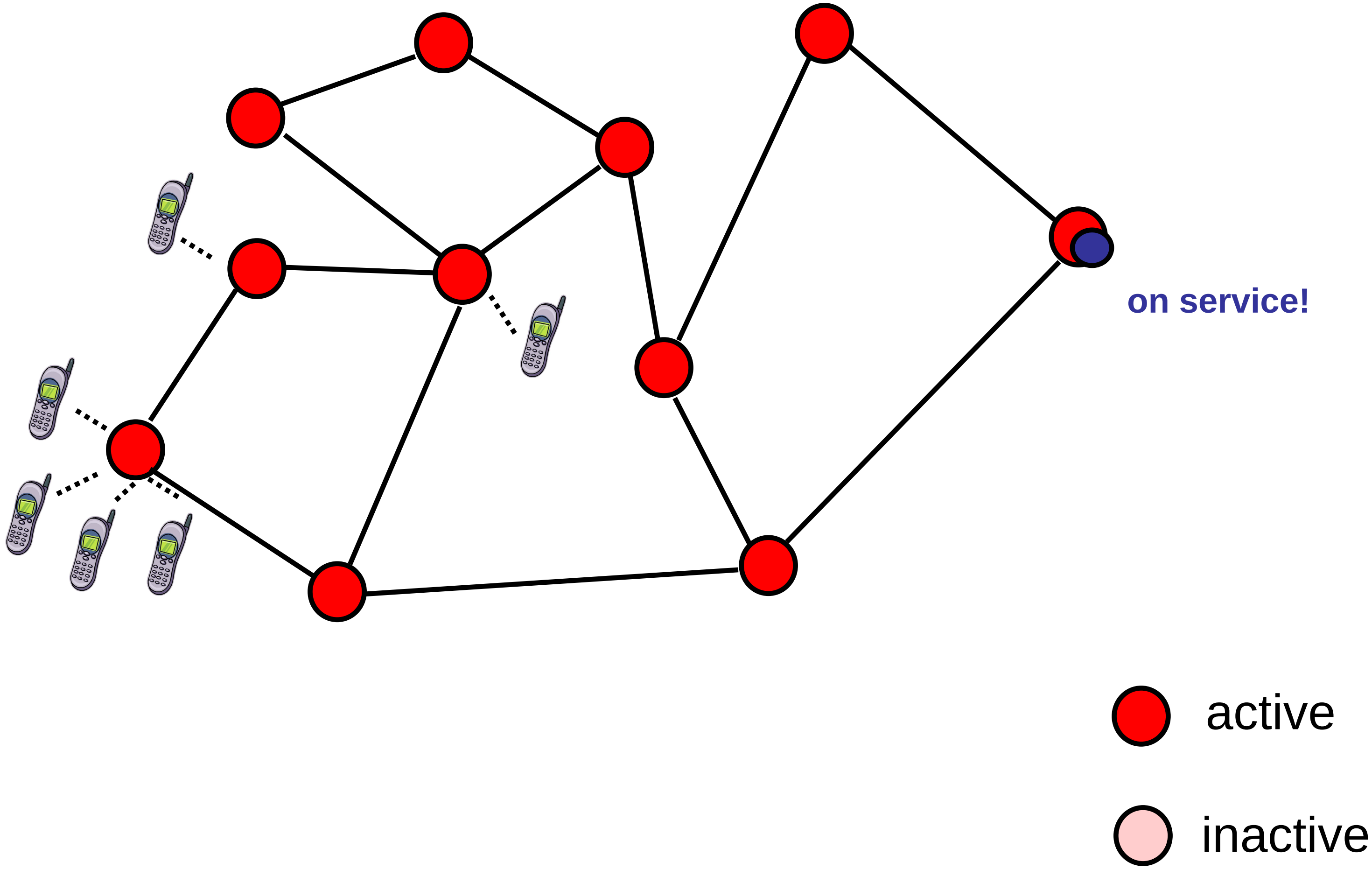
Time between two migrations: *phase*, multiple phases constitute an *epoch:*
In each phase go to center of nodes which are better!

Center of Gravity

For each node *v*, use COUNT(v) to count access cost if service was at *v* during entire ***epoch***. Call nodes v with COUNT(v) < *m*/40 ***active***. If service is at node *w*, a ***phase*** ends when COUNT(w)≥m: the service is migrated to the ***center of gravity*** of the remaining active nodes („center node“ wrt latency or hop distance). If no such node is left, the epoch ends.

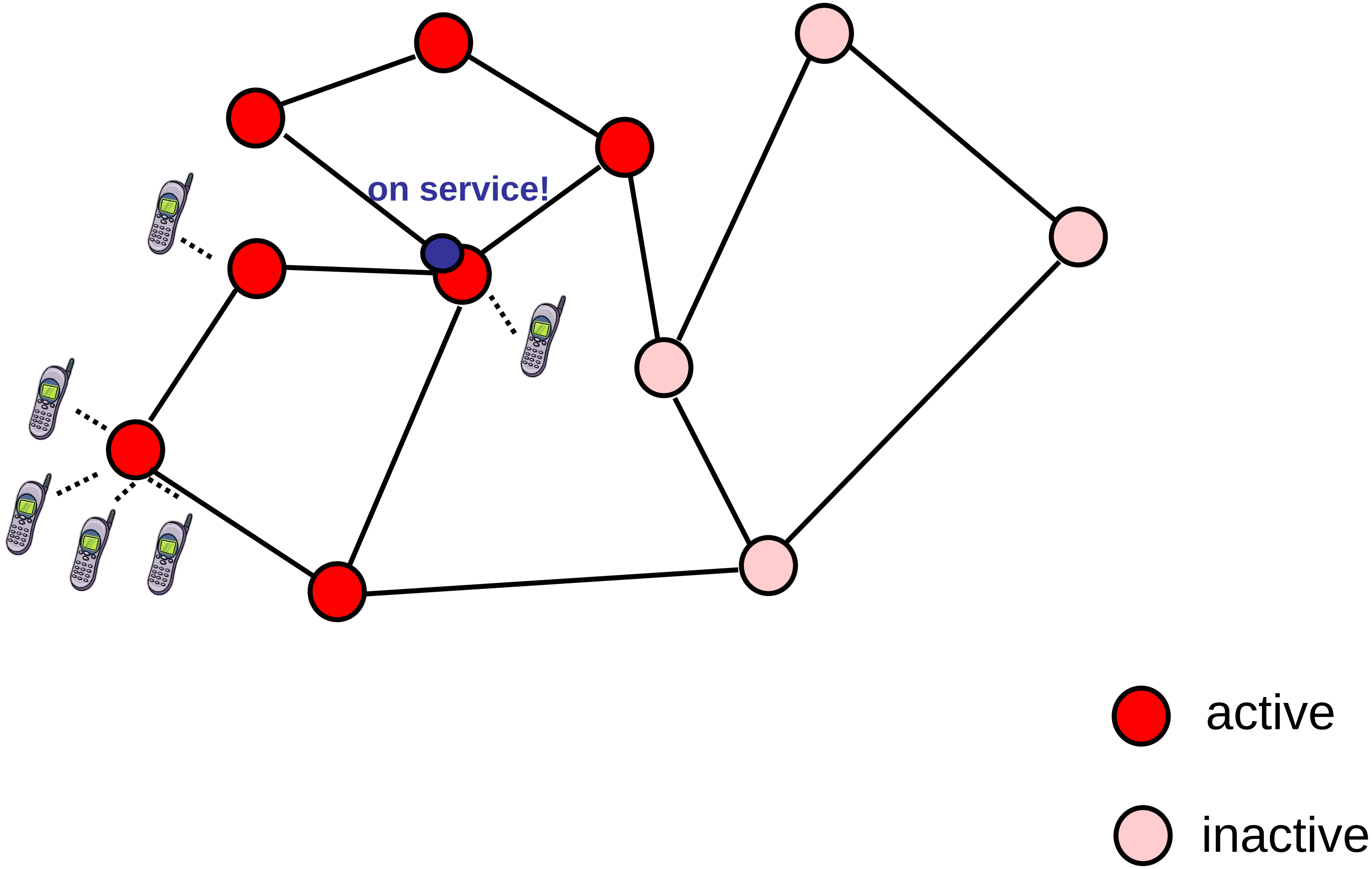
Online Algorithm: Visualization

Before phase 1:



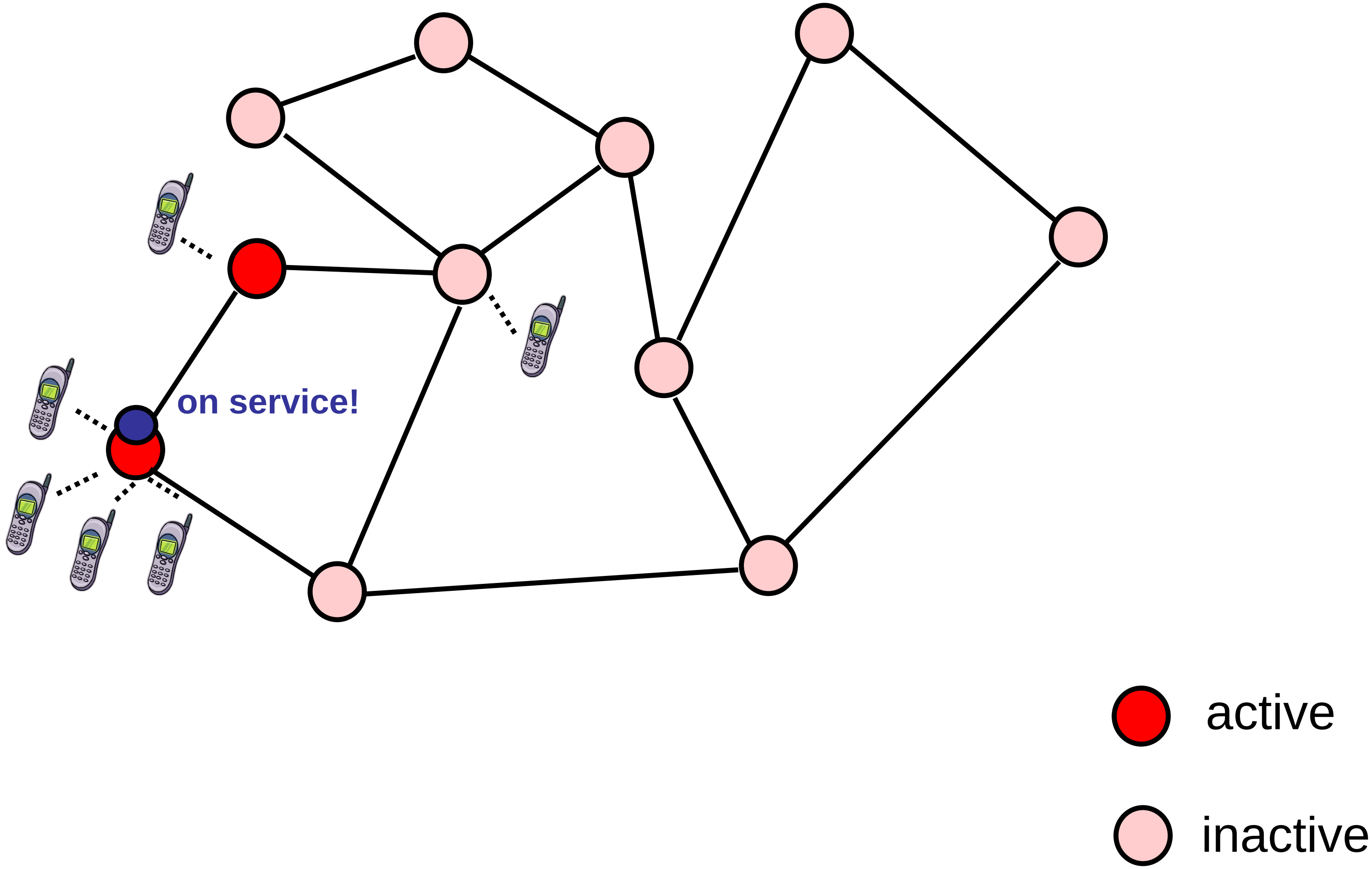
Online Algorithm: Visualization

Before phase 2:



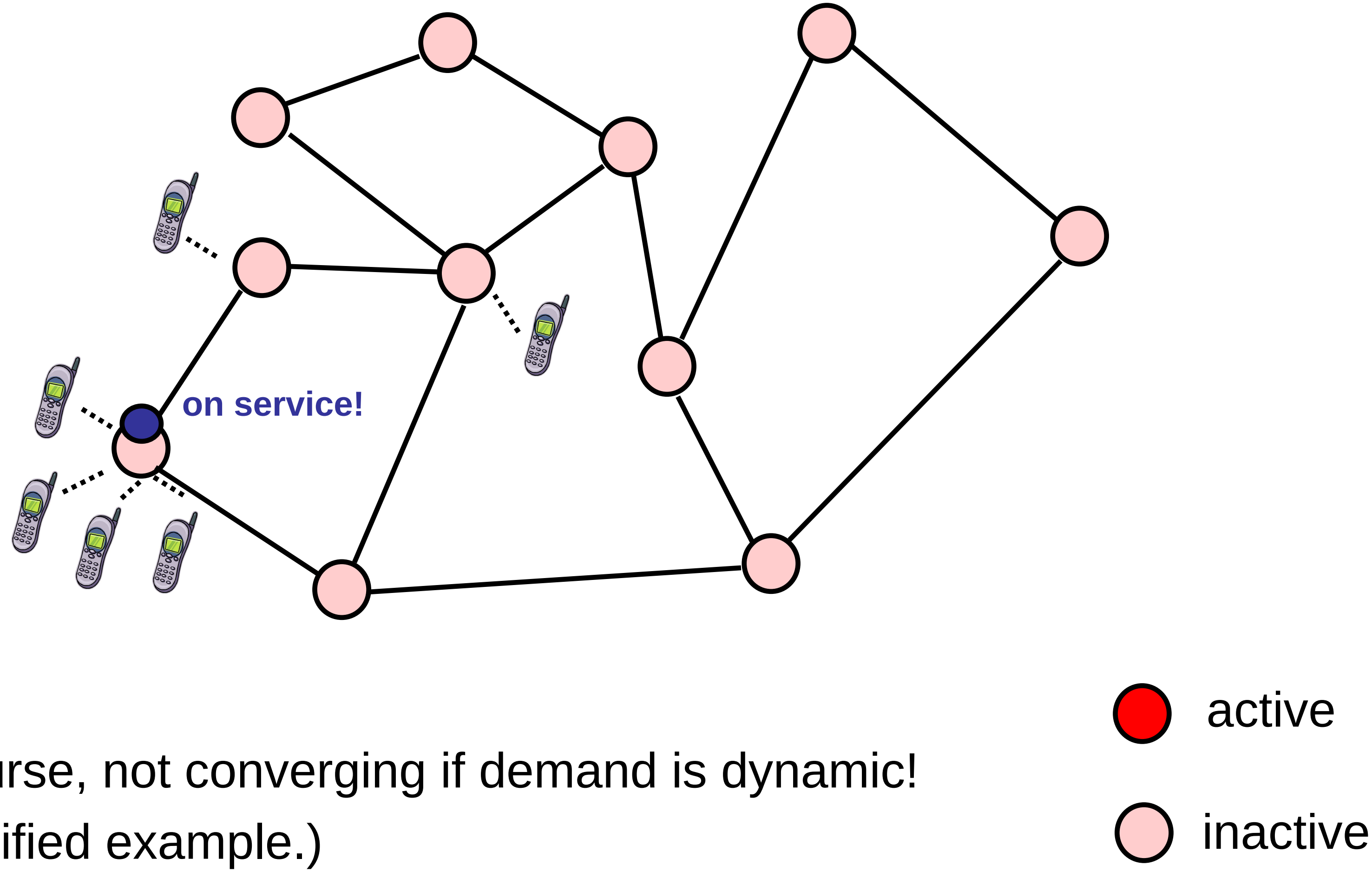
Online Algorithm: Visualization

Before phase 3:



Online Algorithm: Visualization

Epoch ends!



Of course, not converging if demand is dynamic!

(Simplified example.)

Online Algorithm: Analysis

Competitive analysis?

r = ALG / OPT $\leq$?

Lower bound cost of OPT:

In an epoch, each node has at least access cost ***m***, or there was a migration of cost ***m***.

Upper bound cost of ALG:

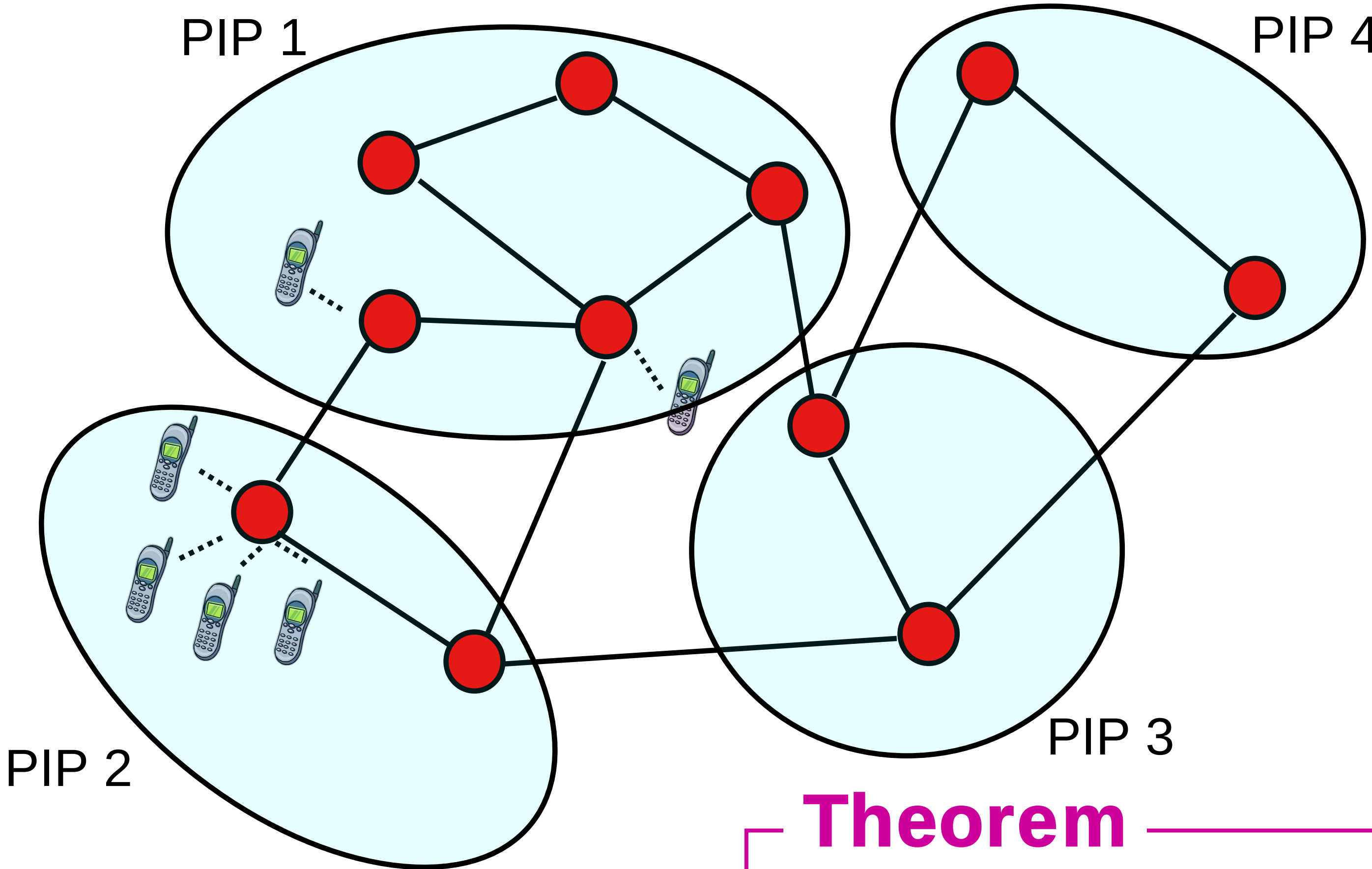
We can show that each phase has cost at most *2m* (access plus migration), and there are at most *log(m)* many phases per epoch!

Theorem

ALG is log(m) competitive!

Reality is more complex...: Multiple PIPs

Migration across provider boundary costs transit/roaming costs, detailed topology not known, etc.

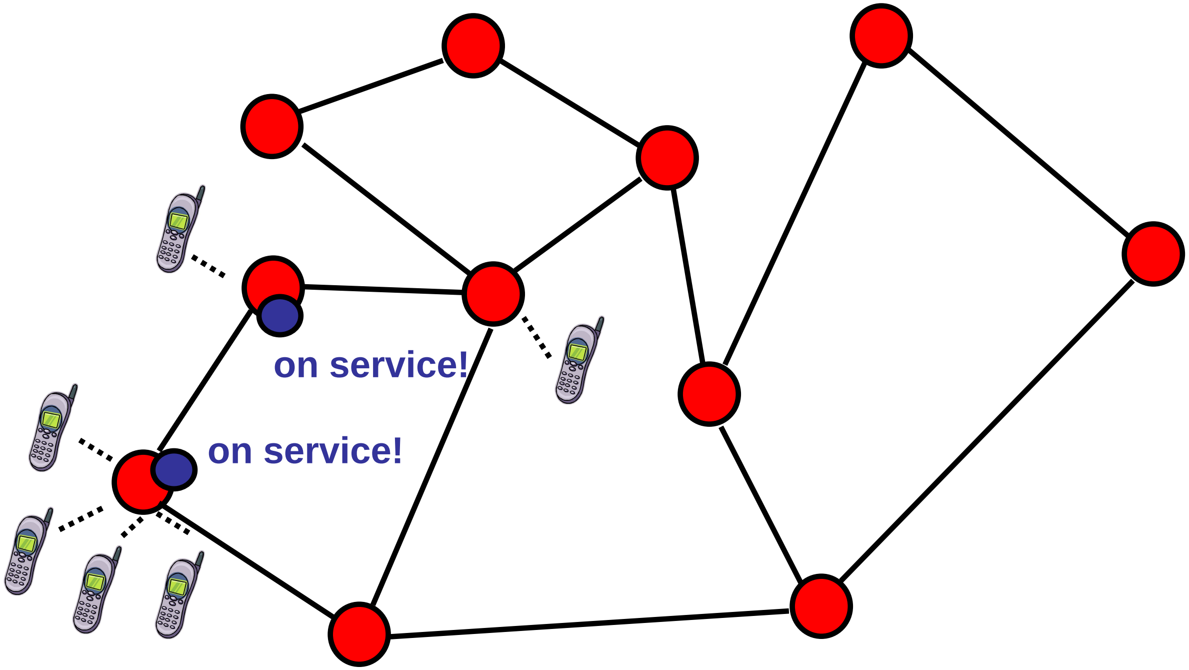


Theorem

Competitive ALGs still exist!

Reality is more complex...: Multiple Servers

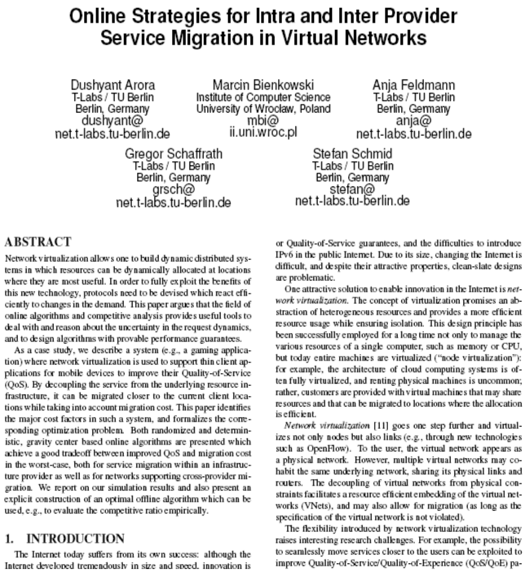
Multiple servers allocated and migrated dynamically depending on demand and load, servers have running costs, etc.



Theorem

Competitive ALGs still exist!

Summary of Theoretical Contribution



Online Strategies for Intra and Inter Provider Service Migration in Virtual Networks

Dushyant Arora, T-Labs / TU Berlin, Berlin, Germany, dushyant@net.t-labs.tu-berlin.de
Marcin Bienkowski, Institute of Computer Science, University of Wrocław, Poland, mbi@ii.uni.wroc.pl
Anja Feldmann, T-Labs / TU Berlin, Berlin, Germany, anja@net.t-labs.tu-berlin.de
Gregor Schaffrath, T-Labs / TU Berlin, Berlin, Germany, grsch@net.t-labs.tu-berlin.de
Stefan Schmid, T-Labs / TU Berlin, Berlin, Germany, stefan@net.t-labs.tu-berlin.de

ABSTRACT

Network virtualization allows one to build dynamic distributed systems in which resources can be dynamically allocated at locations where they are most useful. In order to fully exploit the benefits of this new technology, protocols need to be devised which react efficiently to changes in the demand. This paper argues that the field of online algorithms and competitive analysis provides useful tools to deal with and reason about the uncertainty in the request dynamics, and to design algorithms with provable performance guarantees.

As a case study, we describe a system (e.g., a gaming application) where network virtualization is used to support thin client applications for mobile devices to improve their Quality-of-Service (QoS). By decoupling the service from the underlying resource infrastructure, it can be migrated closer to the current client locations while taking into account migration cost. This paper identifies the major cost factors in such a system, and formalizes the corresponding optimization problem. Both randomized and deterministic, gravity center based online algorithms are presented which achieve a good tradeoff between improved QoS and migration cost in the worst-case, both for service migration within an infrastructure provider as well as for networks supporting cross-provider migration. We report on our simulation results and also present an explicit construction of an optimal offline algorithm which can be used, e.g., to evaluate the competitive ratio empirically.

1. INTRODUCTION

The Internet today suffers from its own success: although the Internet developed tremendously in size and speed, innovation is or Quality-of-Service guarantees, and the difficulties to introduce IPv6 in the public Internet. Due to its size, changing the Internet is difficult, and despite their attractive properties, clean-slate designs are problematic.

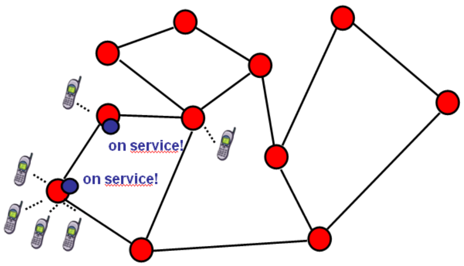
One attractive solution to enable innovation in the Internet is *network virtualization*. The concept of virtualization promises an abstraction of heterogeneous resources and provides a more efficient resource usage while ensuring isolation. This design principle has been successfully employed for a long time not only to manage the various resources of a single computer, such as memory or CPU, but today entire machines are virtualized ("node virtualization"): for example, the architecture of cloud computing systems is often fully virtualized, and renting physical machines is uncommon; rather, customers are provided with virtual machines that may share resources and that can be migrated to locations where the allocation is efficient.

Network virtualization [11] goes one step further and virtualizes not only nodes but also links (e.g., through new technologies such as OpenFlow). To the user, the virtual network appears as a physical network. However, multiple virtual networks may cohabit the same underlying network, sharing its physical links and routers. The decoupling of virtual networks from physical constraints facilitates a resource efficient embedding of the virtual networks (VNets), and may also allow for migration (as long as the specification of the virtual network is not violated).

The flexibility introduced by network virtualization technology raises interesting research challenges. For example, the possibility to seamlessly move services closer to the users can be exploited to improve Quality-of-Service/Quality-of-Experience (QoS/QoE) pa-

Cost model

- migration cost: service interruption (duration: depends on bandwidth)
- access costs: latency (triangle inequality)
- roaming costs: inter-provider migration



inequalities, we obtain

$$
\begin{aligned}
\sum_{u \in V'} d(\mathcal{G}(V'), u) &= \sum_{u \in V''} d(\mathcal{G}(V'), u) + \sum_{u \in V' \setminus V''} d(\mathcal{G}(V'), u) \\
&\geq \sum_{u \in V''} \left[d(\mathcal{G}(V'), r) - d(u, r) \right] \\
&\quad + \sum_{u \in V' \setminus V''} d(\mathcal{G}(V'), u) \\
&> (1 - \lambda_1) \cdot |V''| \cdot F + \sum_{u \in V' \setminus V''} d(\mathcal{G}(V'), u) \\
&= \frac{4}{5} \cdot |V''| \cdot F + \sum_{u \in V' \setminus V''} d(\mathcal{G}(V'), u) \ ,
\end{aligned}
$$

because $d(\mathcal{G}(V'), r) = F$ and $d(u, r) \leq \lambda_1 \cdot F$, and by substituting $\lambda_1 = 1/5$. On the other hand, note that $|V' \setminus V''| \leq |V'|/4 \leq |V''|/3$ and

$$
\begin{aligned}
\sum_{u \in V'} d(\xi, u) &= \sum_{u \in V''} d(\xi, u) + \sum_{u \in V' \setminus V''} d(\xi, u) \\
&< 2\lambda_1 \cdot |V''| \cdot F \\
&\quad + \sum_{u \in V' \setminus V''} \left[d(\xi, r) + d(r, \mathcal{G}(V')) + d(\mathcal{G}(V'), u) \right] \\
&< 2\lambda_1 \cdot |V''| \cdot F + |V' \setminus V''| \cdot (1 + \lambda_1) \cdot F \\
&\quad + \sum_{u \in V' \setminus V''} d(\mathcal{G}(V'), u)
\end{aligned}
$$

Contribution

- online and offline algorithms for various scenarios
- take-aways: under what dynamics is flexibility better?

On the Benefit of Flexibility: Dynamics Scenarios*

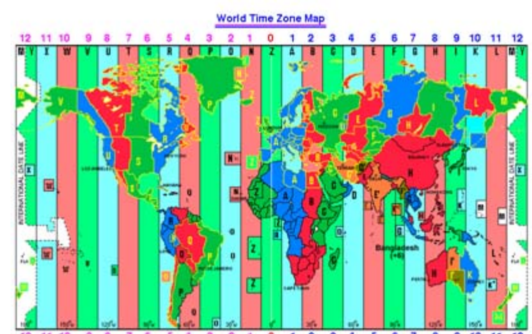
Commuter Scenario

Dynamics due to mobility: requests cycle through a 24h pattern: in the morning, requests distributed widely (people in suburbs), then focus in city centers; in the evening, reverse.

Time Zone Scenario

Dynamics due to time zone effects: request originate in China first, then more requests come from European countries, and finally from the U.S.

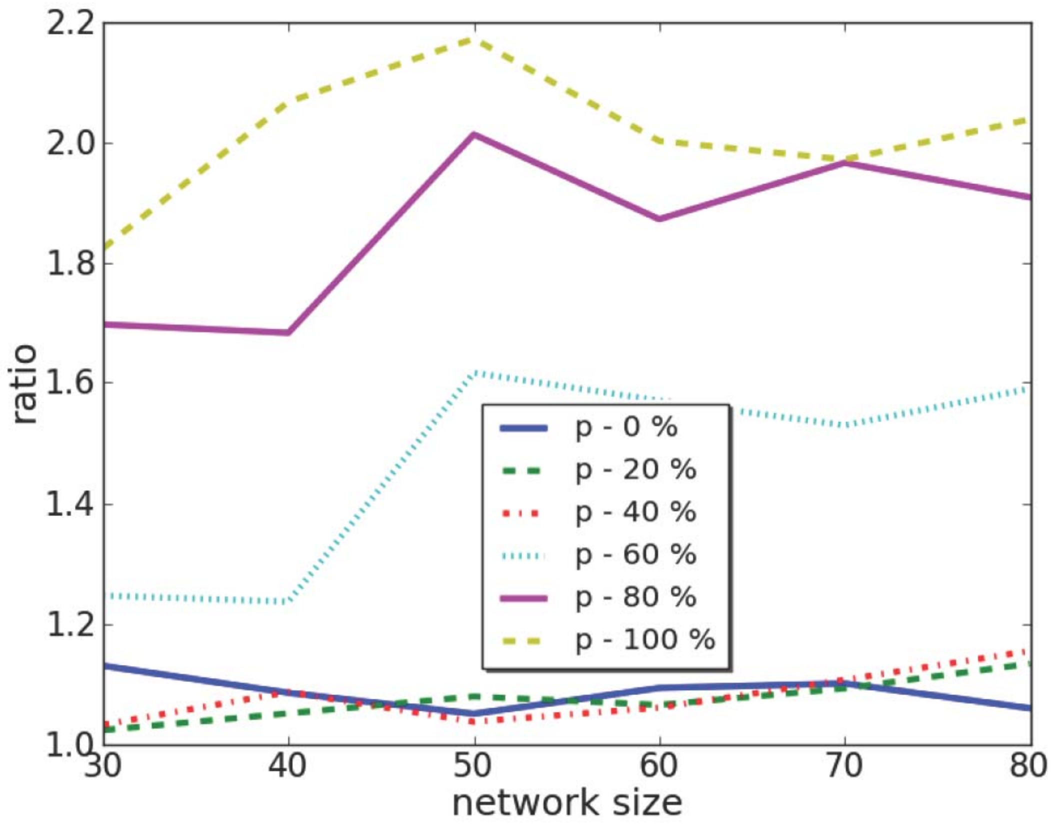
* Predictable scenarios, but we do not exploit. Reality less predictable!



Static Algorithm

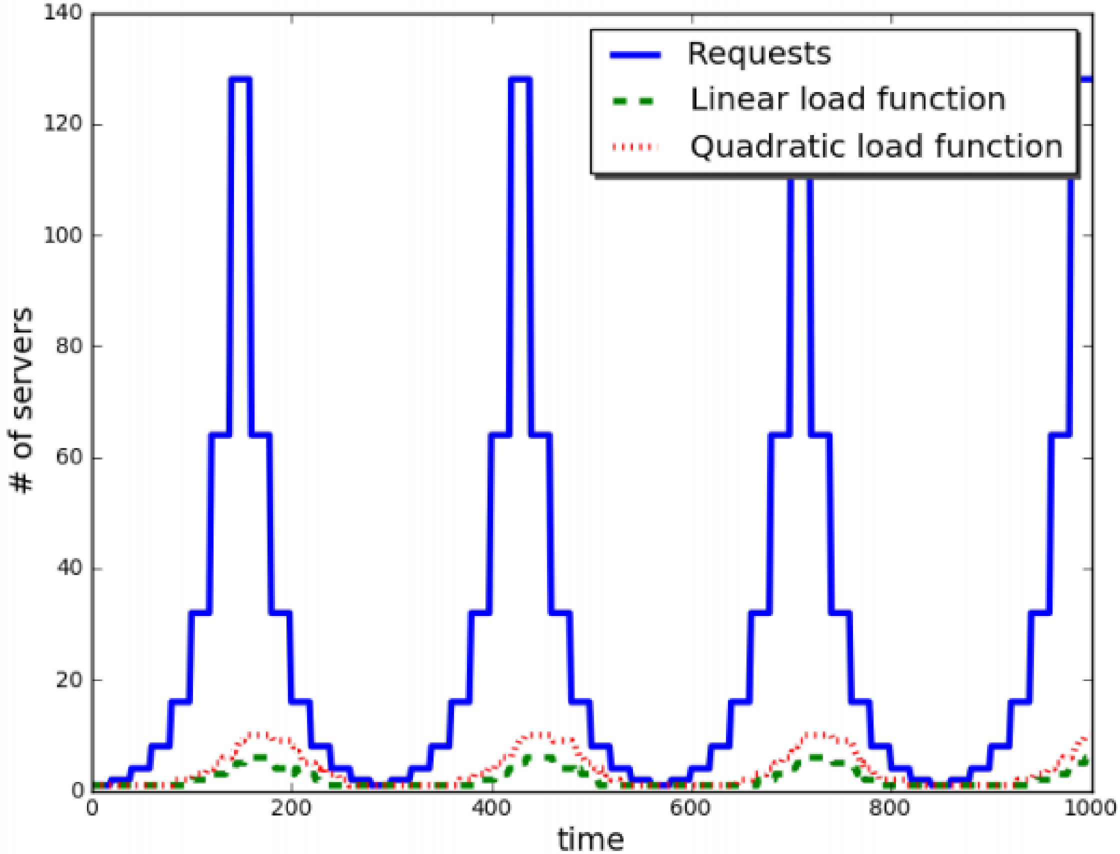
Algorithm which uses optimal static server placements for a given request seq.

Time Zone Scenario with Different Request Correlations



Ratio relatively low and not increasing much in „average case".
Higher correlation increases ratio.

Extensions to Multi-Server Scenarios (Hot-ICE 2011)



Increasing demand triggers creation of additional servers (more for faster growing load functions): have running costs (will be shut down again), maybe standby for faster/cheaper startup.

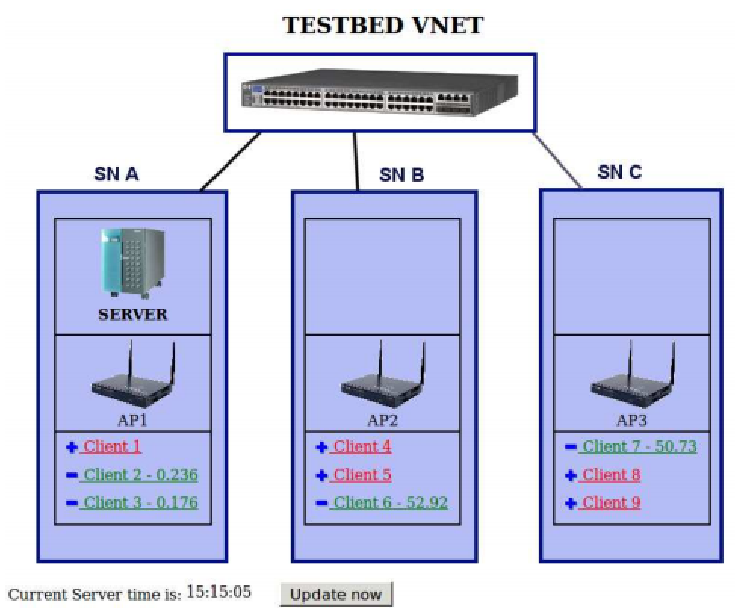
Conclusion and Takeaways

- Flexible server allocation for network virtualization and beyond (e.g., cloud): generalized model for a challenging problem

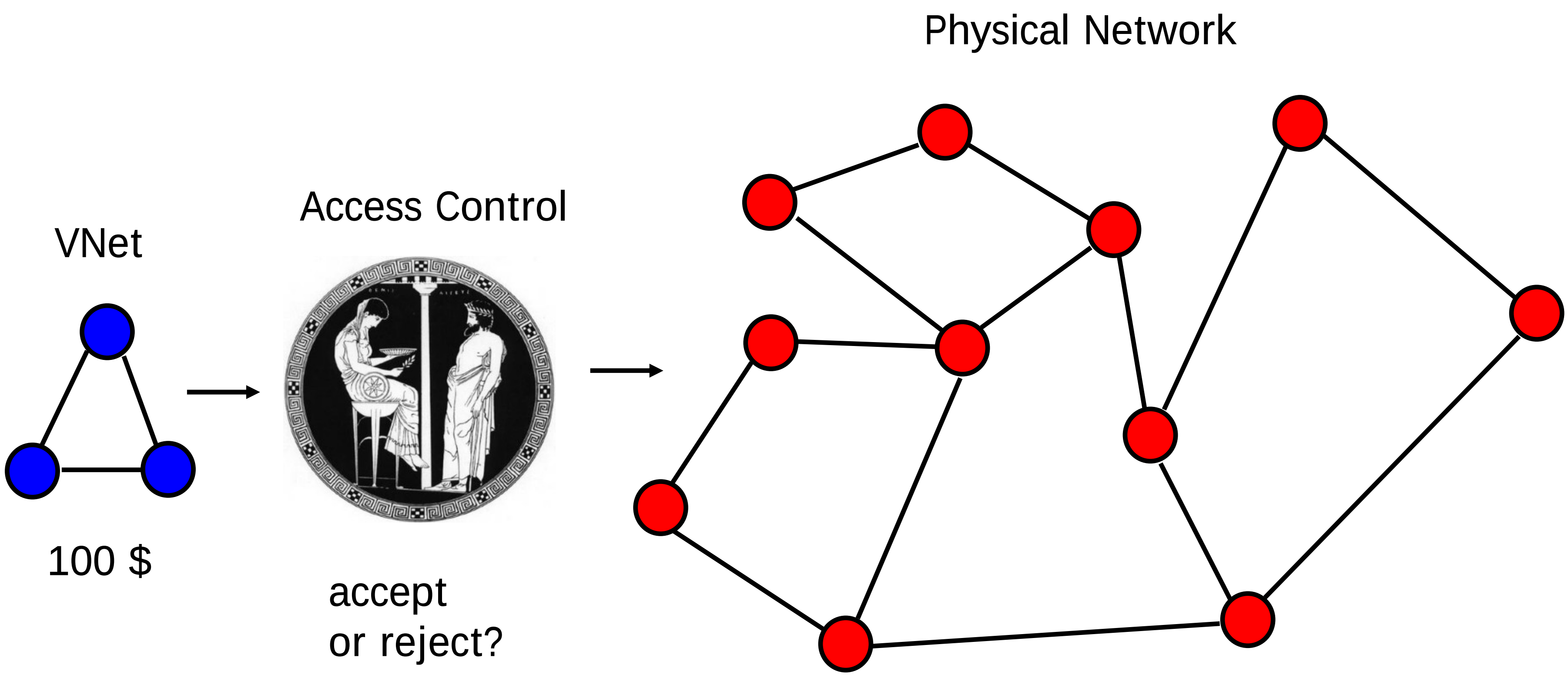
- Online perspective: algorithms have to decide without knowing the future; relevant for many aspects of network virtualization

- When useful? Depends on dynamics!

- Streaming migration demonstrator for our network virtualization prototype (VLAN based):

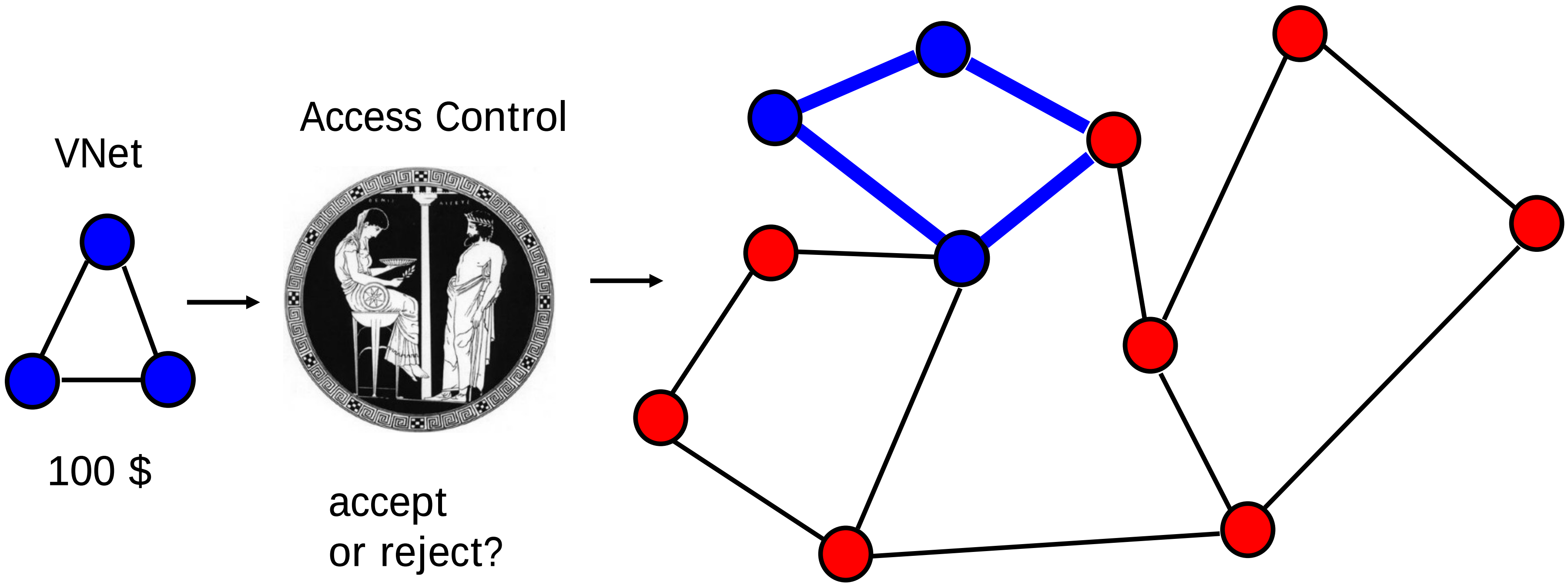


Outlook: Competitive VNet Embedding



Outlook: Competitive VNet Embedding

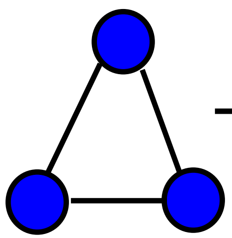
Cheap realization => Yes!



Outlook: Competitive VNet Embedding

Expensive => No!

VNet

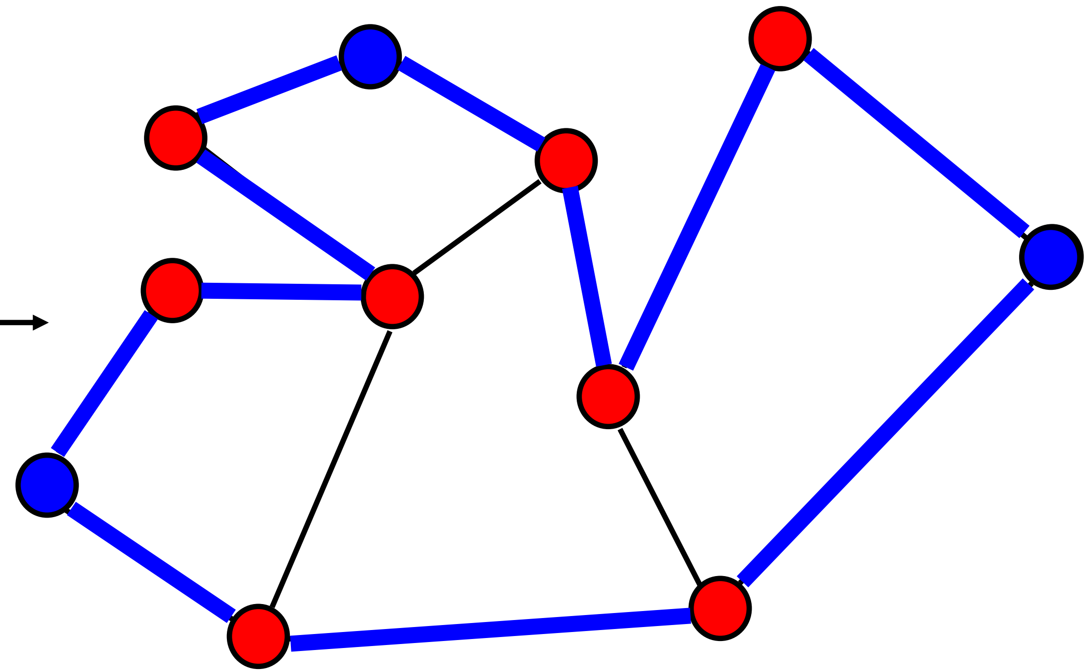


100 $

Access Control

accept
or reject?

Physical Network



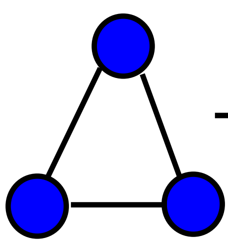
Outlook: Competitive VNet Embedding

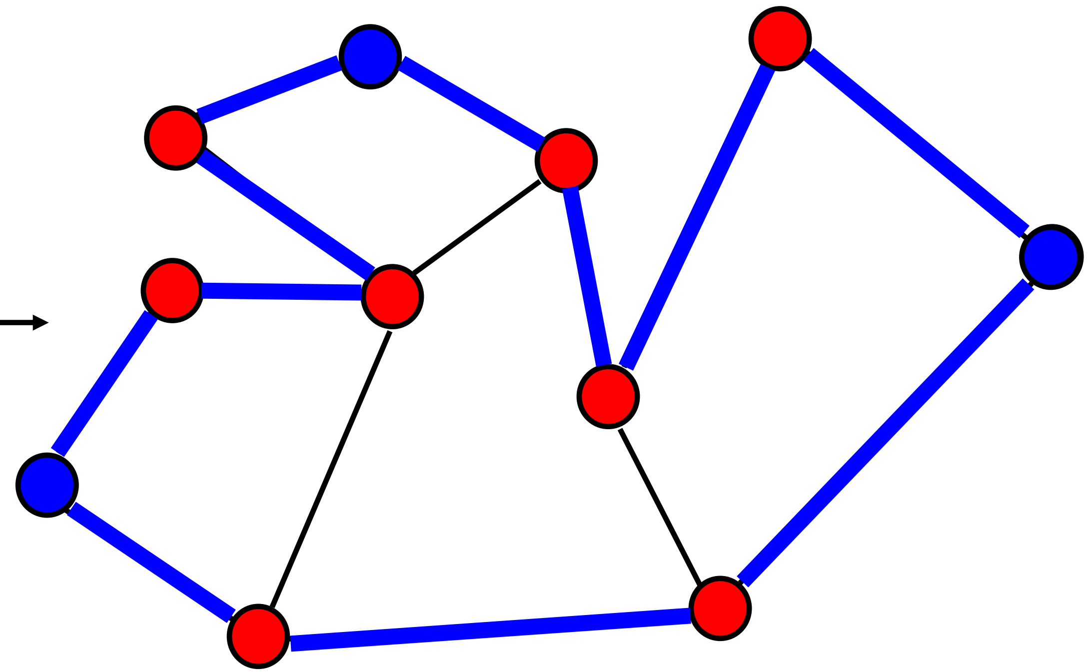
Expensive => No!

Physical Network

Access Control

VNet





100 $

accept
or reject?

Online primal-dual framework by Buchbinder and Naor: log competitive!

Thank you!

Further reading: Project website!

http://www.net.t-labs.tu-berlin.de/~stefan/virtu.shtml

Comparison to Related Work

- Conservative online perspective on resource management: no predictions possible, but with worst-case guarantees

- Detailed costs model for VNet application (multiple PIPs with transit costs, costs depending on scenario: shared NFS, etc.)

- Allows to study the „use of flexibility" (compared to static algorithms)

- Like dynamic facility location problems where additional facilities can be created, migrated and closed (at non-zero cost) and where facilities have running costs and access costs that depend on load

- Often a special case of metrical task systems but sometimes better bounds can be obtained for the more specific model!