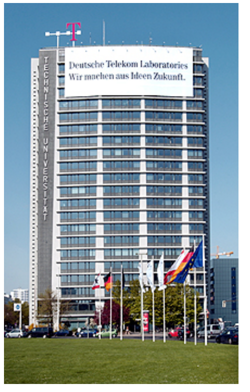
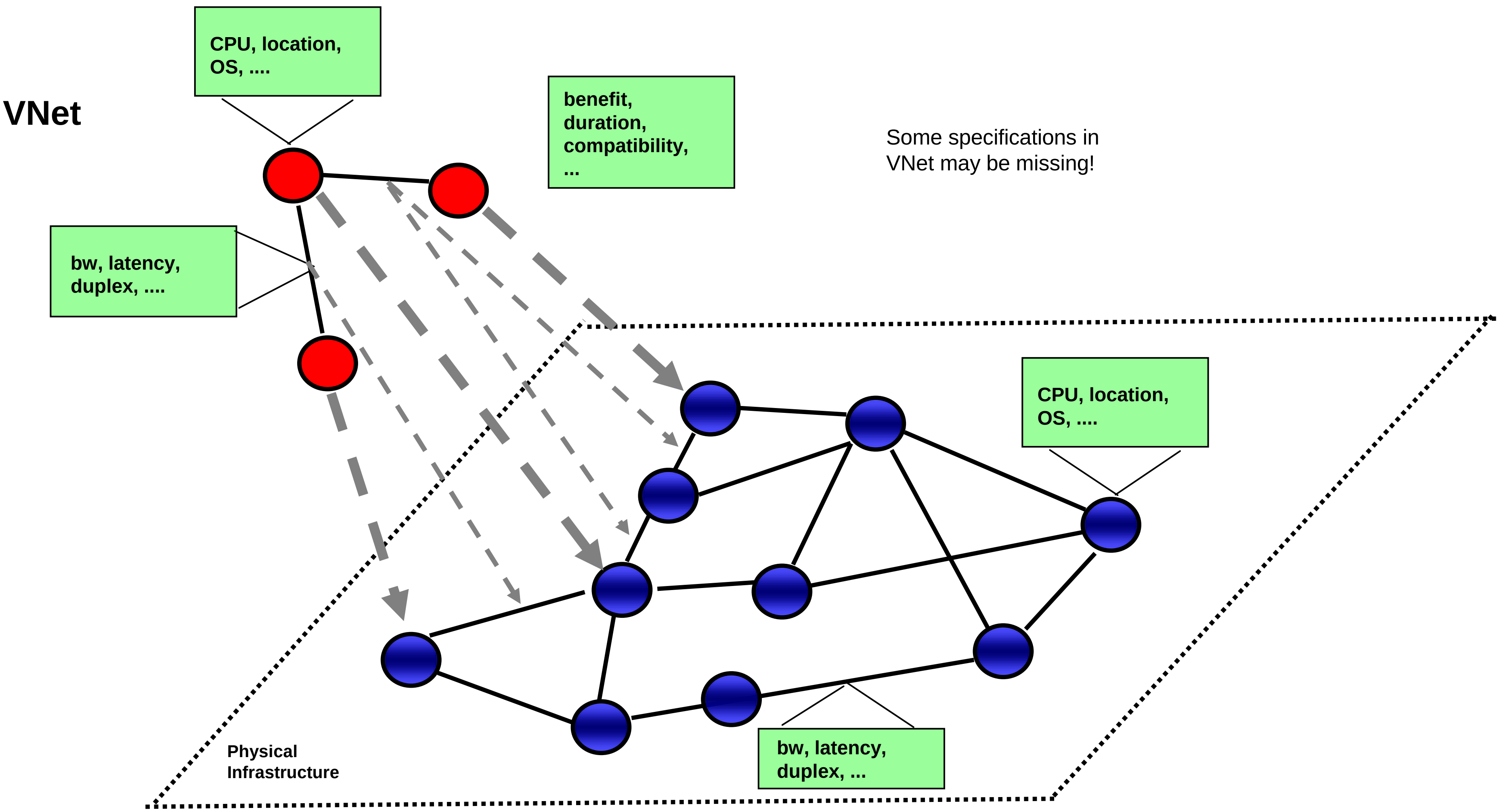
# Competitive and Deterministic Embeddings of Virtual Networks


*Guy Even (Tel Aviv Uni)*
*Moti Medina (Tel Aviv Uni)*
*Gregor Schaffrath (T-Labs Berlin)*
***Stefan Schmid (T-Labs Berlin)***

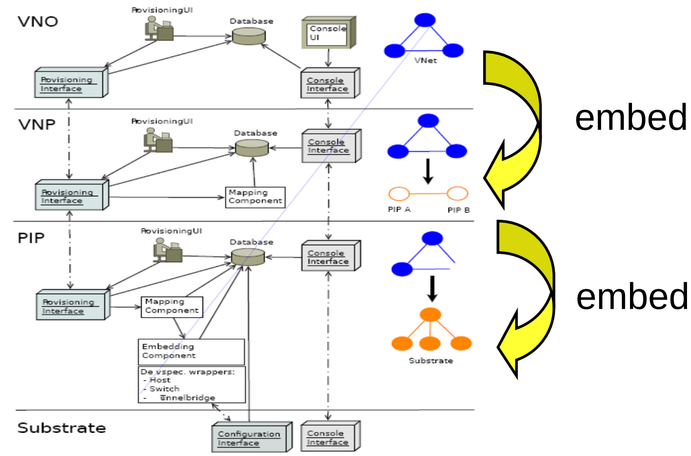



# The Virtual Network Embedding Problem



# Context: VNet Prototype Architecture at T-Labs (& DoCoMo)

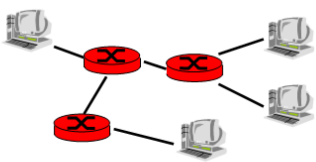

Streaming server moves closer to (mobile) users, yields better QoS (less important VNets can be migrated away):http://www.youtube.com/watch?v=llJce0F1zHQ




## Prototype

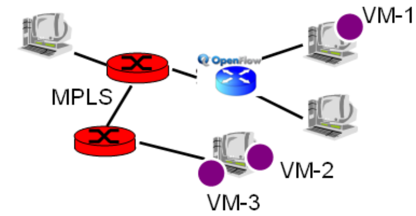
- Layer 2 virtual networks w/ QoS (based on VLANs)
- Testbed TU Berlin + DoCoMo Eurolab Munich (multi-provider)
- Hierarchical provider model: economic roles communicate requirements by contract by contract...
- ... virtual networks get mapped in multiple stages!
- So far: Physical Infrastructure Provider and Virtual Network Operator
- Seamless migration (VNet configuration stays the same!), embedding

# It‘s a virtual world...: What are virtual networks?



## Virtualization

Attractive design principle: virtualization abstracts heterogeneous resources and allows for resource sharing




| Node virtualization | Link virtualization |
|---|---|
| - success over last years: revamped server business!<br>- Xen, VMWare...<br>- today‘s clouds hardly offer any physical resources | - past and current trend<br>- router virtualization (Cisco, Juniper, ...)<br>- MPLS / GMPLS (too?) ubiquituous<br>- same holds for VPNs<br>- hype of SDN (split) architectures |

**Idea: Combine and generalize the two to entire virtual networks!**

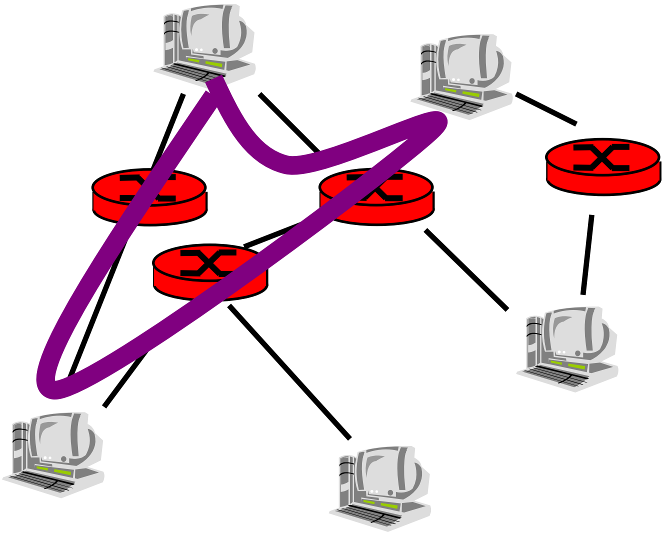
# The next step: offer entire virtual networks!

## Virtual Networks (VNets)

Basic unit: set of links (graph) rather than single links! VNets co-habit same physical network (substrate), but have different characteristics, protocol stack, ... (beyond IP). E.g., VNets may provide QoS guarantees on links (unlike VPNs)!

**Use Cases:**

- Traveling agents set up video conference
- Start-up company / academic institution experiments with new transmission protocols
- Data centers: performance guarantees despite dynamic demand to meet deadlines (improves service, safes bandwidth and energy, e.g., Octopus @ SIGCOMM 2011)
- Wireless access
- Migration for network maintenance or resilience



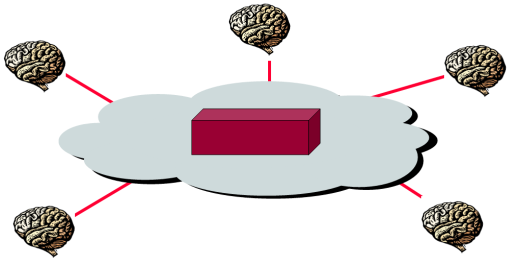
**Advantages:**

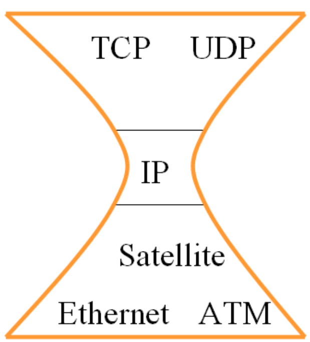
- On demand and short notice: as long as needed
- QoS guarantees and isolation
- No need to buy/build infrastructure, ...

# VNets can help to innovate the Internet!

## Ossification in today's Internet:

- innovation is only possible at lower and higher layers...
- ... but we cannot experiment with different network cores: Layer 3 is („ossification")...
- different applications need different technologies: *bulk data transfers* vs *social networking* vs *gaming* vs *live streaming*... (distributions news vs social networking?)





IP hourglass

## Network virtualization can help!

- experiment with different technologies
- service-tailored, but co-habit same substrate network
- enabling technologies: OpenFlow, VLANs, ...

# Virtu Prototype Architecture: Challenges

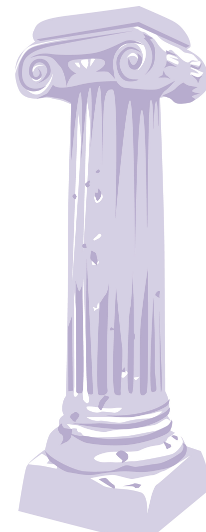
Anja Feldmann, Gregor Schaffrath, Stefan Schmid (T-Labs/TU Berlin)

Implementation

**Algorithms**

Economics

(e.g., paper by Pan Hui
in this session)

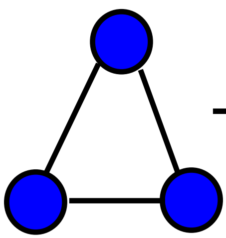


A joint project with Deutsche Telekom, TU berlin and NTT docomo:

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

# This Paper: Competitive VNet Embedding

VNet

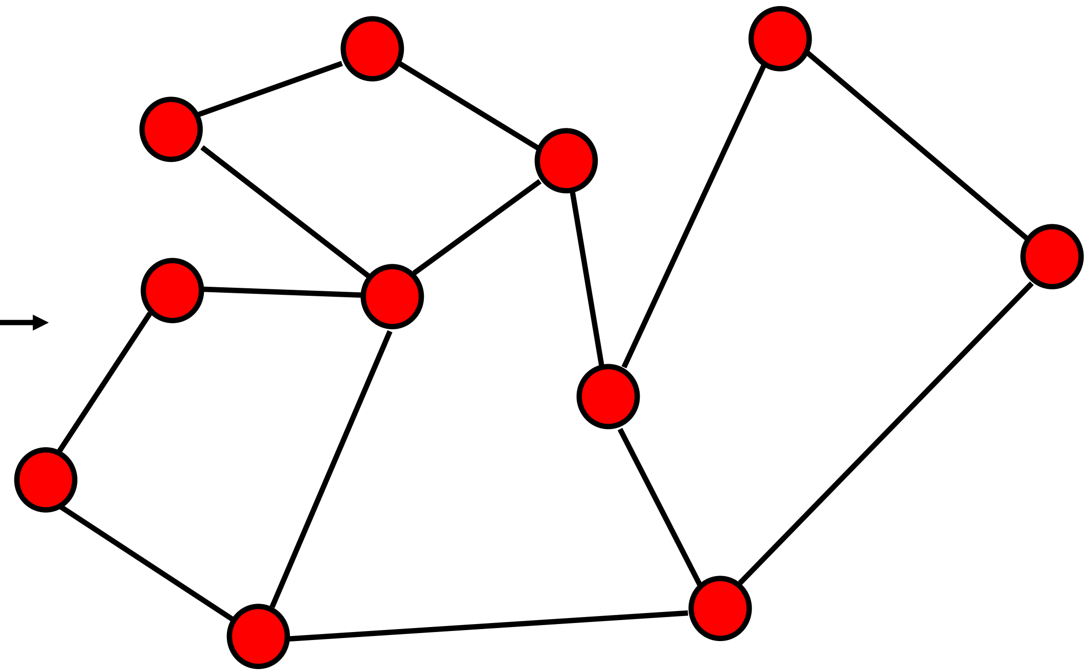


100 $

Access Control

accept
or reject?

Physical Network

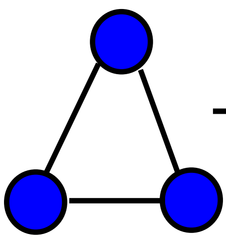


Physical network specified by node and link capacities.

# This Paper: Competitive VNet Embedding

**Cheap realization => Yes!**

VNet

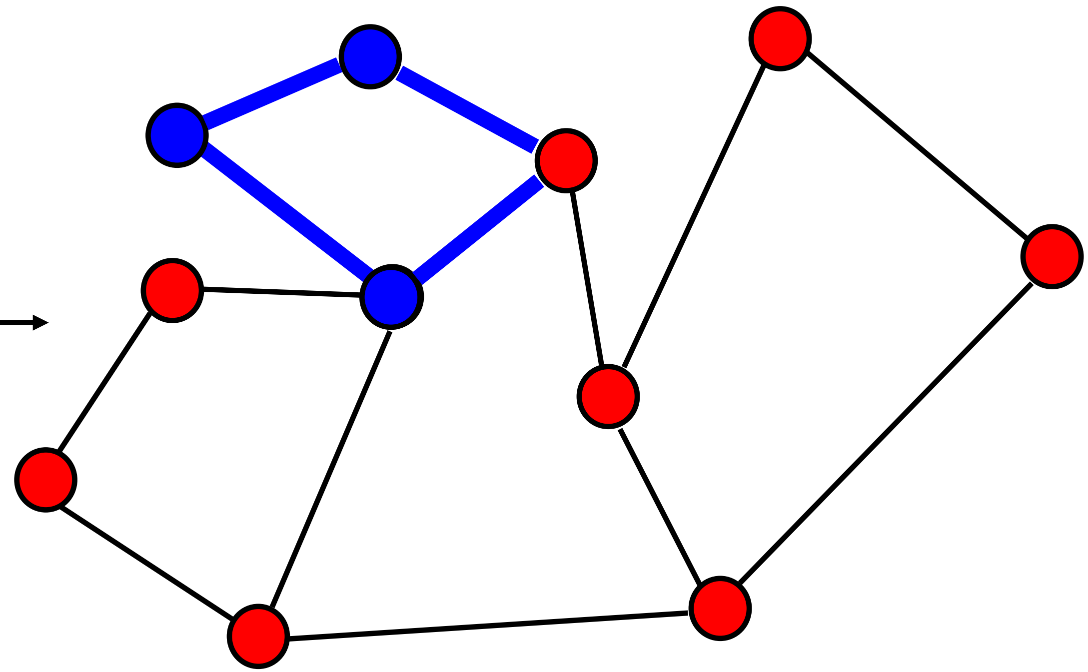


100 $

Access Control

accept
or reject?

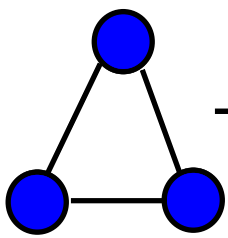
Physical Network



# This Paper: Competitive VNet Embedding

**Expensive => No!**

VNet

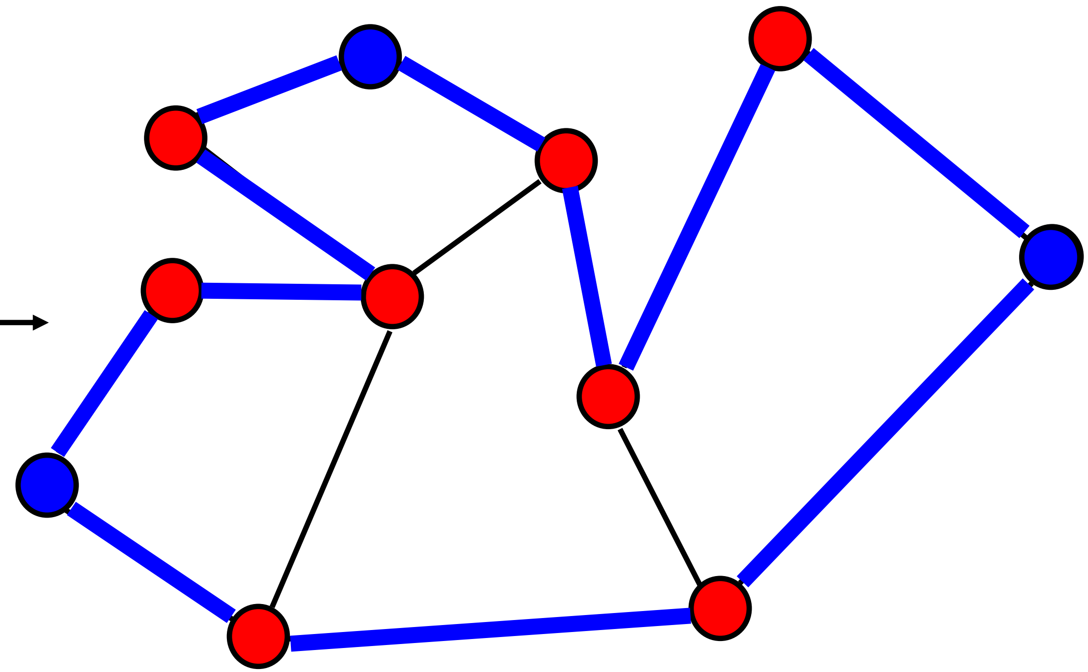


100 $

Access Control

accept
or reject?

Physical Network



**It is like „online call admission“ for entire networks („telcos“)!**

# 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!

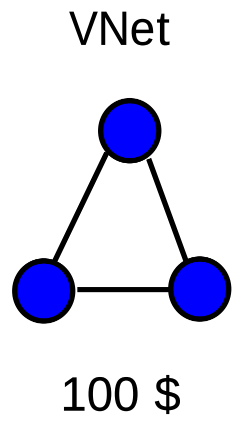
In virtual networks, many decisions need to be made online: online algorithms and network virtualization are a perfect match! ☺

No need for complex predictions but still good! ☺

# VNet Specification and Model

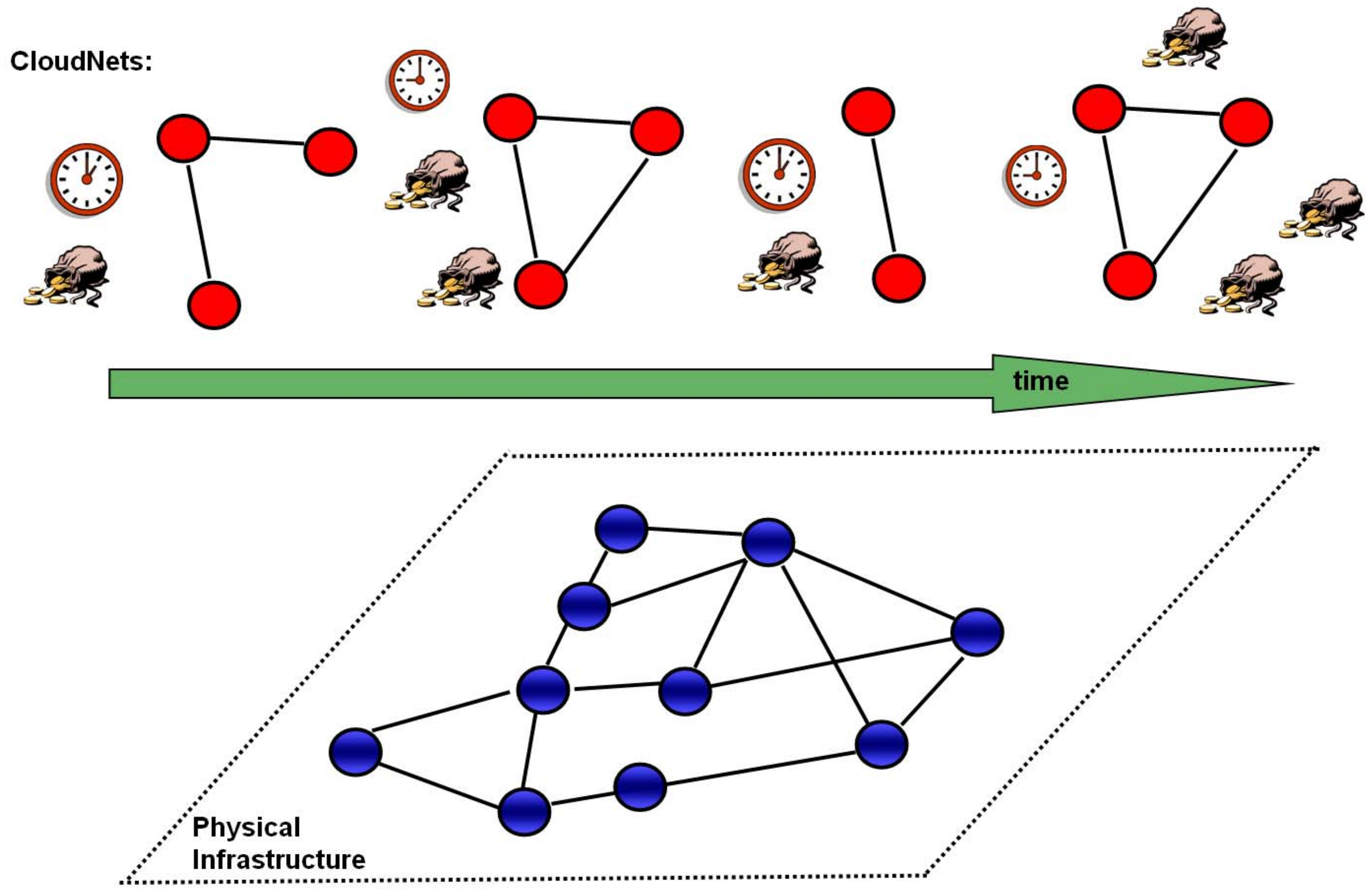
A VNet request can specify a benefit and many different QoS requirements:

- benefit if accepted
- a set of terminals to connect (or just some reqs!)
- desired bandwidth and allowed traffic patterns
- a routing model
- duration (from when until when?)




If VNets with these specifications arrive over time, which ones to accept online?!

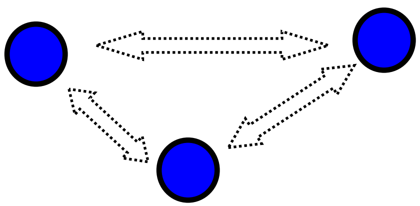
# VNet Specification and Model



# Supported VNet Traffic Patterns

## Customer Pipe

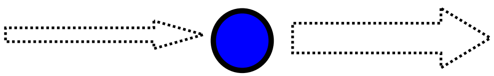
Every pair (u,v) of nodes requires a certain bandwidth.



Detailed constraints, only this traffic matrix needs to be fulfilled!

## Hose Model

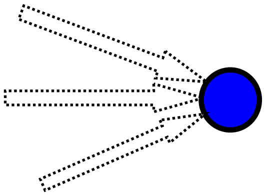
Each node v has max ingress and max egress bandwidth: each traffic matrix fulfilling them must be served.



More flexible, must support many traffic matrices!

## Aggregate Ingress Model

Sum of ingress bandwidths must be at most a parameter I.

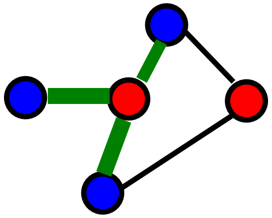


Simple and flexible! Good for multicasts etc.: no overhead, duplicate packets for output links, not input links already!

# Supported VNet Routing Models

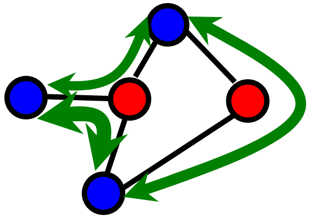
## Tree

VNet is embedded as Steiner tree:



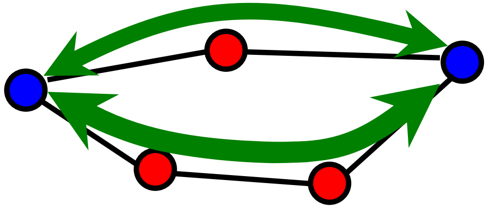
## Single Path

Each pair of nodes communicates along a single path.



## Multi Path

A linear combination specifies split of traffic between two nodes.



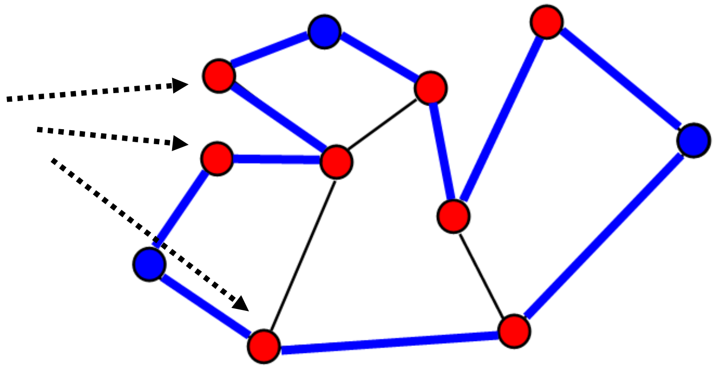
**VNets arriving over time can request different models!**

# Result

## Theorem

**The presented online algorithm GIPO is log-competitive in the amount of resources in the physical network! If capacities can be exceeded by a log factor, it is even constant competitive.**

Also works for router loads (determined by packet rate)!



# Analysis Overview

Algorithm design and analysis follows online primal-dual approach recently invented by Buchbinder&Naor!
(Application to general VNet embeddings, traffic&routing models, router loads, duration, approx oracles, ...)

1. **Formulate dynamic primal and dual LP**

| (I) | (II) |
| --- | --- |
| $\min Z_j^T \cdot \mathbf{1} + X^T \cdot C \;\; s.t.$ <br> $Z_j^T \cdot D_j + X^T \cdot A_j \geq B_j^T$ <br> $X, Z_j \geq \mathbf{0}$ | $\max B_j^T \cdot Y_j \;\; s.t.$ <br> $A_j \cdot Y_j \leq C$ <br> $D_j \cdot Y_j \leq \mathbf{1}$ <br> $Y_j \geq \mathbf{0}$ |

Fig. 1: (I) The primal covering LP. (II) The dual packing LP.

2. **Derive GIPO algorithm which always produces feasible primal solutions and where Primal $\geq 2 \cdot$ Dual**

**Algorithm 1** The General Integral (all-or-nothing) Packing Online Algorithm (GIPO).

Upon the $j$th round:

1. $f_{j,\ell} \leftarrow \text{argmin}\{\gamma(j,\ell) : f_{j,\ell} \in \Delta_j\}$ (oracle procedure)
2. If $\gamma(j,\ell) < b_j$ then, (accept)
   (a) $y_{j,\ell} \leftarrow 1$.
   (b) For each row $e$ : If $A_{e,(j,\ell)} \neq 0$ do
   $$x_e \leftarrow x_e \cdot 2^{A_{e,(j,\ell)}/c_e} + \frac{1}{w(j,\ell)} \cdot (2^{A_{e,(j,\ell)}/c_e} - 1).$$
   (c) $z_j \leftarrow b_j - \gamma(j,\ell)$.
3. Else, (reject)
   (a) $z_j \leftarrow 0$.

# Ideas of GIPO

GIPO invokes an oracle procedure to determine cost of VNet embedding!

**Algorithm 1** The General Integral (all-or-nothing) Packing Online Algorithm (GIPO).

Upon the $j$th round:

1. $f_{j,\ell} \leftarrow \text{argmin}\{\gamma(j,\ell) : f_{j,\ell} \in \Delta_j\}$ (oracle procedure)
2. If $\gamma(j,\ell) < b_j$ then, (accept)
   (a) $y_{j,\ell} \leftarrow 1$.
   (b) For each row $e$ : If $A_{e,(j,\ell)} \neq 0$ do

$$x_e \leftarrow x_e \cdot 2^{A_{e,(j,\ell)}/c_e} + \frac{1}{w(j,\ell)} \cdot (2^{A_{e,(j,\ell)}/c_e} - 1).$$

   (c) $z_j \leftarrow b_j - \gamma(j,\ell)$.
3. Else, (reject)
   (a) $z_j \leftarrow 0$.

Algorithm efficient... except for oracle (static, optimal embedding)!
What if we only use a suboptimal embedding here?!

# Effect of Approximate Oracles

Problem: computation of optimal embeddings NP-hard!
Thus: use approximate embeddings! (E.g., Steiner tree)

**GIPO:**

**Algorithm 1** The General Integral (all-or-nothing) Packing Online Algorithm (GIPO).

Upon the $j$th round:

1. $f_{j,\ell} \leftarrow \text{argmin}\{\gamma(j,\ell) : f_{j,\ell} \in \Delta_j\}$ (oracle procedure)
2. If $\gamma(j,\ell) < b_j$ then, (accept)
   (a) $y_{j,\ell} \leftarrow 1$.
   (b) For each row $e$ : If $A_{e,(j,\ell)} \neq 0$ do

$$x_e \leftarrow x_e \cdot 2^{A_{e,(j,\ell)}/c_e} + \frac{1}{w(j,\ell)} \cdot (2^{A_{e,(j,\ell)}/c_e} - 1).$$

   (c) $z_j \leftarrow b_j - \gamma(j,\ell)$.
3. Else, (reject)
   (a) $z_j \leftarrow 0$.

**Competitive ratio ρ**

**Embedding Approx.:**

**<insert your favorite approx algo>**

**Approx ratio r**

## Lemma

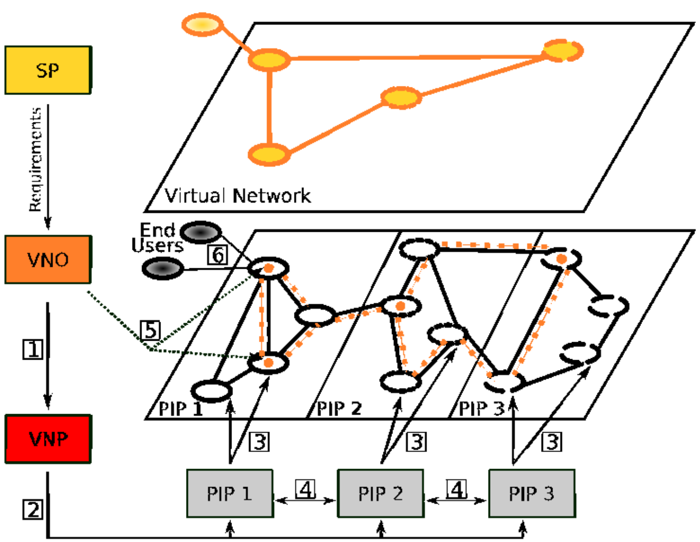
**The approximation does not reduce the overall competitive ratio by much: we get ρ*r ratio!**

# Conclusion

**VNet admission control problem = „online call admission for telcos“**

Summary:

Very general online VNet embedding algorithm.



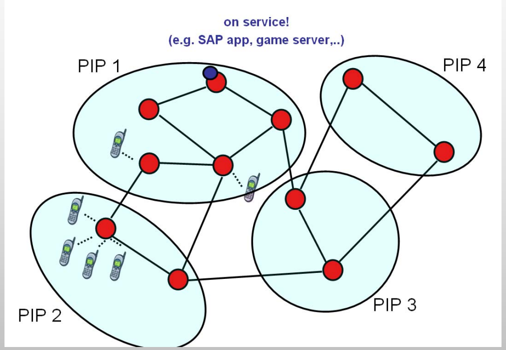

Future work:

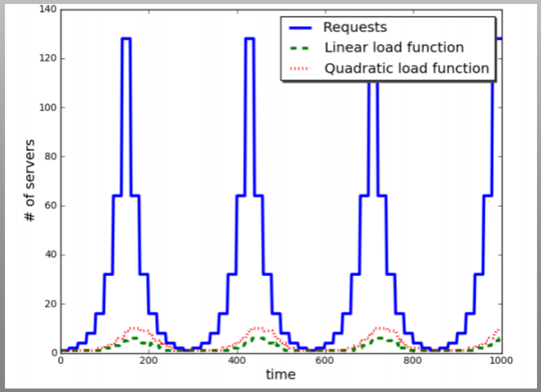
1. There is a lower bound of log(n*T) from online circuit switching
2. More complex embedding constraints, full-duplex links mapped on asymmetric, half-duplex network? Or architecture compatibility?
3. With preemption better competitive ratio possible?
4. Non-linear objective functions? Maybe also with Buchbinder&Naor framework, using semi-definite programming!
5. Embedding support in prototype only offline so far!

# Other algorithmic VNet problems studied so far...

## Online Migration and Allocation

Goal:
*Online VNet migration in multi-provider environments and online server allocation*




**VISA 2010, Hot-ICE 2011, IPTCOMM 2011**

## Even More General Embeddings

Goal:
*E.g.,different link types*

**Nodes:**

map_node: $\sum_{v \in NE_S} new(u,v) = 1$ $\quad \forall u \in NE_{VN}$
set_new: $alloc_{r_S}(u,v,r_V) \leq cap_{r_S}(v) new(u,v)$ $\quad \forall u \in NE_{VN}, v \in NE_S, r_V \in R_V, r_S \in R_S$
req_min: $alloc_{r_V}(u,v) \geq new(u,v) req(u,r_V,s)$ $\quad \forall u \in NE_{VN}, r_V \in R_V, r_S \in R_S, s = \text{minimum}$
req_max: $alloc_{r_V}(u,v) \leq new(u,v) req(u,r_V,s)$ $\quad \forall u \in NE_{VN}, r_V \in R_V, r_S \in R_S, s = \text{maximum}$
req_con: $alloc_{r_V}(u,v) = new(u,v) req(u,r_V,s)$ $\quad \forall u \in NE_{VN}, r_V \in R_V, r_S \in R_S, s = \text{constant}$

**Mapping:**

relate_V: $alloc_{r_V}(u,v) \geq min\_alloc_{r_V} \cdot new(u,v)$ $\quad \forall u \in NE_V, v \in NE_S, r_V \in R_V$
allowed: $suit(u,v) \geq new(u,v)$ $\quad \forall u \in NE_V, v \in NE_S$
ne_capacity: $\sum_{u \in NE_V} \sum_{r_V \in R_V} alloc_{r_S}(u,v,r_V) \leq cap_{r_S}(v)$ $\quad \forall v \in NE_S, r_S \in R_S$
capacity: $\sum_{v \in NE_S} \sum_{u \in NE_V} \sum_{r_V \in R_V} alloc_{r_S}(u,v,r_V) \leq cap(r_S)$ $\quad \forall r_S \in R_S$
load: $weight_{r_S} / cap(r_S) \cdot \sum_{v \in NE_S} \sum_{u \in NE_V} \sum_{r_V \in R_V} alloc_{r_S}(u,v,r_V) \leq load(r_S)$ $\quad \forall r_S \in R_S$
max_load: $load(r_S) \leq max\_load$ $\quad \forall r_S \in R_S$

**Resource-Variable Relation:**

resource: $\sum_{r_S \in R_S} prop(r_V, r_S) alloc_{r_S}(u,v,r_V) = alloc_{r_V}(u,v)$ $\quad \forall u \in NE_V, v \in NE_S, r_V \in R_V$
flow_res: $\sum_{r_S \in R_S} prop(r_V, r_S) flow_{r_S}(f,v,w,r_V) = flow_{r_V}(f,v,w)$ $\quad \forall f \in Fl(u), (v,w) \in NE_S^2, r_V \in R_f, \forall u \in NE_{VL}$

**Links:**

map_link: $\sum_{v \in NE_S} new(u,v) \geq 1$ $\quad \forall u \in NE_{VL}$
map_flow: $new(f,v) \leq new(u,v)$ $\quad \forall f \in Fl(u), v \in NE_S, \forall u \in NE_{VL}$
map_src: $new(f,v) \geq new(q_f,v)$ $\quad \forall f \in Fl(u), v \in NE_S, q_f$ source of $f; \forall u \in NE_{VL}$
map_sink: $new(f,v) \geq new(d_f,v)$ $\quad \forall f \in Fl(u), v \in NE_S, d_f$ sink of $f; \forall u \in NE_{VL}$
req_min: $\sum_{w \in NE_S} (flow_{r_V}(f,v,w) - flow_{r_V}(f,w,v)) \geq new(q_f,v) req(u,r_V,s) - new(d_f,v)\infty$
$\forall f \in Fl(u), v \in NE_S, r_V \in R_f; \forall u \in NE_{VL}, s = \text{minimum}$
req_max: $\sum_{w \in NE_S} (flow_{r_V}(f,v,w) - flow_{r_V}(f,w,v)) \leq new(q_f,v) req(u,r_V,s) + new(d_f,v)\infty$
$\forall f \in Fl(u), v \in NE_S, r_V \in R_f; \forall u \in NE_{VL}, s = \text{maximum}$
req_const: $\sum_{w \in NE_S} (flow_{r_V}(f,v,w) - flow_{r_V}(f,w,v)) = new(q_f,v) req(u,r_V,s) - new(d_f,v) req(u,r_V,s)$
$\forall f \in Fl(u), v \in NE_S, r_V \in R_f; \forall u \in NE_{VL}, s = \text{constant}$

**Link Allocation:**

exp_out: $\sum_{w \in NE_S} flow_{r_S}(f,v,w,r_V) \leq alloc_{r_S}(u,v,r_V)$ $\quad \forall f \in Fl(u), v \in NE_S, r_V \in R_f, r_S \in R_S, \forall u \in NE_{VL}$
exp_in: $\sum_{w \in NE_S} flow_{r_S}(f,w,v,r_V) \leq alloc_{r_S}(u,v,r_V)$ $\quad \forall f \in Fl(u), v \in NE_S, r_V \in R_f, r_S \in R_S, \forall u \in NE_{VL}$
direction: $flow_{r_S}(f,v,w,r_V) \leq new(u,v) cap_{r_S}(v,w)$ $\quad \forall f \in Fl(u), (v,w) \in NE_S^2, r_V \in R_f, r_S \in R_S, \forall u \in NE_{VL}$
relate_f: $\sum_{w \in NE_S} flow_{r_S}(f,v,w,r_V) + flow_{r_S}(f,w,v,r_V) \geq new(f,v)$ $\quad \forall f \in Fl(u), \forall u \in NE_{VL}, v \in NE_S, r_V \in R_f, r_S \in R_S$

**Migration:**

new: $\sum_{v \in NE_S} old(u,v) \geq mig(u)$ $\quad \forall u \in NE_V$
migrated: $old(u,v) - new(u,v) \leq mig(u)$ $\quad \forall u \in NE_V, v \in NE_S$

**ArXiv 2011**

# Thank you!

Further reading: project website!

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

# Simplified LP

$$\min \sum_{e\in E} x_e \cdot c(e) + \sum_{v\in V} x_v \cdot c(v) + \sum_i z_i \cdot d_i \quad s.t.$$

$$\text{(Covering Const.)} \forall i \forall \Delta \in \Delta_i z_i + \alpha(i,\Delta) \geq b_i$$

$$\forall i \forall \Delta \in \Delta_i x_e, x_v, z_i \geq 0$$

(I)

maximize benefit!

$$\max \sum_i b_i \cdot \sum_{\Delta_{ij}\in\Delta_i} f_{ij} \quad s.t.$$

realization of i-th request (will be integer, accept fully or not at all)

... while ensuring capacity and no more than demand!

$$\text{(Vertex Capacity Const.)} \quad \forall v \in V \;\; flow(v) \leq c(v)$$

$$\text{(Edge Capacity Const.)} \quad \forall e \in E \;\; flow(e) \leq c(e)$$

$$\text{(Demand Const.)} \quad \forall i \sum_{\Delta_{ij}\in\Delta_i} f_{ij} \leq d_i$$

$$f \geq 0$$

(II)

Fig. 1: (I) The Primal linear embedding program. (II) The Dual linear embedding program.

# Simplified LP

essentially, exponential load...

$$\min \sum_{e \in E} x_e \cdot c(e) + \sum_{v \in V} x_v \cdot c(v) + \sum_i z_i \cdot d_i \quad s.t.$$

$$\text{(Covering Const.)} \forall i \forall \Delta \in \Delta_i z_i + \alpha(i, \Delta) \geq b_i$$

$$\forall i \forall \Delta \in \Delta_i x_e, x_v, z_i \geq 0$$

(I)

$$\max \sum_i b_i \cdot \sum_{\Delta_{ij} \in \Delta_i} f_{ij} \quad s.t.$$

$$\text{(Vertex Capacity Const.)} \quad \forall v \in V \quad \mathit{flow}(v) \leq c(v)$$

$$\text{(Edge Capacity Const.)} \quad \forall e \in E \quad \mathit{flow}(e) \leq c(e)$$

$$\text{(Demand Const.)} \quad \forall i \sum_{\Delta_{ij} \in \Delta_i} f_{ij} \leq d_i$$

$$f \geq 0$$

(II)

Fig. 1: (I) The Primal linear embedding program. (II) The Dual linear embedding program.

# Simplified Algo

**Algorithm 1** The ISTP Algorithm.

Input: $G = (V, E)$ (possibly infinite), sequence of requests $\{r_i\}_{i=1}^{\infty}$ where $r_i \triangleq (U_i, c_i, d_i, b_i)$.

Upon arrival of request $r_i$:

1) $j \leftarrow \text{argmin}\{\alpha(i,j) : \Delta_{ij} \in \Delta_i\}$ (find a lightest realization over the terminal set $U_i$ using an oracle).
2) If $\alpha(i,j) < b_i$ then, (accept $r_i$)
   a) $f_{ij} \leftarrow d_i$.
   b) For each $e \in E(\Delta_{ij})$ do
   $$x_e \leftarrow x_e \cdot 2^{d_i/c(e)} + \frac{1}{|V(\Delta_{ij})|} \cdot (2^{d_i/c(e)} - 1).$$
   c) For each $v \in V(\Delta_{ij})$ do
   $$x_v \leftarrow x_v \cdot 2^{c_i/c(v)} + \frac{d_i/c_i}{|V(\Delta_{ij})|} \cdot (2^{c_i/c(v)} - 1).$$
   d) $z_i \leftarrow b_i - \alpha(i,j)$.
3) Else, (reject $r_i$)
   a) $z_i \leftarrow 0$.

oracle
(triangle only)

update primal
variables if accepted

# Simplified Analysis

Step (2b) increases the cost $\sum_e x_e \cdot c(e)$ as follows (change $\Delta(x_e) = \sum_e (x_e^t - x_e^{t-1}) \cdot c(e)$):

$$
\begin{aligned}
\Delta(x_e) &\leq \sum_{e\in\Delta} \left[ x_e \cdot (2^{d_i/c(e)} - 1) + \frac{1}{|V(\Delta_{ij})|} \cdot (2^{d_i/c(e)} - 1) \right] \cdot c(e) \\
&= \sum_{e\in\Delta} \left( x_e + \frac{1}{|V(\Delta_{ij})|} \right) \cdot (2^{d_i/c(e)} - 1) \cdot c(e) \\
&\leq c_{\min}(e) \cdot (2^{d_i/c_{\min}(e)} - 1) \sum_{e\in\Delta} \left( x_e + \frac{1}{|V(\Delta_{ij})|} \right) \\
&\leq d_i \cdot (2^1 - 1) \sum_{e\in\Delta} \left( x_e + \frac{1}{|V(\Delta_{ij})|} \right) \\
&\leq d_i \cdot \sum_{e\in\Delta} x_e + d_i \cdot \sum_{e\in\Delta} \frac{1}{|V(\Delta_{ij})|} \\
&\leq d_i \cdot \sum_{e\in\Delta} x_e + d_i \, . \qquad (1)
\end{aligned}
$$

Step (2c) increases the cost $\sum_v x_v \cdot c(v)$ as follows (change $\Delta(x_v) = \sum_v (x_v^t - x_v^{t-1}) \cdot c(v)$):

$$
\begin{aligned}
\delta(x_v) &\leq \sum_{v\in\Delta} \left[ x_v \cdot (2^{c_i/c(v)} - 1) + \frac{d_i/c_i}{|V(\Delta_{ij})|} \cdot (2^{c_i/c(v)} - 1) \right] \cdot c(v) \\
&= \sum_{v\in\Delta} \left( x_v + \frac{d_i/c_i}{|V(\Delta_{ij})|} \right) \cdot (2^{c_i/c(v)} - 1) \cdot c(v) \\
&\leq c_{\min}(v) \cdot (2^{c_i/c_{\min}(v)} - 1) \sum_{v\in\Delta} \left( x_v + \frac{d_i/c_i}{|V(\Delta_{ij})|} \right) \\
&\leq c_i \cdot (2^1 - 1) \sum_{v\in\Delta} \left( x_v + \frac{d_i/c_i}{|V(\Delta_{ij})|} \right) \\
&\leq c_i \cdot \sum_{v\in\Delta} x_v + c_i \cdot \sum_{v\in\Delta} \frac{d_i/c_i}{|V(\Delta_{ij})|} \\
&\leq c_i \cdot \sum_{v\in\Delta} x_v + d_i \, . \qquad (2)
\end{aligned}
$$

after each request, primal variables constitute feasible solutions...