南京大學
NANJING UNIVERSITY

Leveraging Tenant Flexibility in Resource Allocation for Virtual Networks

Presenter: Sheng Zhang

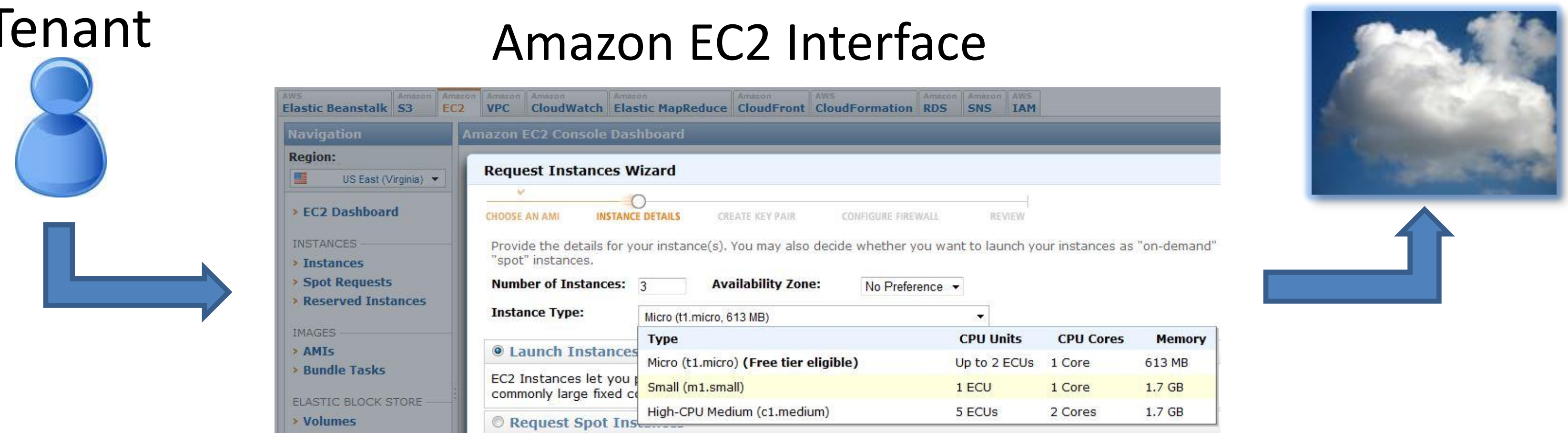
Sheng Zhang, Zhuzhong Qian, Jie Wu, and Sanglu Lu
Nanjing University & Temple University
5-AUG-2014
ICCCN'14@Shanghai, China

Cloud Computing

Bandwidth Guarantee? No.

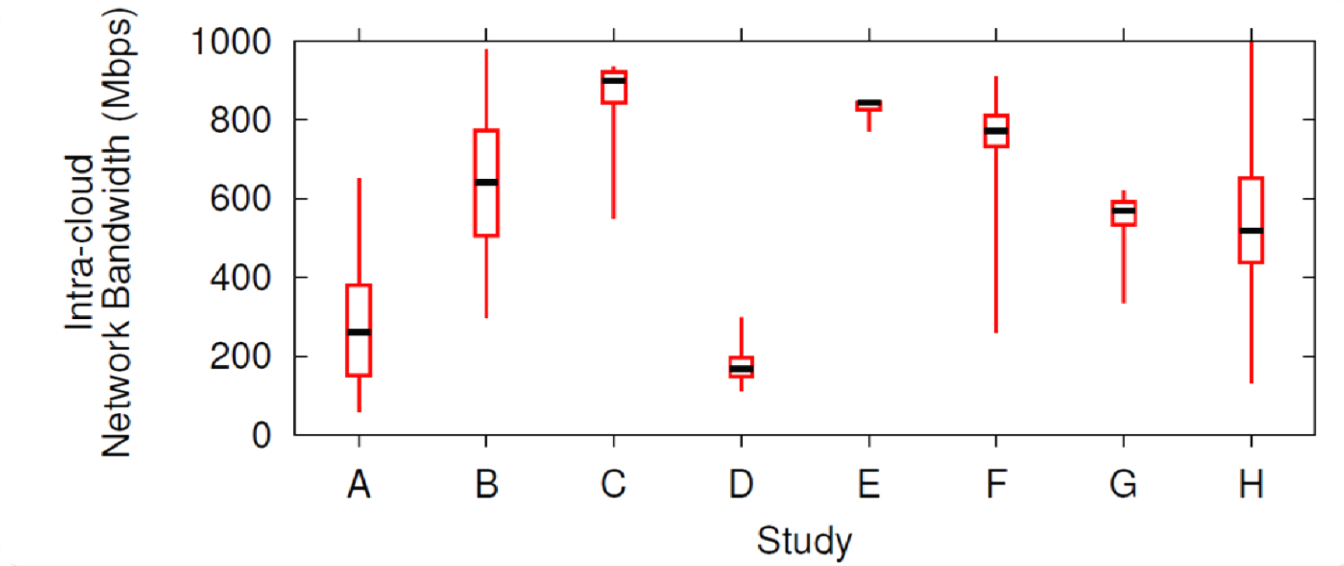
Tenant

Amazon EC2 Interface



- Current billing model is per-VM (CPU, storage, etc)
  - Amazon EC2 small instances: $0.085/hour
  - No intra-datacenter network cost

Unpredictable Performance

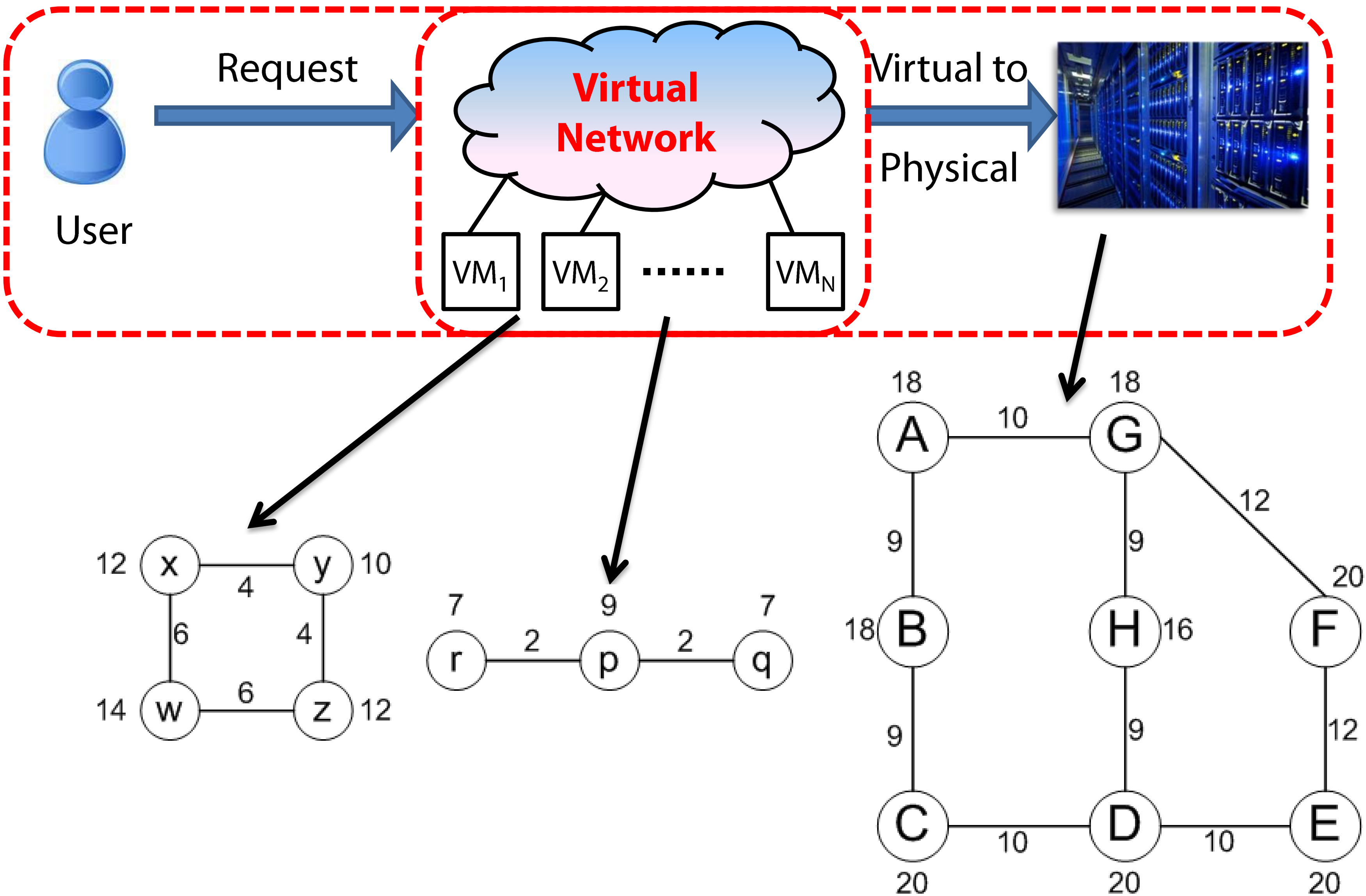


[BCKR11]

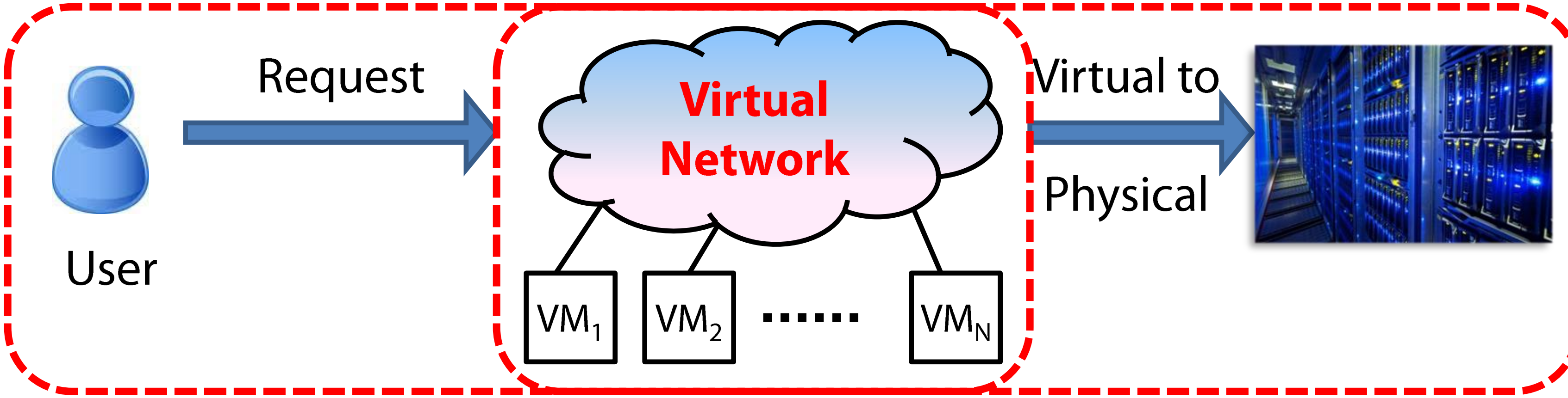
- When there is no bandwidth guarantee between VMs
  - Tenants will not migrate certain applications to clouds
  - Providers cannot achieve high resource utilization, and thus lose revenue.

[BCKR11] H. Ballani, P. Costa, T. Karagiannis, and A. Rowstron, "Towards predictable datacenter networks," in Proc. of ACM SIGCOMM 2011, pp. 242–253.

Virtual Networks as Better Interfaces



Previous Work



- Bin packing-based VM consolidation [WML11]
- Network-aware VM placememt [AL12]
- VC and VOC [BCKR11]
- Path splitting [YYRC08]
- Subgraph isomorphism [LK09]

[WML11] Consolidating virtual machines with dynamic bandwidth demand in data centers, INFOCOM 2011

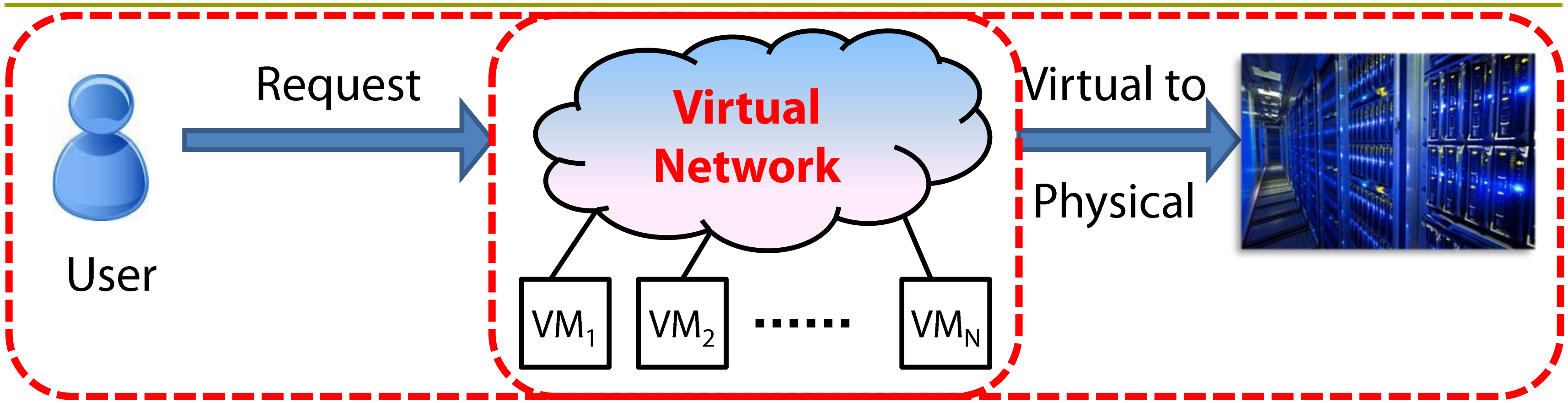
[AL12] Network aware resource allocation in distributed clouds, INFOCOM 2012

[BCKR11] Towards predictable datacenter networks, SIGCOMM 2011

[YYRC08] Rethinking virtual network embedding: substrate support for path splitting and migration

[LK09] A virtual network mapping algorithm based on subgraph isomorphism detection, VISA 2009

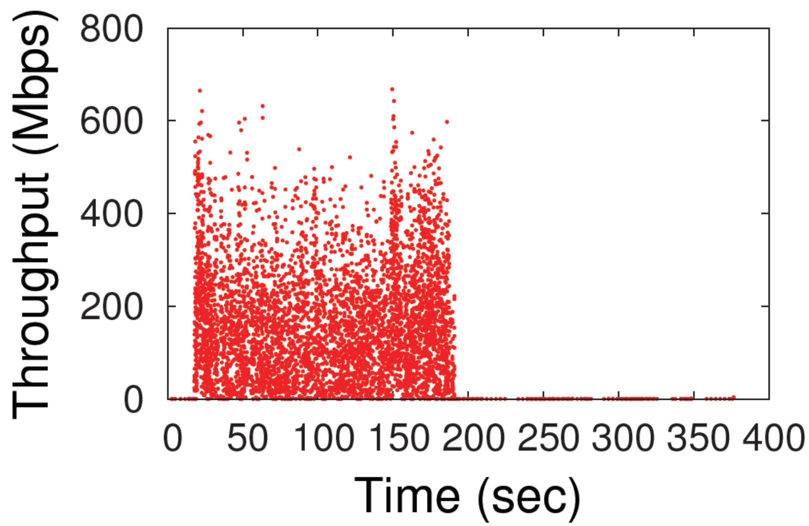
Limitations of Previous Work



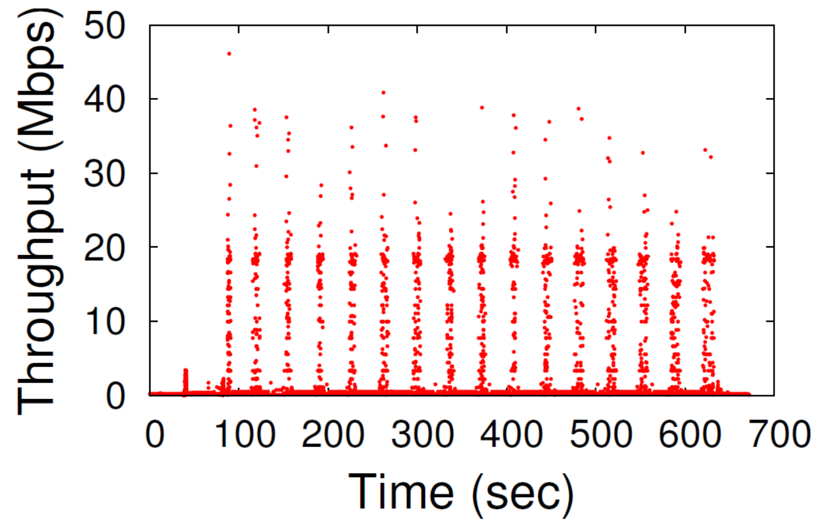
- Ignoring network requirements [WML11] [AL12]
- Tree topology [OCKR11]
- Fixed resource reservation [YYRC08] [LK09]

[WML11] Consolidating virtual machines with dynamic bandwidth demand in data centers, INFOCOM 2011
[AL12] Network aware resource allocation in distributed clouds, INFOCOM 2012
[BCKR11] Towards predictable datacenter networks, SIGCOMM 2011
[YYRC08] Rethinking virtual network embedding: substrate support for path splitting and migration
[LK09] A virtual network mapping algorithm based on subgraph isomorphism detection, VISA 2009

Time-Varying Resource Requirements



Map-Reduce Sort [XDHK12]



Map-Reduce Word Count [XDHK12]

- Applications: different resource requirements during different executing phases
- Users change over time, causing fluctuating resource demands.

[XDHK12] D. Xie, N. Ding, Y. C. Hu, and R. Kompella, "The only constant is change: incorporating time-varying network reservations in data centers," in Proc. of ACM SIGCOMM 2012, pp.199–210.

Content

- Demand Model
- Problem Formulation
- Solution
- Evaluation
- Conclusions

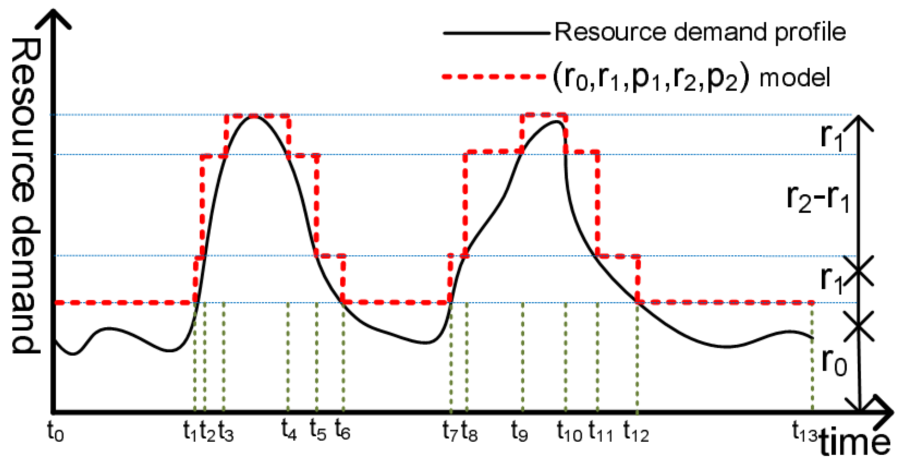
Demand Model

- The resource demand of a VM v at time t is denoted by $R(v,t)$, which consists of
  - $R_0(v,t)$: basic part r_0, always exists
  - $R_1(v,t)$: variable part r_1, exists with a probability of p_1^v
  - $R_2(v,t)$: variable part r_2, exists with a probability of p_2^v
- Tuple $<r_0, r_1, p_1, r_2, p_2>$

$R(v,t)$	r_0^v	$r_0^v + r_1^v$	$r_0^v + r_2^v$	$r_0^v + r_1^v + r_2^v$
P	$(1-p_1^v)(1-p_2^v)$	$p_1^v(1-p_2^v)$	$(1-p_1^v)p_2^v$	$p_1^v p_2^v$

Probability distribution of $R(v,t)$.

Tenant Flexibility

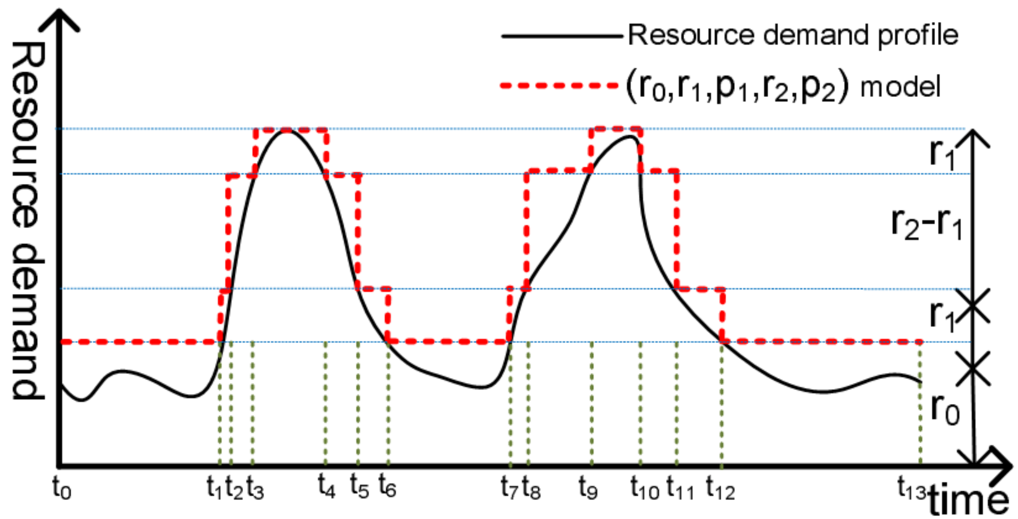


- Flexibly control the trade-off between performance and cost through adjusting *(r_0,r_1,p_1,r_2,p_2)*

- Providers charge less for shared resources than dedicated resources (i.e., the unit price for r_1 or r_2 is less than that for r_0).

- At one extreme, if a tenant cares only performance
  - Set $r_1=r_2=p_1=p_2=0$
- At the other extreme, if a tenant wants to minimize cost
  - Set $r_0=0$

Some Other Good Properties



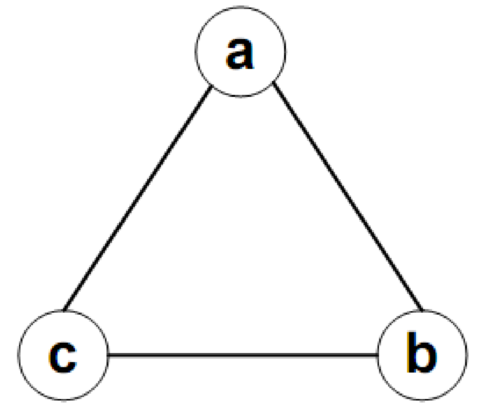
- Backwards-compatible
  - VDC, VC, VOC, etc. are special cases of our model
- Flexibly control the trade-off between model precision and complexity
  - By tuning # of parts in the model

Content

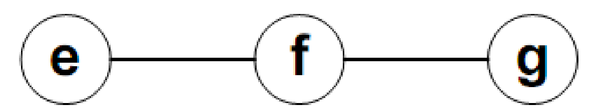
- Demand Model
- Problem Formulation
- Solution
- Evaluation
- Conclusions

Virtual Network: *VNet*

- A weighted undirected graph
  - Vertices: VMs
  - Edges: links between VMs
- Each vertex *v* (resp. edge *e)* has a time-varying resource demand *R(v,t)* (resp. *R(e,t)*)
- lifetime: *lt*



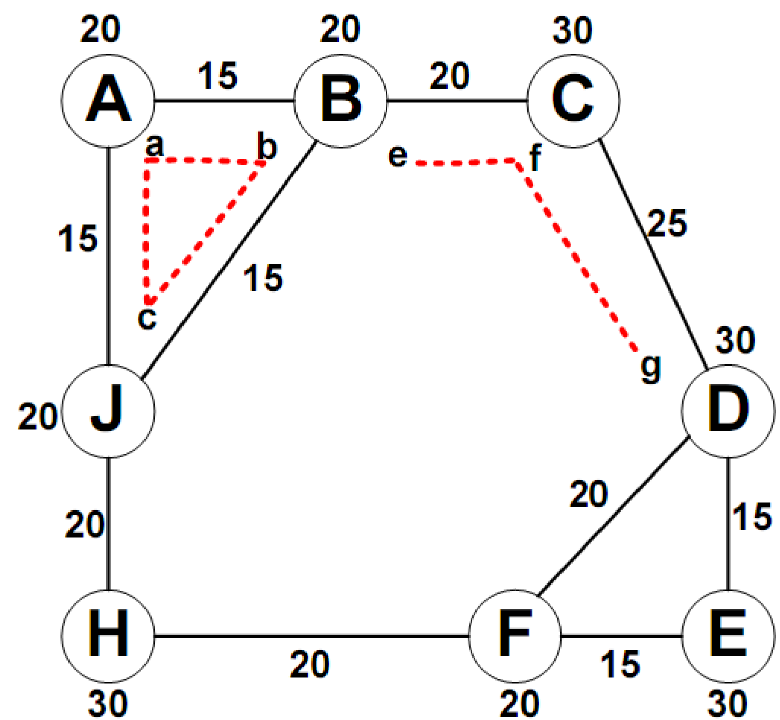
$VNet_1$



$VNet_2$

Physical Network

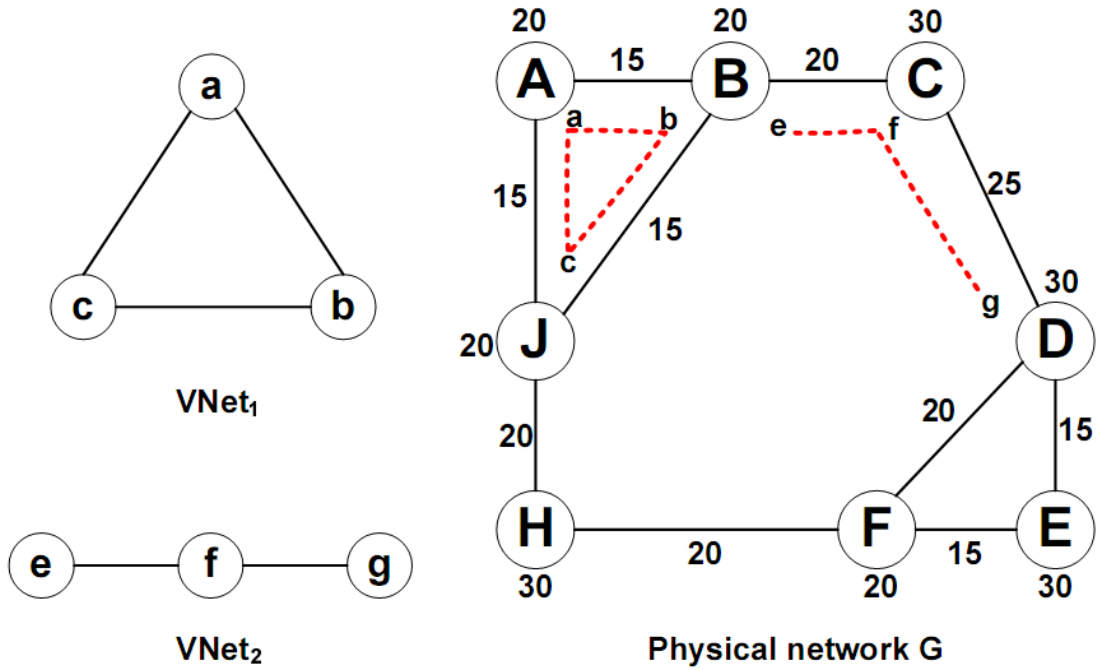
- A weighted undirected graph
  - Vertices: PMs
  - Edges: links between PMs
- Each vertex *n* (resp. edge *e*) has CPU (resp. bandwidth) capacity *C(n)* (resp. *B(e)*)
- Denote the set of simple paths between n_i and n_j by $P(n_i,n_j)$



Physical network G

Resource Allocation

- Virtual machine mapping
  - Different VMs map to different PMs
- Virtual link mapping
  - Virtual links map to physical paths

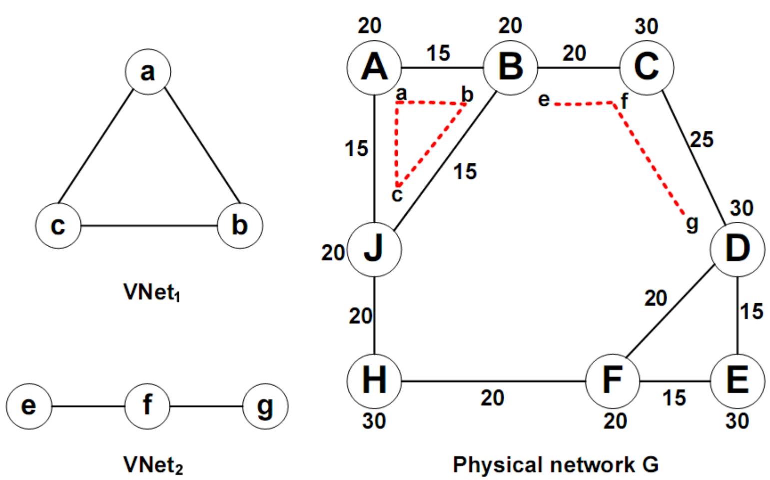


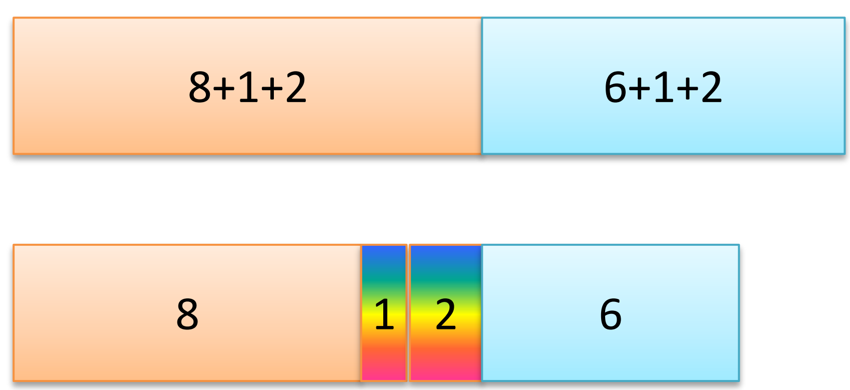
Collision Threshold (1/2)

- Resource demands from different VNets are mutually independent
- To improve physical resource utilization, we propose to share physical resources among variable parts of resource demands
- However, when more than one variable part occurs simultaneously, a collision happens.
- The cloud provider should provide probabilistic performance guarantee by bounding the maximum collision probability p_{th}

Collision Threshold (2/2): Example

- $R(b,t)=<8,1,0.1,2,0.1>$, $R(e,t)=<6,1,0.2,2,0.2>$
- If VMs *b* and *e* do not share physical resource
  - they would occupy a total of 20 units of resources.
- If resource sharing is exploited (assuming $p_{th}=0.1$)
  - Since $0.1 \times 0.2 = 0.02 < p_{th}$, we can safely share 1 (resp. 2) unit of physical resource between the first (second) variable parts of resource demands of these two VMs
  - they would occupy a total of 17 units of resources.





Objective

- The revenue of accepting a VNet is

$$\mathbb{R}(VNet) = [\alpha \sum_{n \in V} (r_0^n + p_1^n r_1^n + p_2^n r_2^n) + \beta \sum_{e \in E} (r_0^e + p_1^e r_1^e + p_2^e r_2^e)] \cdot lt$$

Expectation

Expectation

- Maximize cloud provider's revenue $\sum_{VNet} \mathbb{R}(VNet)$

Content

- Demand Model
- Problem Formulation
- Solution
- Evaluation
- Conclusions

Work-Conserving Allocation (WCA)

- Global stage
  - Virtual Machine Mapping
  - Virtual Link Mapping
- Local stage
  - Physical resource sharing among multiple variable parts of resource demands to achieve work-conserving utilization

Virtual Machine Mapping

- Sort VMs in the descending order of their respective expected resource demands
- Place each VM in that order in the unused PM with the most residual resource
  - Maximum-first fashion
    - Avoiding bottleneck
    - Early detection of requests that cannot be satisfied

Virtual Link Mapping

- Given a pair of VMs, map the virtual link between them to the shortest path between the corresponding PMs
- If we cannot find such a path, divide the bandwidth demand of the VL into two equal parts, and then map them separately.
- Keep splitting the demand into equal parts until we can successfully map them or the number of equal parts becomes larger than a threshold, say *K*.

The Local Stage Sharing (1/2)

Share physical resources among multiple variable parts of resource demands from different virtual networks

- Two demands <30,20,0.4,10,0.3> and <20,15,0.2,10,0.1> shares a physical machine which has 100 units of physical resources. The collision threshold is 0.1.

$r_0^{v1}=30$	$r_1^{v1}=20$ $r_1^{v2}=15$	$r_2^{v1}=10$ $r_2^{v2}=10$	$r_0^{v2}=20$	AC(PM) =20

The Local Stage Sharing (2/2)

Share physical resources among multiple variable parts of resource demands from different virtual networks

$r_0^{v1}=30$	$r_1^{v1}=20$ $r_1^{v2}=15$	$r_2^{v1}=10$ $r_2^{v2}=10$	$r_0^{v2}=20$	AC(PM) =20

- Let's check whether a third demand <20,15,0.3,5,0.1> could be placed in this PM.
  - The basic part (i.e., 20) is OK
  - The first variable part (i.e., 15, 0.3) cannot be placed together with $r1^{v1}$ and $r1^{v2}$, because they would collide with a probability of 0.212, which is larger then 0.1.

Content

- Demand Model
- Problem Formulation
- Solution
- Evaluation
- Conclusions

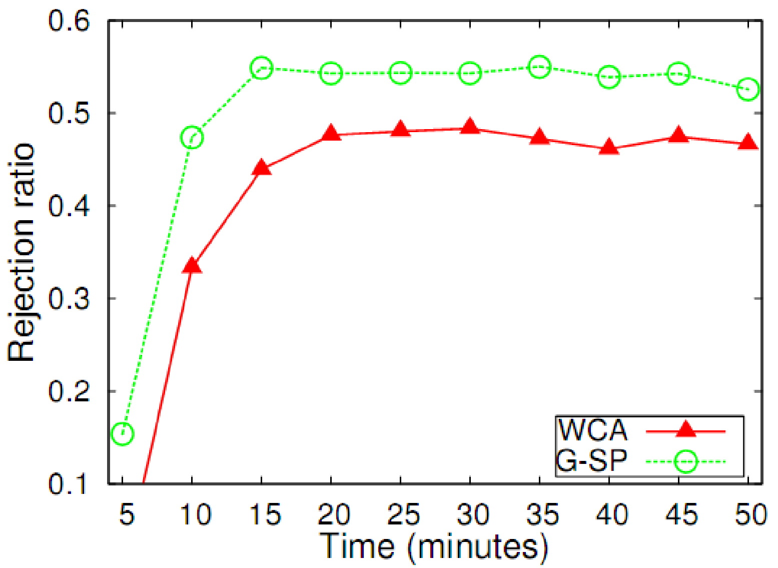
Simulation Setup (1/2)

- Physical Network: 60 PMs, each pair of them is connected with a probability of 0.3
- CPU capacity of each PM: 100
- Bandwidth capacity of each PL: 100
- Collision threshold: 0.2
- # of VMs in a Vnet: [*Avg-4*, *Avg+4*]
- Each pair of VMs is connected with a probability of 0.3
- The peak resource demand of each VM or VL: [*20*, *HR*]
- The lifetime of each VNet follows an exponential distribution with an average of 300 seconds.

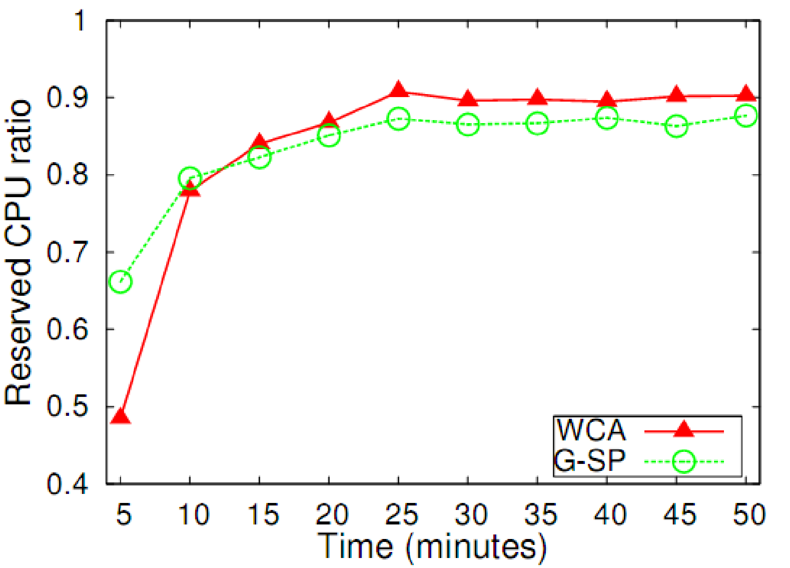
Simulation Setup (2/2)

Notation and its value by default	Definitions
$K = 3$	the maximum number of portions that a networking demand can be split into
$\lambda = 1/12$	the average interval between two consecutive VNets' arrivals
p_{th}=0.2	collision threshold
Avg=6	the average number of VNs in a VNet
HR=30	the maximum resource demand of a VN or VL
p_1=0.3	the occurring probability of the 1st variable part
p_2=0.1	the occurring probability of the 2nd variable part

Simulation Results (1/3)

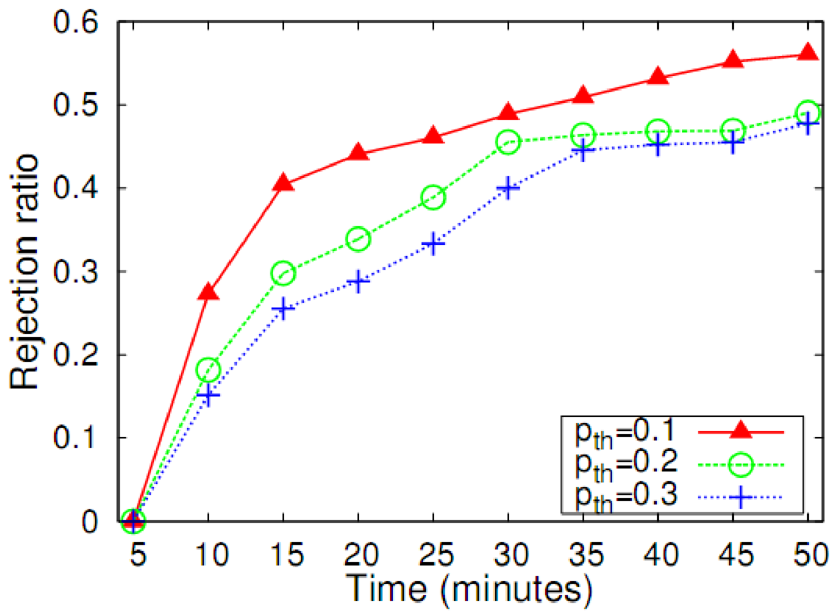


(a) Percentage of rejected requests

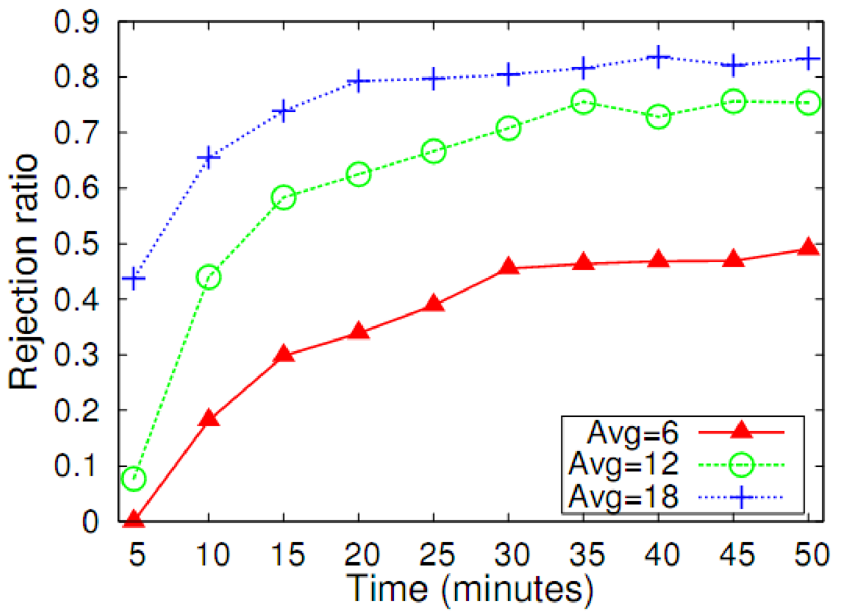


(b) Reserved CPU resource

Simulation Results (2/3)

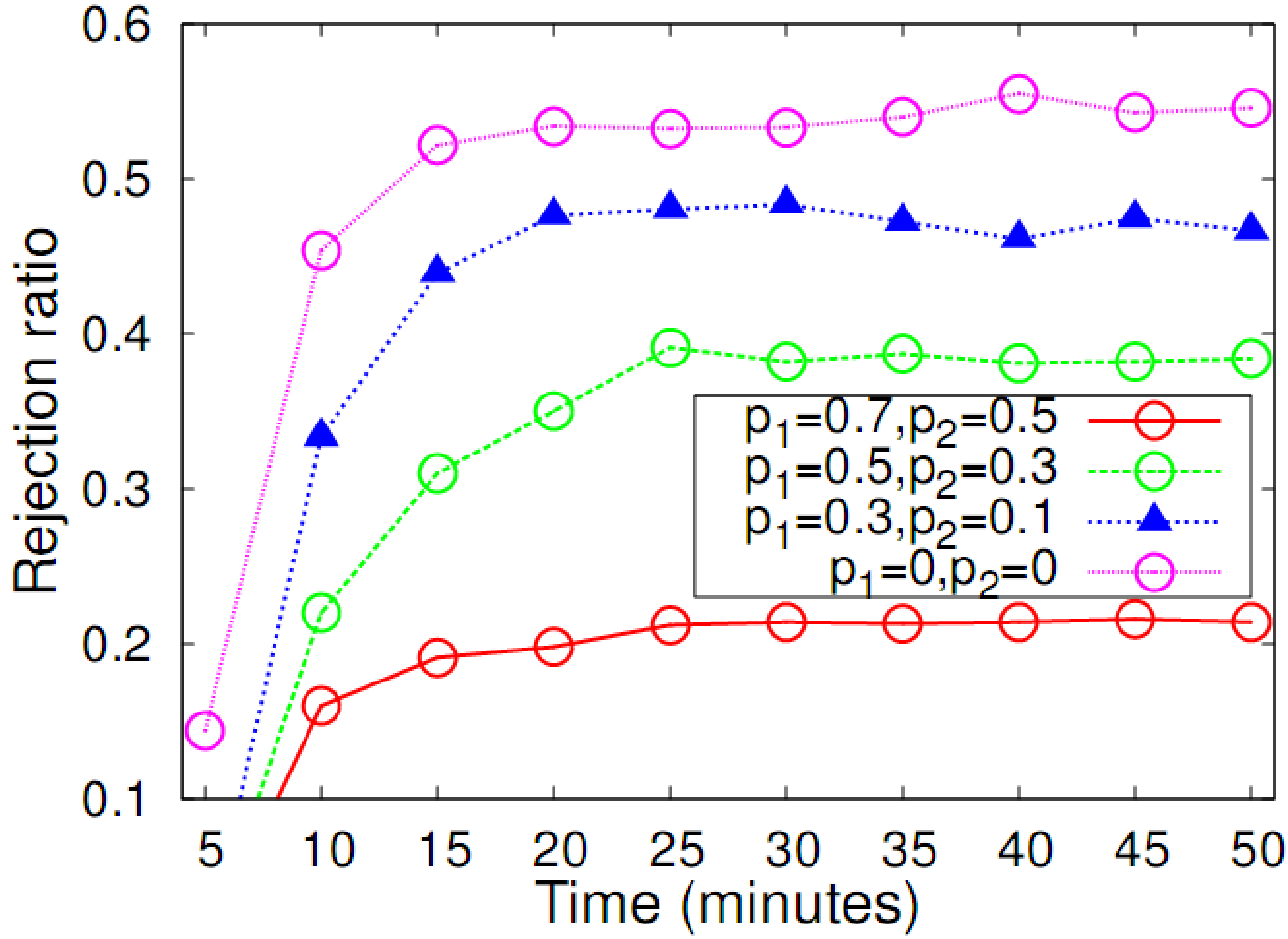


(a) Rejection ratio with varying p_{th}



(b) Rejection ratio with varying Avg

Simulation Results (3/3)



Conclusions

- Dynamic resource demand model
  - Allows a tenant to flexibly control the trade-off between application performance and placement cost
- Work-conserving allocation (WCA) algorithm
  - VM mapping: maximum-first fashion
  - VL mapping: shortest path + adaptive path splitting
  - Local resource sharing: bin-packing
- Evaluations confirm the advantages of WCA.

Thanks for your attention!