# Network and Cloud Resource Packing Games

**Azer Bestavros**

Computer Science Department
Boston University

Joint work with

Jorge Londono (BU→U Pontificia Bolivariana), Vatche Ishakian (BU), Ray Sweha (BU), Nikos Laoutaris (BU→Telefonica), and Shanghua Teng (BU→USC)

http://www.cs.bu.edu/groups/wing

Technical University of Berlin & Deutsche Telekom Labs
June 22, 2010

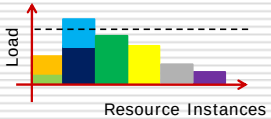
## Pay as you go + Autonomy = Market

- Not your father's Internet
  - Tenants make resource acquisition/control decisions and have no incentive to optimize for, or be fair/friendly to others
  - Infrastructure owners have no incentive to minimize cost for tenants
  - Net neutrality implies no policing
- Holistic system (social) view is passé
  - Challenge is to design the right mechanisms that enable an efficient marketplace

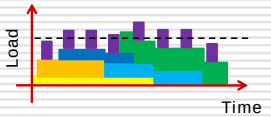


## Resource Packing Problems

- Spatial Packing (where)
  - Bin Packing
  - Load Balancing
  - Multiprocessor scheduling




- Temporal Packing (when)
  - Statistical Multiplexing
  - Real-Time Scheduling
  - Traffic Shaping




- Spatio-Temporal Packing (both)

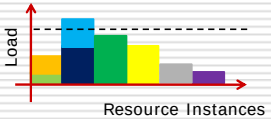


## Talk overview: Two settings

### Cloud resource acquisition

- Colocation Games

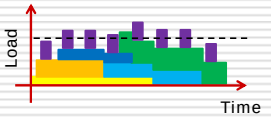
  *A spatial packing game*



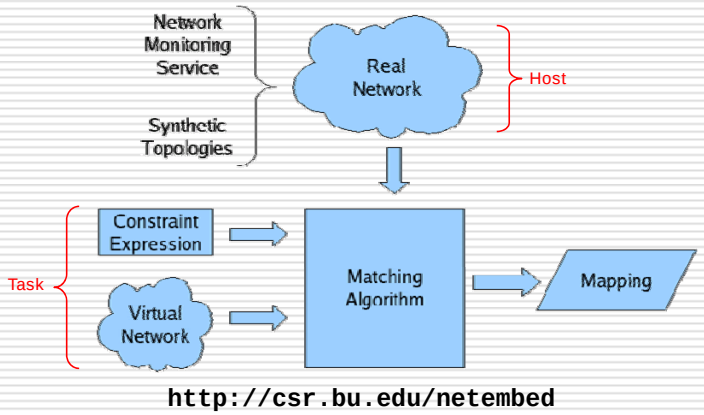

### Shared bandwidth arbitration

- Trade & Cap

  *A temporal packing game*



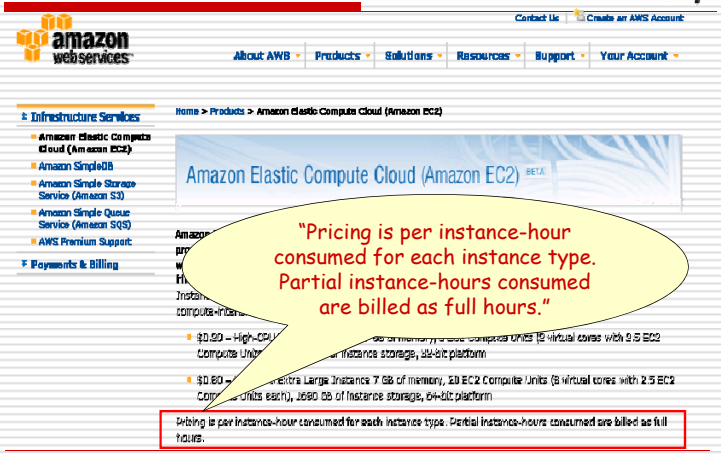
## Spatial Packing: Network Embedding




**http://csr.bu.edu/netembed**



## Motivation: IaaS pricing



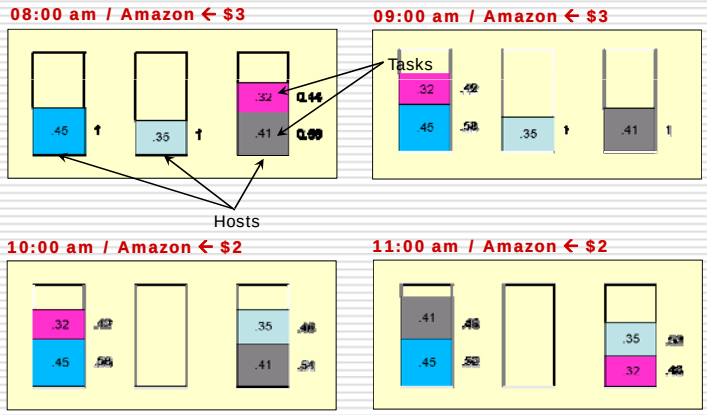



## (Cloud) Colocation Games

- IaaS cloud providers offer fixed-sized instances for a fixed price
- Provider's profit = number of instances sold; no incentive to colocate customers
- Virtualization enables colocation to reduce costs without QoS compromises
- Customers' selfishness reduces the colocation process to a strategic game



## Colocation Games



## Colocation Games: Questions

- Does it reach equilibrium?
- If so, how fast?
- If so, at what price (of anarchy)?
- How about multi-resource jobs/hosts?
- How about multi-job tasks?
- How about job/host dependencies?
- How could it be implemented?
- How would it perform in practice?



## How do we depart from prior work?

- Vickrey-style auctions work†
  - Assumes supply < demand
  - Takes a social perspective
  - Offers a strategy-proof solution
  - Requires central authority
  - Susceptible to collusion

† A. Young, B. Chun, A. Snoeren, and A. Vahdat. *Resource allocation in federated distributed computing infrastructures*. In OS/architectural support for on-demand IT infrastructure, 2004.



## How do we depart from prior work?

- Cooperative cost-sharing games†‡#
  - Find coalition where nobody gains by leaving
  - Computationally hard
  - Applied to best-effort routing problems
  - Player cost not use based; unjustifiable

† V. Misra, S. Ioannidis, A. Chaintreau, and L. Massoulié. *Incentivizing Peer-Assisted Services: A Fluid Shapley Value Approach*. In SIGMETRICS 2010.
‡ H. Chen and T. Roughgarden, T. *Network design with weighted players*. In SPAA 2006.
# E. Anshelevich, A. Dasgupta, J. Kleinberg, E. Tardos, T. Wexler, and T. Roughgarden. *The price of stability for network design with fair cost allocation*. In FOCS 2004.



## Colocation Game: Model

- A hosting graph $G=(V,E)$
  - $V$ & $E$ labeled by capacity vector $R$ and fixed price $P$
- A set of task graphs $T_i=(V_i,E_i)$
  - $V_i$ & $E_i$ labeled by a utilization vector $W$
- Valid mappings
  - $V_i \rightarrow V$ & $E_i \rightarrow E$: $\Sigma W \leq R$ ; supply meets demand
- Shapley Cost function
  - Cost $P$ of a resource is split among tasks mapped to it in proportion to use

$$c_{Sh}(T_i)=\sum_{j<|V_i,E_i|} P_j \frac{w_{ij}}{U_j}$$

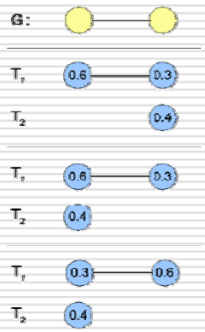
## The General Colocation Game (GCG)

- GCG is a pure strategies game:
  Each task is able to make a (better response) "move" from a valid mapping $M$ into another $M'$ so as to minimize its own cost
- Example applications:
  - Overlay reservation, e.g., on PlanetLab
  - CDN colocation, e.g., on CloudFront



## General Colocation Game: Properties

- GCG may not converge to a Nash equilibrium
- Theorem:
  Determining whether a GCG has a Nash Equilibrium is NP-Complete (by reduction to 3-SAT problem)
- Need more structure to ensure convergence






## Colocation Games: Variants

- Process Colocation Game (PCG):
  Task graph consists of a single vertex representing an independent process that needs to be assigned to a single host with only one capacitated resource
- Multidimensional PCG (MPCG):
  Same as PCG but with multi capacitated resources
- Example applications:
  - VM colocation, e.g., on a Eucalyptus cluster
  - Streaming server colocation



## Colocation Games: Variants

- Parallel PCG (PPCG):
  Task graph consists of a set of disconnected vertices (independent processes), each with multidimensional resource utilization needs
- Uniform PPCG:
  Same as PPCG but with identical resource utilization for all processes
- Example applications:
  - Map-Reduce paradigm
  - MPI scientific computing paradigm

## Colocation Games: Theoretical results

- PCG converges to a Nash Equilibrium under better-response dynamics
- PCG converges to a Nash Equilibrium in $O(n^2)$ better-response moves, where $n = |V|$
- Price of Anarchy for PCG is 3/2 when hosting graph is homogeneous and 2 otherwise
- MPCG converges to a Nash equilibrium under better-response dynamics
- Uniform PPCG converges to a Nash equilibrium under better-response dynamics
- ...



## PCG: Better Response

Best-Response moves require knowledge of utilizations of all processes – not practical

### Local Better-Response solution:

1. Select a random target hosting node and obtain process utilizations of all processes on that node
2. Determine if a cost-reducing "legal" move to that node is possible – an NP-hard Knapsack problem
   - Dynamic Programming solution in pseudo-polynomial time for small number (100s) of processes/host [DPKP]
   - Breadth-First branch & bound Search heuristic [BFS]
   - Depth-First branch & bound Search heuristic [DFS]



## PCG: Performance Evaluation

### Workloads

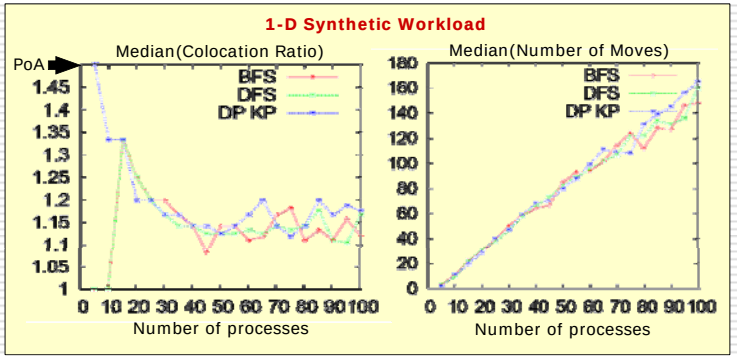
- Trace-driven: CoMon PlanetLab traces
  - Real hosting environment with 3-dimensional resource utilizations
  - Infeasible to compute optimal colocation
- Synthetic
  - Allows systematic exploration of the space
  - Optimal colocation is known by construction

### Metrics (over 100 experiments)

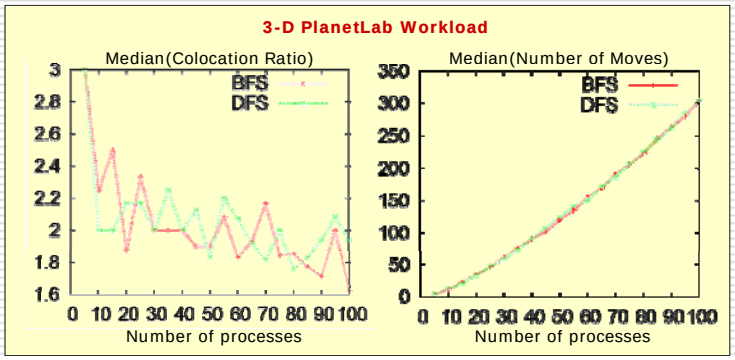
- Colocation Ratio (bounded by PoA)
  - How inefficient is the resulting colocation compared to optimal or best?
- Number of moves (not migrations) until NE is reached
  - How much churn (overhead) to be expected?



## PCG: Synthetic baseline results

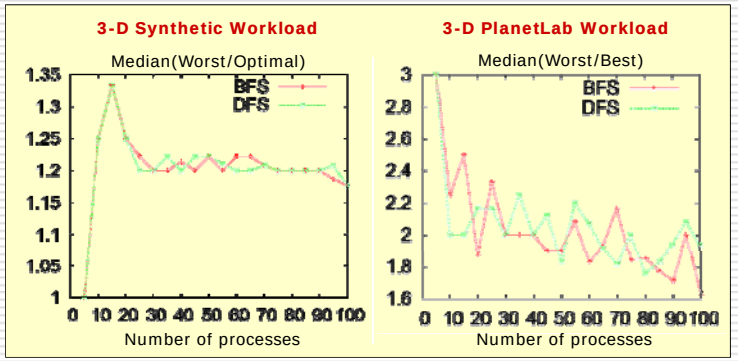


## MPCG: PlanetLab baseline results



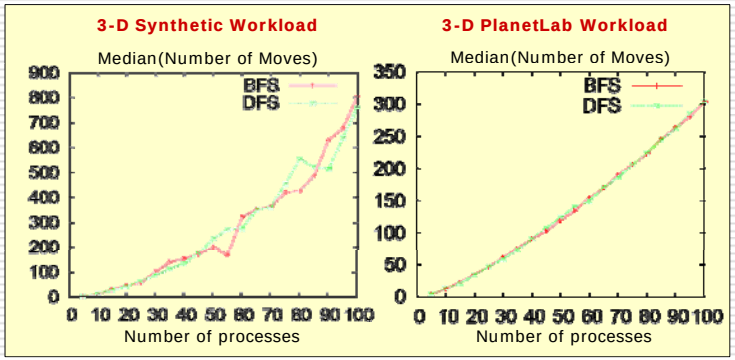



## MPCG: Colocation Ratio






## MPCG: Number of Moves

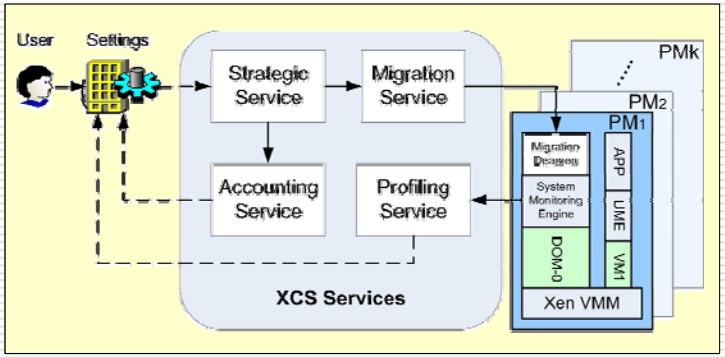





## The CLOUDCOMMONS prototype

- API for Strategic Services
  To facilitate colocation, e.g., allow users to find each other, compute strategic responses, …
- API for Operational Services
  To enforce outcomes of colocation, e.g., migration, reconfiguration, accounting, …
- Implemented over Xen (XCS)

## CloudCommons: Architecture



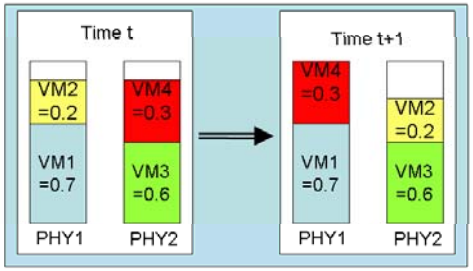



## CloudCommons: Migration Service

- Identify VMs to migrate
  - Minimize number of migrations
  - Minimize amount of data that needs to move
- Determine migration plan
  - Exploit potential for parallelism
  - Minimize need for staging hosts
- Evaluate impact of migration
  - On performance of the migrating VMs
  - On performance of non-migrating VMs



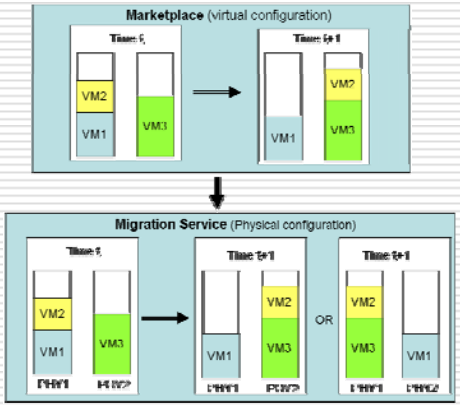
## Need for staging hosts



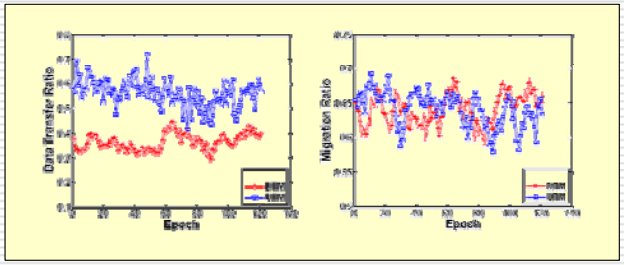

A temporary (staging) host is needed to swap VM2 and VM4



## Need for a migration plan



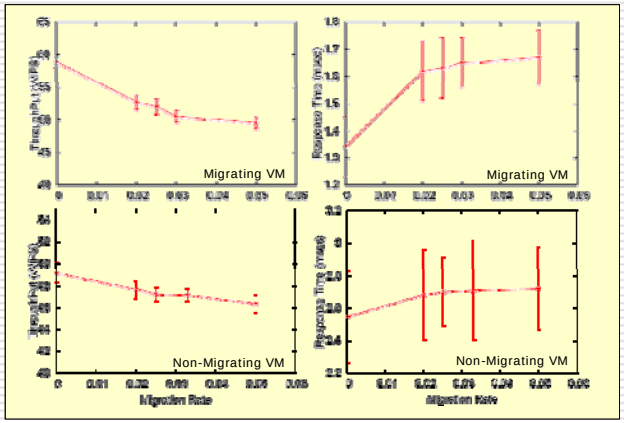
## Data/User Transfer Minimization




**Theorem:** The DTM (UTM) heuristic results in at most twice the amount of data transfer (migrations) incurred by an optimal (NP-hard) algorithm.

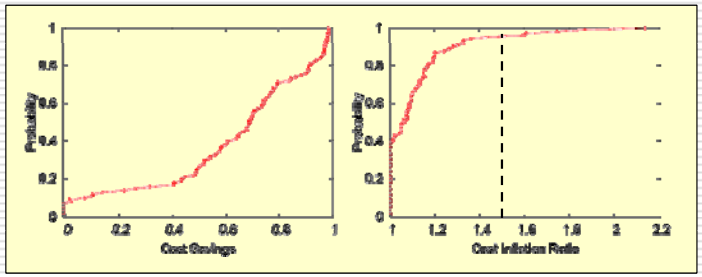


## Impact on TPC-W from migration






## CLOUDCOMMONS: Benefit to users



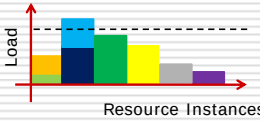

Planet-Lab trace-driven experiments
(Overheads/costs of all XCS services included)



## Talk overview: Two settings

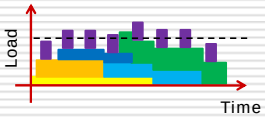
### Cloud resource acquisition

- Colocation Games
  *A spatial packing game*




### Shared bandwidth arbitration

- Trade & Cap
  *A temporal packing game*



## Today's last mile

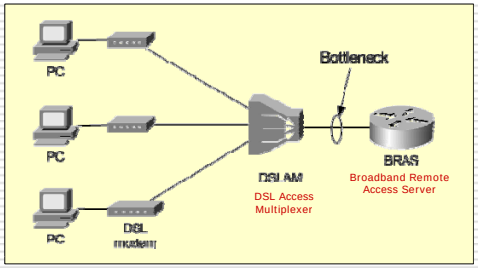



## The perils of the fixed pricing model

- It's here to stay; metered pricing rejected
- Implications:
  - Customer has no incentive to save bandwidth
  - ISP cost depends on peak demand – 95/5 rule
  - Reigning in bandwidth hogs is incompatible with Net Neutrality
- Must devise mechanisms that take ISPs out of the "traffic shaping" business



## DSLAM "last-mile" architecture




Traffic shaping done at BRAS



## Solution: Create a marketplace

- Recognize the two types of user traffic:
  - Interactive Traffic (IT)
    - Browsing, VoIP, Video, Messaging, Gaming, …
    - Limited bandwidth; highly sensitive to response time
  - Fluid Traffic (FT)
    - P2P, Network backup, Netflix/software downloads, …
    - Open-ended bandwidth; less sensitive to response time
- Create a marketplace:
  1. Give users rights to DSLAM bandwidth, and
  2. Let users trade IT & FT allocations over time

## The Marketplace

- Each user gets a fixed budget per epoch
  - Budget proportional to level of service
  - An epoch is a fixed number of time-slots, e.g., 1 day = 288 5-min slots
- Trade & Cap
  - User engages in a pure strategies game that yields a schedule for its IT sessions
  - User acquires as much FT bandwidth as its remaining budget would allow



## Trading Phase: Strategy Space

- Session:

  An IT session is the sequence of slots during which an IT application is active
- Slack:

  User may have flexibility in scheduling IT sessions; slack specifies the number of slots that an IT session is allowed to be shifted back/forth
- Strategy Space:

  The set of all possible arrangements of IT sessions within allowable slack define the strategy space for a user



## Trading Phase: Cost Function

- Let $x_{ik}$ be the bandwidth used in slot $k$ by a chosen IT session schedule for user $i$.
- The cost incurred by user $i$ is given by:

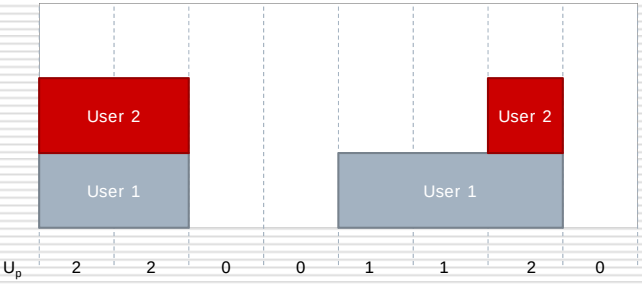
$$c_i = \sum_{k \in slots} x_{ik} \cdot U_k = \sum_{k \in slots} x_{ik} \left( \sum_{j \in users} x_{jk} \right)$$

- Cost of user $i$ depends on the choices made by other users – hence the game!

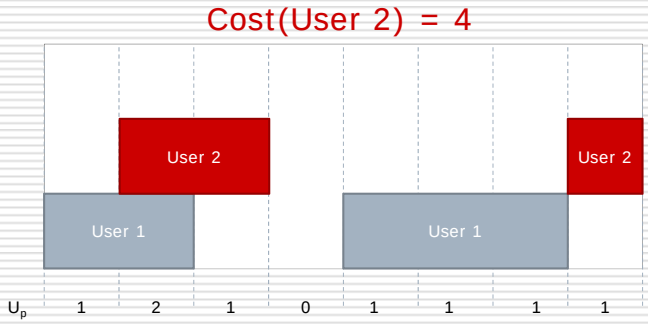


## Trading Phase: Illustration

Cost(User 2) = 6



## Trading Phase: Illustration






## Trading Phase: Best Response

- BR of user $i$ is the schedule of IT sessions that minimizes its cost $c_i$
- Computing BR is NP-hard, equivalent to solving a generalized knapsack problem
- Dynamic programming solution is pseudo-polynomial in the product of the number of sessions and number of slots
- Scales well for all practical settings – 100s of users and 100s of slots



## Trading Phase: Findings

- Provably converges to Nash Equilibrium, even in presence of constraints
- For $n$ users, Price of Anarchy is $n$, but in practice below 2, especially for $n$>10
- Experimentally, large reduction of peak utilization, even with small flexibility



## Capping Phase: Best Response

- BR of user $i$ is to maximize total FT allocation

$$w_i = \sum_{k \in slots} w_{ik}$$

subject to the budget constraint

$$\sum_{k \in slots} w_{ik} \cdot \left( U_p + \sum_{j \in users} w_{jk} \right) = B_i - c_i$$

## Capping Phase: Budget

- Let $V$ be an upper-bound on traffic per slot
- The ISP sets a target capacity $C = V/R$, where $R \geq 1$ reflects its "resistance" to traffic
- The ISP allocates $C$ in some proportion (e.g., equally) to all users over all slots
- This constitutes the budget $B$ assigned to a user over an epoch

$$B = \frac{C}{n} \cdot T$$



## Capping Phase: Findings

- Computing BR is efficient using Lagrange Multipliers method
- Provably, converges to a unique global (social) optimum that maximizes the FT allocations of all users
- Experimentally, smoothes the aggregate IT+FT traffic to any desirable level controlled by resistance parameter R



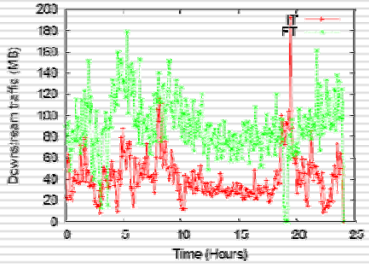
## Experimental Evaluation

### Workload

Derived from WAN traces of MAWI project

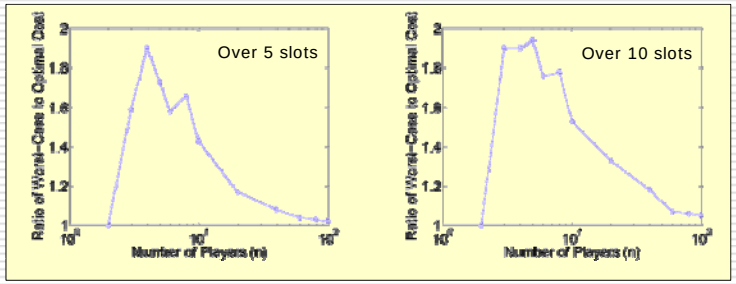
- Identify users from volume and direction of flows to known ports (e.g., most traffic destined to port 80)
- Identify user IT sessions using thresholds on per-IP traffic intensities over time
- Slack introduced using various models (e.g., fixed, proportional, etc.)

| Period | 2008-03-31 08:00 – 2008-03-31 23:59 |
|---|---|
| Total packets | 1,351,089,843 |
| TCP packets | 1,194,699,655 |
| UDP packets | 4,921,832 |
| Total TCP bytes (payload) | 924,549,189,860 |






## Trading Phase: Experimental PoA

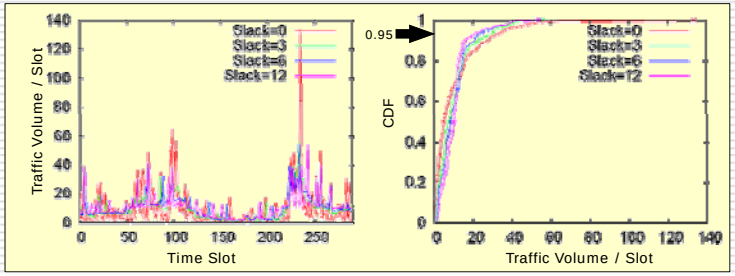



Theoretical PoA is $n$ but not in practice

## Trading Phase: Smoothing effect

Value proposition to ISPs

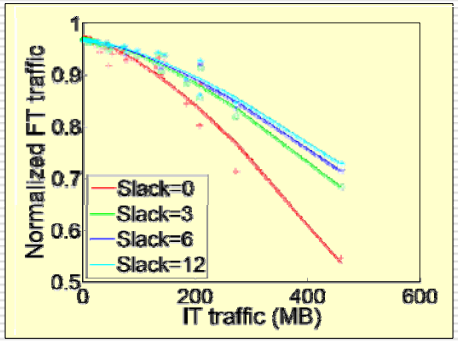
| Max Slack | Reduction in 95% |
|---|---|
| 3 | 15% |
| 6 | 24% |
| 12 | 31% |






## Trade & Cap: Flexibility pays off!

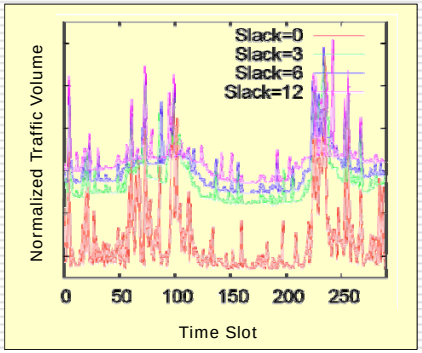
Value proposition to customers






## Trade & Cap

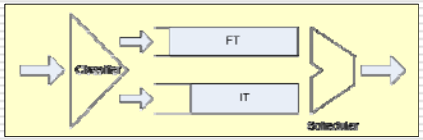
A win-win for ISPs and customers





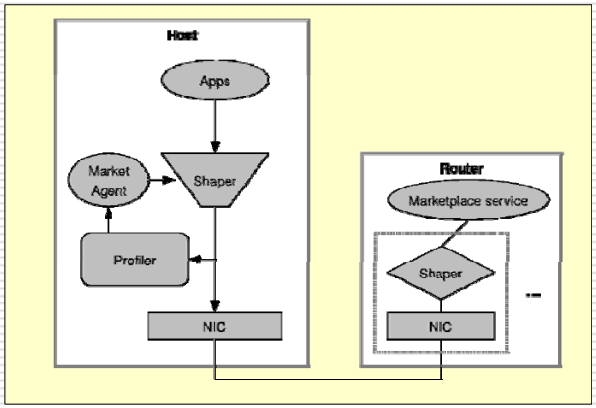

## Trade & Cap: Implementation

- On Client Side (DSL Modem):
  - + Strategic agent to execute Trade & Cap
  - + Operational service to classify and schedule




- ISP Side (DSLAM or BRAS):
  - + Support exchange between strategic agents
  - + Enforce traffic/slot/user from Trade & Cap

## Trade & Cap: Implementation






## Conclusion

- In many settings, resource management must be seen as a strategic game among peers or tenants of an infrastructure
- By setting up the right mechanism, one can ensure convergence and efficiency
- New services are needed to support strategic and operational aspects of these game-theoretic mechanisms



## Publications

"netEmbed: A service for embedding distributed applications (Demo)". Londono and Bestavros. *ACM/Usenix Middleware'07*.

"netEmbed: A resource mapping service for distributed applications". Londono and Bestavros. *IEEE/ACM IPDPS'08*.

"Colocation games with application to distributed resource management". Londono, Bestavros, and Teng. *USENIX HotCloud'09*.

"*Colocation as a Service: Strategic & operational cloud colocation services*". Ishakian, Sweha, Londono, and Bestavros. *IEEE NCA'10*.

"Trade & Cap: A customer-managed system for trading bandwidth at a shared link". Londono, Bestavros, and Laoutaris. *BUCS-TR-2009-025*.



## Network and Cloud Resource Packing Games

**Azer Bestavros**

Computer Science Department
Boston University

Joint work with

Jorge Londono (BU→U Pontificia Bolivariana), Vatche Ishakian (BU), Ray Sweha (BU), Nikos Laoutaris (BU→Telefonica), and Shanghua Teng (BU→USC)

http://www.cs.bu.edu/groups/wing

Technical University of Berlin & Deutsche Telekom Labs
June 22, 2010