Network and Cloud Resource Management Games

Azer Bestavros

Computer Science Department
Boston University

Joint work with

J. Londono (BU), V. Ishakian (BU), G. Smaragdakis (BU→Deutsche Telecom),
N. Laoutaris (BU→Telefonica), S. Teng (BU→USC), J. Byers (BU), P. Michiardi (Eurecom),
V. Lekakis (U. Crete→ U Maryland) and M. Roussopoulos (Harvard→FORTH)

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

Texas State University, San Marcos
April 19, 2010

Pay as you go + Autonomy = Market

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

Talk overview: Three settings

Overlay network connectivity management
- Selfish Neighbor Selection (SNS) game

Cloud resource acquisition
- Colocation Games

Shared bandwidth arbitration
- Trade & Cap

Overlay connectivity management

- Neighbor selection is a key building block for overlay applications
- Love your neighbors as yourself (assuming you can't easily move)
- Changing neighborhood in overlays is cheap; just rewire!
- Implications?

Choosing thy neighborhood game

- Given an established overlay network
  - A node evaluates the advantage (if any) from picking a different set of neighbors
  - If rewiring is warranted, the node changes its (outbound) neighbors accordingly
  - This rewiring may trigger more rewiring by other nodes

and the "Selfish Neighbor Selection" (SNS) game goes on…

SNS Game: Interesting questions

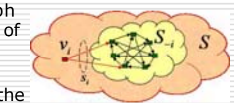
- What is the optimal strategy? How does it compare to empirical ones (e.g., random)?
- Under what conditions will neighborhoods stabilize, i.e., reach Nash-like equilibrium?
- What do the resulting Nash equilibria look like?
- What is the price of anarchy?
- What if some (most) nodes are naïve? malicious? or adversarial?
- What is the impact of partial knowledge, of churn, and of changes in physical network?
- Could answers to the above questions inform systems/protocol design?

SNS: Target applications

- Routing Networks (e.g., Skype):
  - Send unicast traffic from one overlay node to another
  - Node's objective is to minimize its average (or maximum) routing cost to all destinations
- Broadcast Networks (e.g., MS update):
  - Send data from one node to all nodes in the overlay
  - Node's objective is to minimize its average (or maximum) broadcast cost to all destinations
- Query Networks (e.g., Gnutella):
  - Find content available in some (unknown) overlay node
  - Node's objective is to query the most number of overlay nodes using scoped flooding

Formulation of SNS for routing

- S_{-i} is the residual wiring graph defined by the local wirings of all nodes except node v_i



- Best-Response (BR) of v_i is the local wiring s_i that minimizes

$$C_i(S) = C_i(S_{-i} \cup \{s_i\}) = \sum_{v_j \in V_{-i}} p_{ij} \cdot d_S(v_i, v_j)$$

$$|s_i| \leq k$$

where:

- p_{ij} is the preference of v_i for destination v_j
- $d_S(v_i, v_j)$ is the cost of the shortest path from v_i to v_j in S

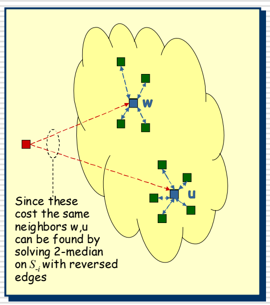
How we depart from prior work?

- Selfish routing†
  - Game input: Fixed network topology
  - Game outcome: Selfishly constructed source-based routes over the topology
- ➔ Our SNS work:
  - Game input: Shortest-path routing
  - Game outcome: Selfishly constructed network topology

† References: [Roughgarden & Tardos, JACM'02] [Qiu *et al*, Sigcomm'03]

Best Response for SNS is NP hard

- Theorem:
  Under uniform overlay link weights (e.g., hop-count), finding BR to S_{-i} is equivalent to solving the asymmetric k-median on S_{-i} with reversed edges
- Corollary:
  Constant approximation with an $O(\log n)$ blow-up in k is possible [Lin and Vitter, '92]



Game theoretic results for SNS†

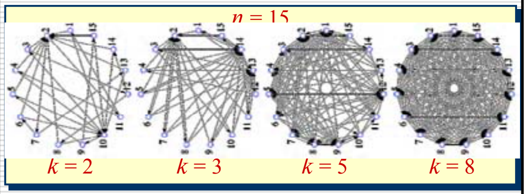
- Theorem:
  All games with uniform node preference, node degree, and link costs have pure Nash equilibria (stable graph).
  - In any such stable graphs, the cost of any node is at most $2 + k^{-1} + O(1)$ that of any other node.
  - The diameter of the stable graph for a uniform game is $O(\mathrm{sqrt}(n \log_k n))$.
- Theorem:
  There exist non-uniform games with no pure Nash equilibria.

† Proofs, constructions, and more results in *Laoutaris, Rajaraman, Sundaram, Teng, "A bounded-degree network formation game"*, from arXiv-CoRR cs.GT/0701071.

Topology of stable (NE) SNS graphs

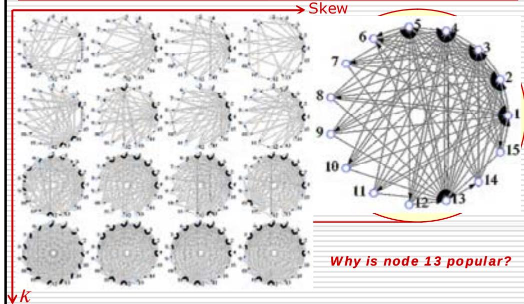
- Under unit link costs and uniform routing preference to all destinations, we know that a Nash-equilibrium exists.
- What are the characteristics of the resulting wiring graphs?
  - Are they random?
  - Do they exhibit a uniform in-degree distribution?

Selfishness yields skewed attachment



- Not uniform, but skewed in-degree distribution
- Selfishness yields preferential attachment to "accidentally" popular nodes
- Phenomenon more evident for small k/n – why?

… distinct from skew in preferences



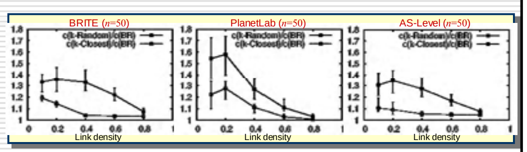
Results in non-uniform networks

- Link cost generation
  1. Synthetically using BRITE:
     - Barabasi-Albert (BA) model with heavy-tailed 2D placement
     - Euclidean distance used to derive cost of overlay links
  2. Empirically from PlanetLab:
     - 300-node PlanetLab topology
     - All-pair ping traces used to derive cost of overlay links
  3. Empirically from AS-level maps:
     - 12/2001 Rocket-Fuel data of the Internet topology
     - AS-level hop-count used to derive cost of overlay links
- Control parameter
  - Bound on out-degree (k) ≈ link density (β)

How should a new comer connect?

- Neighbor selection strategy
  a. The k-random heuristic
  b. The k-closest heuristic, a.k.a. greedy
  c. SNS Best Response (BR) wiring using ILP
- Experiments done in nine permutations
  - Three strategies for a new comer, each assuming residual graph was wired using one of the three strategies
- Performance metrics
  - Individual Cost = Average cost for a newcomer
    - ➔ Cost ratio for strategy x = C(x)/C(BR)
  - Social Cost = Sum of cost for all nodes
    - ➔ Social Cost ratio for strategy x = SC(x)/SC(BR)

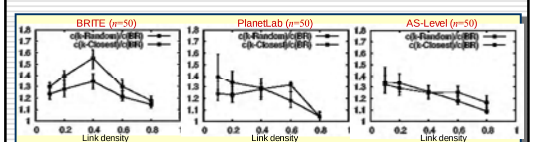
SNS over random residual networks



➔ BR is dominant, with k-closest decidedly better than k-random. BR's benefit pronounced for small k – why?

If your neighbors are naïve, it pays to be selfish

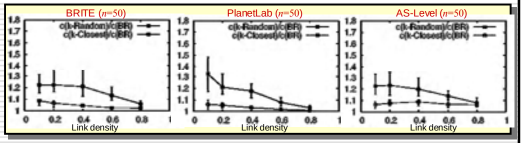
SNS over greedy residual networks



➔ BR is dominant, with k-random slightly better than k-closest – why?

If your neighbors are greedy, it pays to be selfish

SNS over selfish residual networks



- ➔ BR is dominant, but not by a significant margin, with k-closest being quite competitive – why?

If your neighbors are selfish, it's OK to be naïve

Social cost benefit from SNS

n=50	$\beta = 0.1$		$\beta = 0.2$	
	k-Random/BR	k-Closest/BR	k-Random/BR	k-Closest/BR
BRITE	1.44	1.53	1.52	1.84
PlanetLab	2.23	1.48	1.75	1.23
AS-level	2.04	1.90	1.83	1.61

- ➔ Adopting BR as a neighbor selection strategy results in a significant reduction in the social cost (by 30-60%) over naïve (random/greedy) approaches.

The network is better off with selfish nodes!

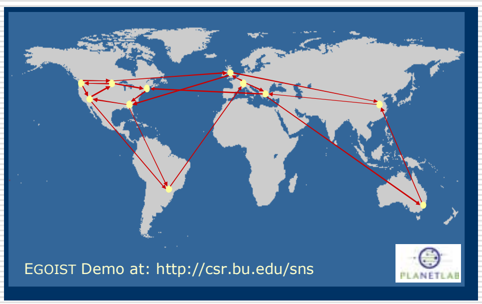
EGOIST: Implementation

Protocol for EGOIST overlay node v_i

1. Bootstraps by connecting to arbitrary neighbors
2. Joins link-state protocol to get residual graph
3. Measures cost to candidate neighbors
4. Wires according to chosen strategy (default: BR)
5. Monitors and announces overlay links

† We have also implemented a light-weight version of this protocol, in which steps 2, 4, and 5 are implemented on a central server.

EGOIST: SNS prototype



EGOIST Demo at: http://csr.bu.edu/sns

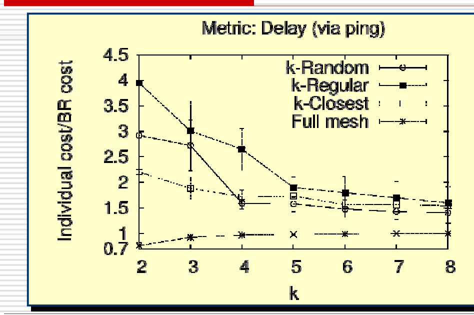
EGOIST: Features

Supported metrics:
- Delay (actively/passively monitored with `ping/Pyxida`)
- Available bandwidth (monitored with `pathChirp`)
- Node load (monitored with `loadavg`)

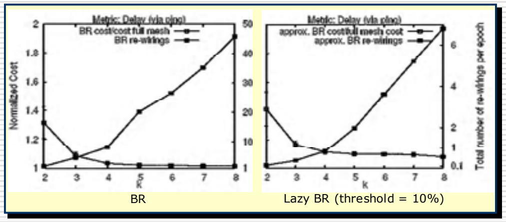
Supported wiring strategies:
- k-random
- k-closest
- k-regular
- Best-Response (Delay and AvailBw formulations)
- Hybrid Best-Response (subset of links donated to the network)

BR Computation:
- By using the full residual graph
- By sampling the residual graph

EGOIST: Baseline results (n=50)

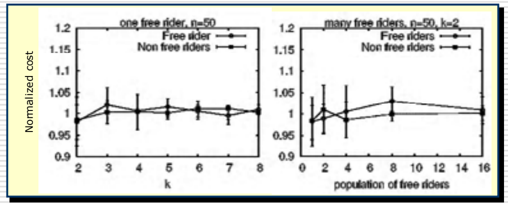


EGOIST: Marginal utility of re-wiring



- Most of the benefit achieved with $k \sim 3\text{-}4$
- Re-wirings could be reduced using "lazy" BR

EGOIST: Vulnerability to abuse



- Free riders avoid being chosen as neighbors by inflating cost of their outgoing links (*2 above)
- EGOIST is robust to abuse by free riders (not the case with greedy neighbor selection)

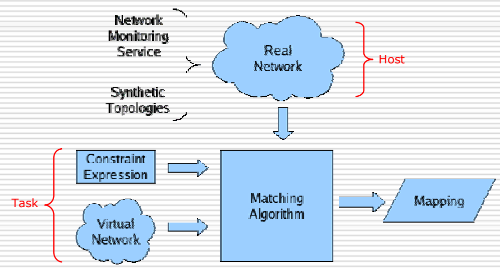
Talk overview: Three settings

Overlay network connectivity management
- Selfish Neighbor Selection (SNS) game

Cloud resource acquisition
- Colocation Games

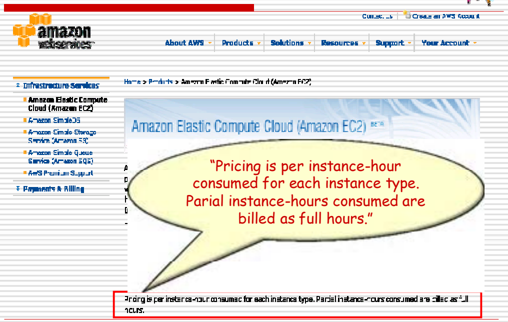
Shared bandwidth arbitration
- Trade & Cap

Network Embedding



http://csr.bu.edu/netembed

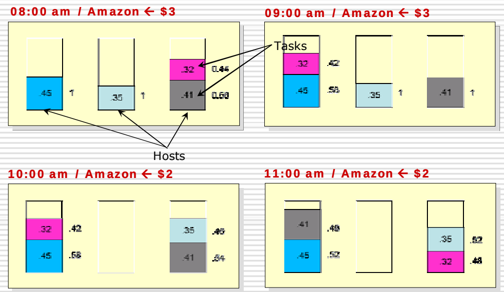
Motivation: IaaS pricing



(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? 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

† Chen, H.-L. & 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

$$c_M(T_i) = \sum_{j \subset \{V_i, E_i\}} P_j \frac{w_{ij}}{U_j}$$

The General Colocation Game (GCG)

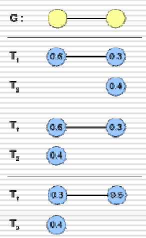
- GCG is a pure strategies game:

  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 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

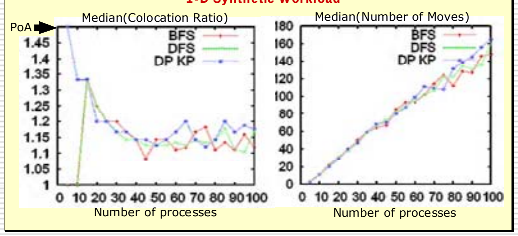
- Trace-driven: CoMon PlanetLab traces
  - Real hosting environment with 3-dimensional resource utilizations
  - Infeasible to compute optimal colocation
- Synthetic
  - Allows systematic exploration of the space
  - Optimal colocation is known by construction

Metrics (over 100 experiments)

- Colocation Ratio (bounded by PoA)
  - How inefficient is the resulting colocation compared to optimal or best?
- Number of moves until NE is reached
  - How much churn (overhead) to be expected?

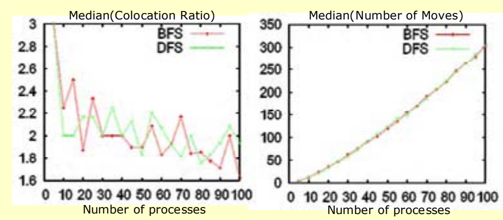
PCG: Synthetic baseline results

1-D Synthetic Workload



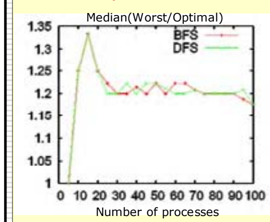
MPCG: PlanetLab baseline results

3-D PlanetLab Workload

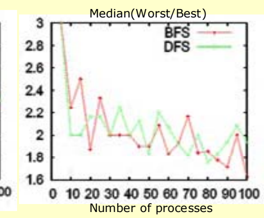


MPCG: Colocation Ratio

3-D Synthetic Workload

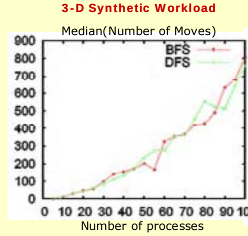


3-D PlanetLab Workload

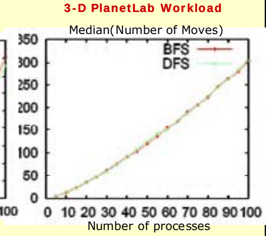


MPCG: Number of Moves

3-D Synthetic Workload



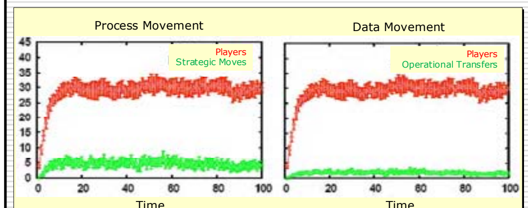
3-D PlanetLab Workload



The CLOUDCOMMONS prototype

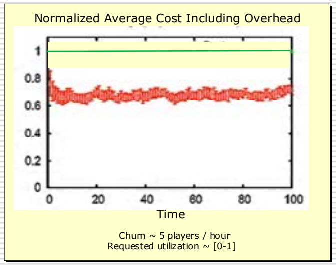
- API for Strategic Services
  To facilitate colocation, e.g., allow users to find each other, compute strategic responses, …
- API for Operational Services
  To enforce outcomes of colocation, e.g., migration, reconfiguration, accounting, …
- Implemented over Xen

CLOUDCOMMONS: Sample Results



Churn ~ 5 players / hour
Requested utilization ~ [0-1]

CLOUDCOMMONS: Sample Results



Talk overview: Three settings

Overlay network connectivity management
- Selfish Neighbor Selection (SNS) game

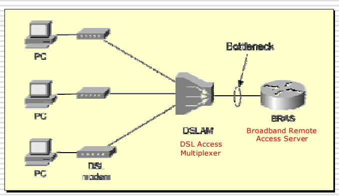
Cloud resource acquisition
- Colocation Games

Shared bandwidth arbitration
- Trade & Cap

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 <u>budget</u> per <u>epoch</u>
  - 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

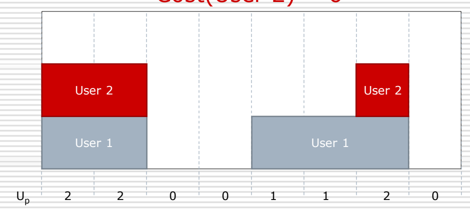
- Let x_{ik} be the bandwidth used in slot k by a chosen IT session schedule for user i.
- The cost incurred by user i is given by:

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

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

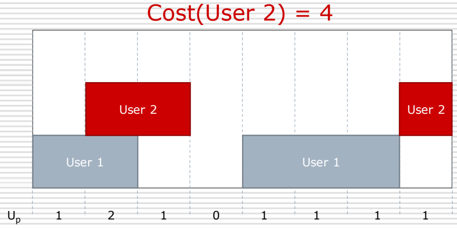
Trading Phase: Illustration

Cost(User 2) = 6



Trading Phase: Illustration

Cost(User 2) = 4



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

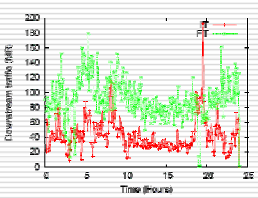
Experimental Evaluation

Workload

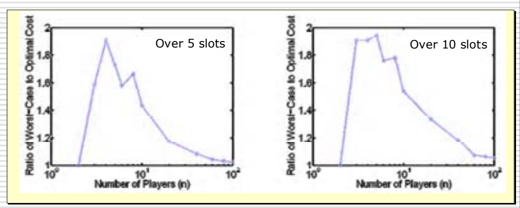
Period	2009-03-31 00:00 – 2009-03-31 23:59
Total packets	1,551,089,845
TCP packets	1,194,409,653
UDP packets	4,321,852
Total TCP bytes (payload)	924,540,189,060

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.)



Trading Phase: Experimental PoA

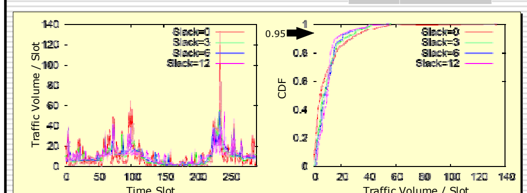


Theoretical PoA is n but not in practice

Trading Phase: Smoothing effect

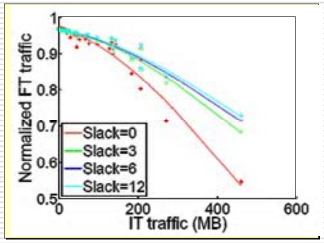
Value proposition to ISPs

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



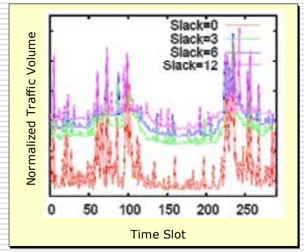
Trade & Cap: Flexibility pays off!

Value proposition to customers



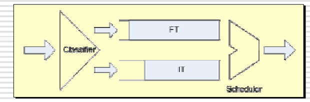
Trade & Cap

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

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

"Implications of selfish neighbor selection in overlay networks". Laoutaris, Smaragdakis, Bestavros and Byers. *Infocom'07*.

"Swarming on optimized graphs for n-way broadcast". Smaragdakis, Laoutaris, Michiardi, Bestavros, Byers, and Roussopoulos. *Infocom'08*.

"EGOIST: Overlay routing using selfish neighbor selection". Smaragdakis, Lekakis, Laoutaris, Bestavros, Byers, and Roussopoulos. *ACM CoNEXT'08*.

"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. *BUCS-TR-2010-003*.

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