Diffusing Focused Loads in Networks using Pricing

Balaji Prabakar
Kevin Leyton-Brown
Ryan Porter
Yoav Shoham
Shobha Venkataraman

Computer Science Department
Stanford University
Focused Loading

- Many users demand network resources at some focal time, predictable in advance
- Canonical example: long distance phone
  - people want to talk as early as possible, minimize cost
  - utility maximized when rates drop at 5 PM: network demand spikes
- Computer networks: load can be even more focused
  - sudden onset: TicketMaster server as tickets go on sale
  - deadline: IRS server just before taxes are due
Managing Network Congestion

• Share bandwidth fairly, even when agents may act selfishly to maximize bandwidth available to them

• Technological: isolate packet flows
  – problem: difficult to implement

• Economic: give agents incentives
  – Smart Market: use bids to set price for network usage at each time slot [MacKie-Mason and Varian; Gibbens, Kelly, Key]
  – Paris Metro Pricing: partitions of the network that differ only in price [Odlyzko; Altmann’s system from 1st talk]
Diffusing Focused Loads

• Existing schemes are not designed to deal gracefully with sudden changes in load
  – technological: queues may be overwhelmed, leading to many dropped packets and degraded service for everyone
  – Smart Market will suddenly charge unpredictably higher prices
  – Paris Metro Pricing assumes that users have enough information about current load to choose the right class of service

• Rather than trying to decide which packets to drop, give an incentive for smoothing out the demand
  – possible because focused loads are predictable by definition
  – knowledge about utility functions means more revenue; more modest computational demands
Outline

1. Our game-theoretic model
2. A simple mechanism: “Matching Pennies”
3. A more complex mechanism: “Collective Reward”
4. Future directions

Warning: the length of this talk forces me to gloss over many details. More formal models and analysis are provided in our paper.
Our Model

• Network use is divided into $t$ timeslots
• $n$ risk-neutral agents will use the network for one time slot each
• Each slot has a fixed usage cost $m$
• Agent $a_i$’s valuation for slot $s$ is given by $v_i(s)$
• $d(s)$ is the number of agents who choose slot $s$
• Give agents an incentive to balance load
  – waive the usage fee for slot $s$ with probability $p(s)$
  – agents made aware of the mechanism, including how $p$ is calculated, but not of the actual draws from $p$
Agents, Equilibria

- Agents act to maximize their own utility
  - agent’s action: choosing a slot
  - agent’s strategy: a probability distribution over slot choices
  - $a_i$’s utility for choosing slot $s$ is $u_i(s) = v_i(s) - (1-p(s))m$
  - only consider mechanisms where participation is rational for all agents

- Nash equilibrium for a mechanism $\Phi$:
  - a set of strategies for the agents participating in $\Phi$ where no single agent $a_i$ can benefit from changing his strategy, given that all other agents’ strategies as fixed
  - strict equilibrium: $a_i$ is always worse off changing strategy
  - weak equilibrium: $a_i$ is never better off changing strategy
Mechanism Evaluation, Optimality

• Mechanism $\Phi$ has two goals:
  1. balance load caused by the agents’ selection of slots
     – $g(d)$: the monetary value of $d$ to the network
  2. maximize expected revenue
     – depends on $\Phi$ and $d$: $E[R|\Phi,d]$

• Trade-off between load balancing and revenue
  – load balancing is achieved by offering free slots
   – $z(\Phi,d) = g(d) + E[R|\Phi,d]$

• Optimality of a mechanism-equilibrium pair
  – $z$ maximal as compared to $z$ for all other equilibria of other mechanisms (constant $n$, participation rational)
Our Mechanisms

• I’ll describe two in some detail; two more in our paper
• Why more than one mechanism? Many variables:

<table>
<thead>
<tr>
<th>Type of equilibrium or strategy</th>
<th>Payment only after all slots?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Time cost of coordination phase</td>
<td>Non-optimal equilibria exist?</td>
</tr>
<tr>
<td>Time cost after coordination</td>
<td>Revenue increases if agents deviate?</td>
</tr>
<tr>
<td>Storage cost</td>
<td>Harmful collusion?</td>
</tr>
<tr>
<td>Communication cost</td>
<td>Irrational actions harm other agents?</td>
</tr>
<tr>
<td>Requires agent names?</td>
<td>Agents may have different ( v ) functions?</td>
</tr>
</tbody>
</table>

• To begin with, I’ll add two assumptions:
  1. all agents have the same preferences for slots
  2. mechanism designer knows these preferences
“Matching Pennies”

1. Decide if each slot will be free according to $p$
2. Each agent chooses a slot

Select $p$ so that agents are indifferent between all time slots:

- i.e., $E[u_i]$ constant for all slots
- we’ll call this probability distribution $p^*$
MP: Equilibria

• *Any* set of strategies is a weak equilibrium, e.g.:
  – agents randomize (load balancing)
  – agents pick the “best” slots deterministically: maximize $z$
    • this is a weak, optimal equilibrium
  – agents pick *same* slot deterministically: focused loading!

• Theorem: if
  – agents have identical utility functions
  – payoffs are *independent* of agents’ moves
then a strict, optimal equilibrium *does not exist.*
“Collective Reward”

1. The mechanism assigns agents “names” corresponding to slot numbers
2. Each agent chooses a slot
3. The mechanism computes $p$, and determines which slots will actually be free
   
   - $\text{count}(s)$: the number of agents given name $s$
   - $d^+(s) = |\text{count}(s) - d(s)|$
   - $S$: the set of slots which minimize $d^+$

\[
p(s) = \begin{cases} 
    p^*(s) & s \in S \\
    0 & s \not\in S 
\end{cases}
\]
CR: Equilibrium $\varphi$

- A strict equilibrium: $a_i$ chooses slot $name(i)$
- All other agents play this strategy—$a_i$ could:
  1. play the strategy too
     - $d^+$ is minimized by all slots
     - $a_i$ gets the same utility regardless of her name
  2. select a different slot
     - $a_i$’s slot will never be free
     - if expected utility for cooperation exceeds $v(bestslot)$, deviation is unprofitable, and $\varphi$ is a strict equilibrium
CR: Choosing Names, Optimality

- Problem: we want to assign names to agents before we know how many agents will participate
- Theorem: assigning each agent the name that greedily improves $z$ gives rise to optimal $d$
- Theorem: (CR, $\phi$) is optimal
  - an optimal distribution of agents may be achieved
  - agents can be paid the minimum needed to make deviation unprofitable
CR: Bounds on Utility Functions

- Relax our assumptions:
  1. agents have different preferences for slots
  2. mech. doesn’t know agents’ preferences, knows bounds: $v^l$ and $v^u$
     - impossible to construct optimal mechanisms in this case

- $k$-Optimality of a mechanism-equilibrium pair
  - $z$ is no further than $kn$ from its maximal value

- CR is $k$-optimal, $k = \max_s (v^u(s) - v^l(s))$
  - participation rational for all agents
    - expected cost of each slot less than $v^l$
  - deviation unprofitable
    - expected utility for each slot must exceed $v^u(bestslot)$
Two More Mechanisms

• “Bulletin Board”
  – agents coordinate with each other by broadcasting their intended slot choice
  – agents get free slots according to $p^*$ iff their distribution is optimal; otherwise no slots are free
  – strict, optimal equilibrium

• “Discriminatory”
  – agents are assigned slots by the system
  – each agent gets the slot free according to $p^*$ iff he chose the assigned slot; otherwise he pays $m$
  – dominant strategy: unique, optimal equilibrium
Future Work

• Theoretical:
  – consider other cases where agents’ valuations not known
    • e.g., mechanism announces price of next slot, retroactive payment of
      agents not allowed
    • can we achieve a bound on optimality here?

• Practical:
  – apply one of our mechanisms in a real system
  – beginning to work with Stanford student housing system, which
    experiences focused loads on application deadlines
    • their database can accommodate only 40 simultaneous users
    • this year they were forced to extend the application deadline because of
      system unavailability

• For the whole story, please see our paper:
  available at http://robotics.stanford.edu/~kevinlb