A New Theory for Designing Socio-Computational Systems

Mihaela van der Schaar

Electrical Engineering, UCLA

Web: medianetlab.ee.ucla.edu

Multimedia signal processing, networking and communications

Multimedia Compression and Processing

MPEG, Philips

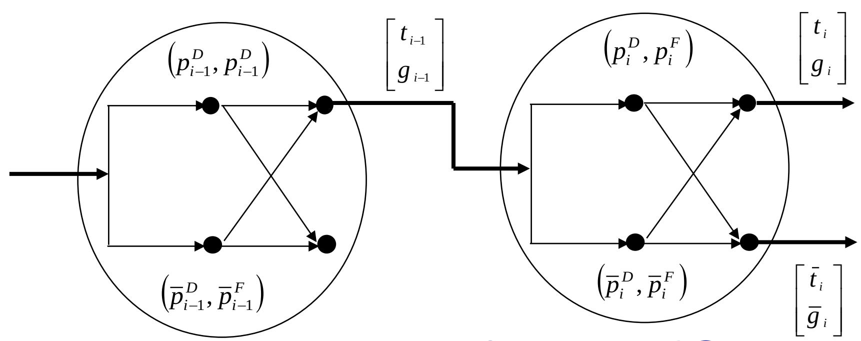
Rigorous methods for cross-layer design (dynamic environments)

NSF, Intel, HP, Microsoft

Delay-critical Networking and Online Learning

NSF, ONR, Intel, Cisco

Real-time Stream Mining



IBM, NSF

Goal: Designing Real-time Stream Mining Systems for a Smarter Planet

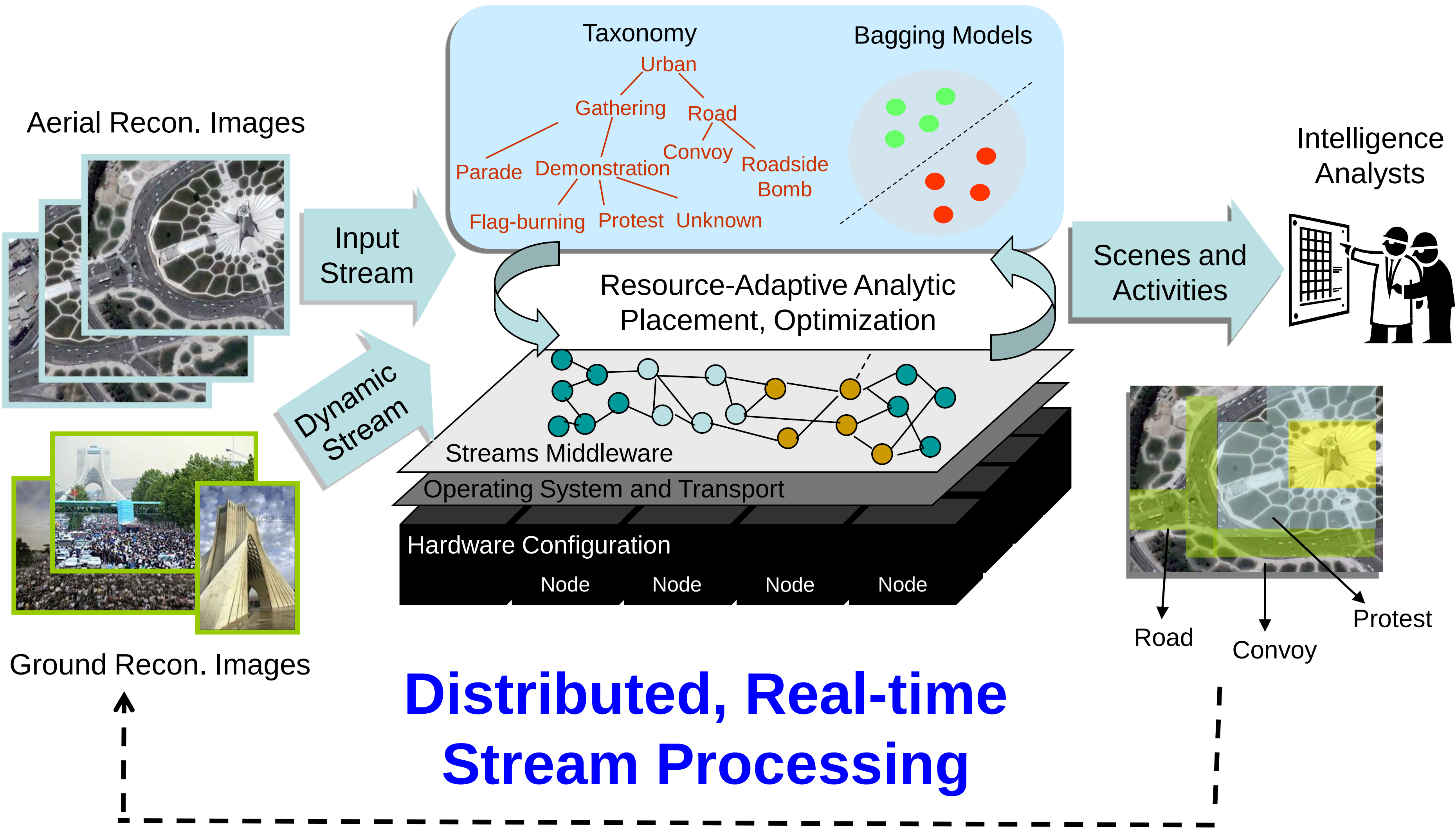
[NSF, IBM]

Applications:

1. Smarter cities
2. Online health monitoring
3. Social networks monitoring
4. Network security
5. Surveillance

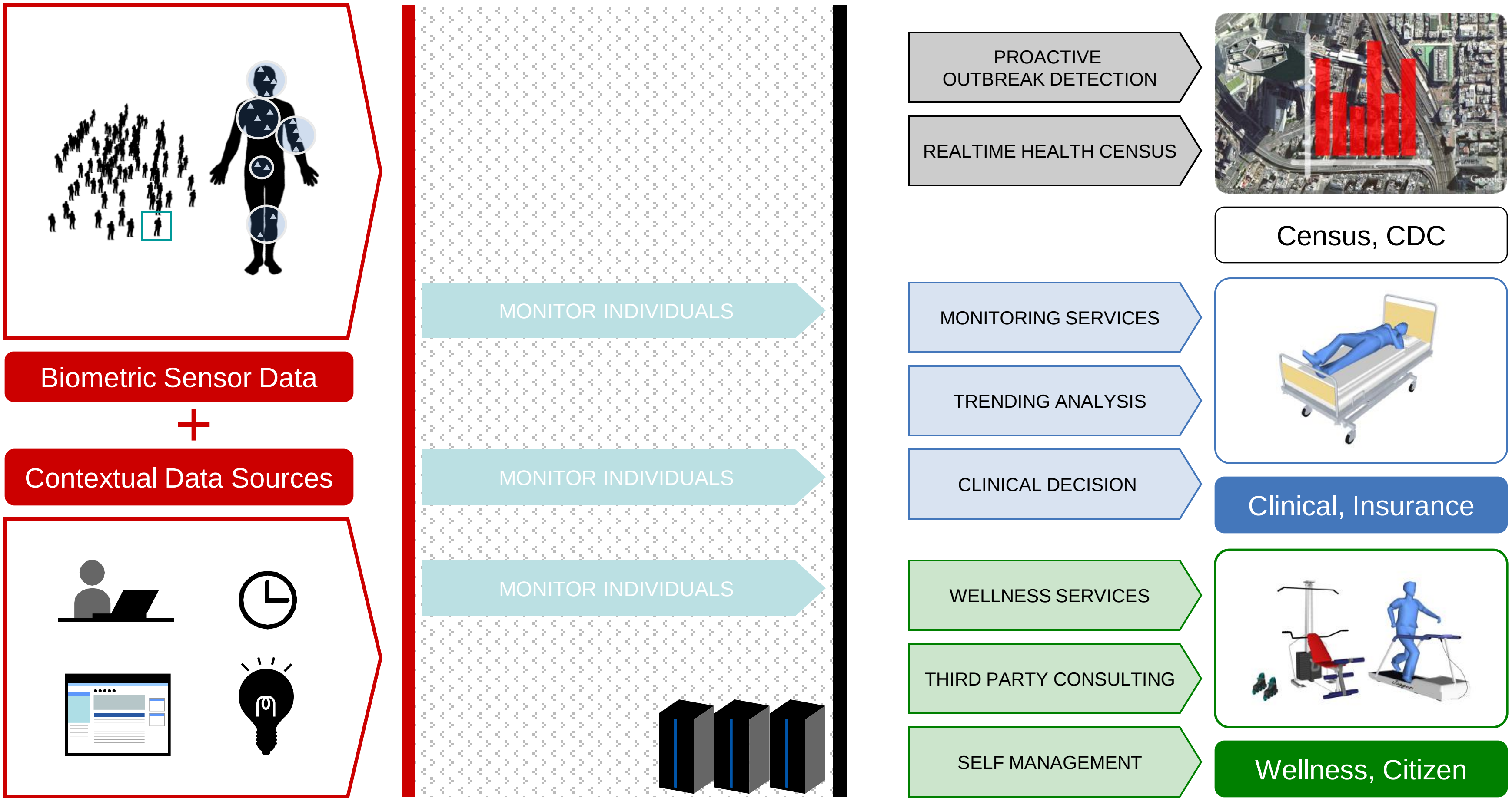
Stream mining - Semantic concept detection

Smarter cities



Distributed, Real-time Stream Processing

Stream mining - Online Healthcare Monitoring



Distributed, Real-time Stream Processing

Stream mining- Analysis for social networks

- Graph = nodes (= people, e.g. bloggers) + links (= interactions)
  - Each node includes a temporal sequence of ‘documents’ (blog posts, tweets, …)

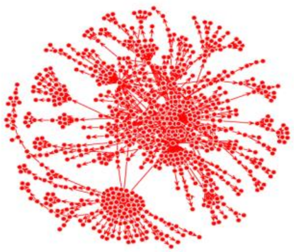
2a. Identify key influencers
Now: page rank, SNA measures, …

2b. Characterize viral potential
Now: use of follower statistics

INFLUENCE

1. Identify relevant content
Now: keyword search

RELEVANCE



3. Characterize objective vs subjective content
Now: lexical and pattern-based models

SUBJECTIVITY

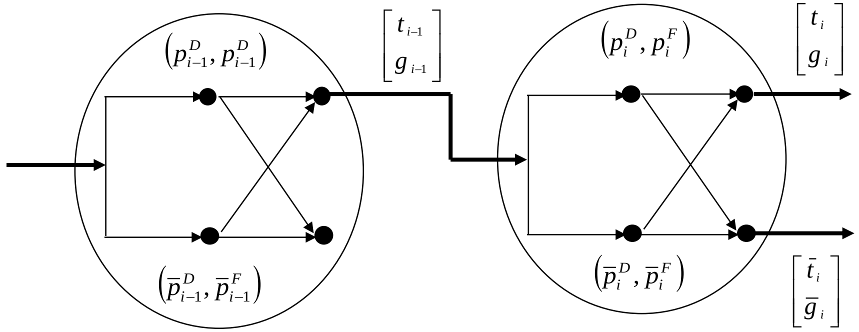
Distributed, Real-time Stream Processing

4a. Topic evolution & emergence
Now: word co-occurrence, clustering

4b. Classify new partially-observed documents
Now: unsupervised clustering

TOPIC IDENTIFICATION AND CLASSIFICATION

Stream mining - Challenges



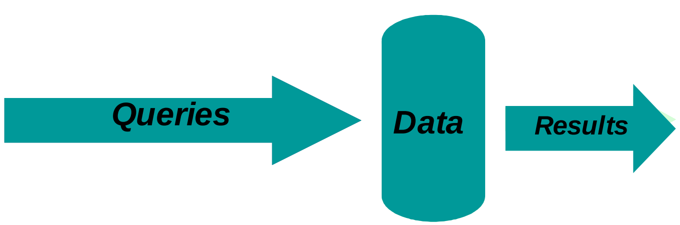
- *High Volume of data*: faster than a database can handle
- *Complex Analytics*: correlation from multiple sources and/or signals; video, audio or other non-relational data types
- *Delay-critical*: responses required in a specified time
- *Other system requirements*:
  - Scalable to the number of flows
  - Resource variability
  - Failure Tolerance
    - Data cannot be stored and reprocessed
    - Requirements on graceful degradation under failure
  - Distributed computation by various entities

Stream Computing: New Paradigm

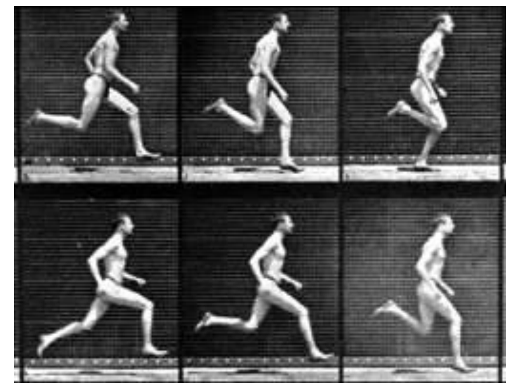
Traditional Computing

Historical fact finding with data-at-rest

- Batch paradigm, pull model
- Query-driven: submits queries to static data
- Relies on Databases, Data Warehouses



Stream Computing



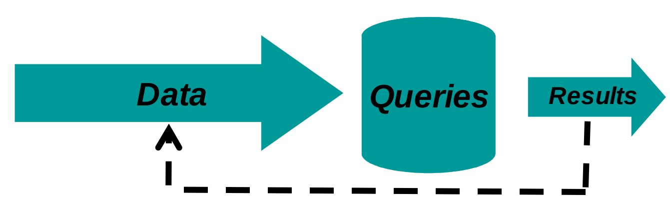
Real time analysis of data-in-motion

Streaming data

- Stream of structured or unstructured data-in-motion

Stream Computing

- Analytic operations on streaming data in real-time



Multi-disciplinary research effort

- Signal Processing and Machine Learning
  - Real-time adaptive analytics
    - Stream data aggregation, filtering, compression, processing
  - Incremental learning
  - Cross-layer design
    - System and Analytics
- Distributed system designs for autonomous and self-interested agents
- Social computing

A New Theory for Designing Socio-Computational Systems

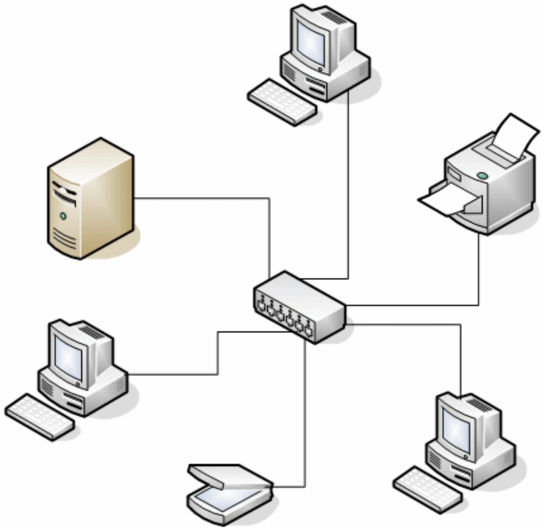
Mihaela van der Schaar
Electrical Engineering, UCLA
Web: medianetlab.ee.ucla.edu

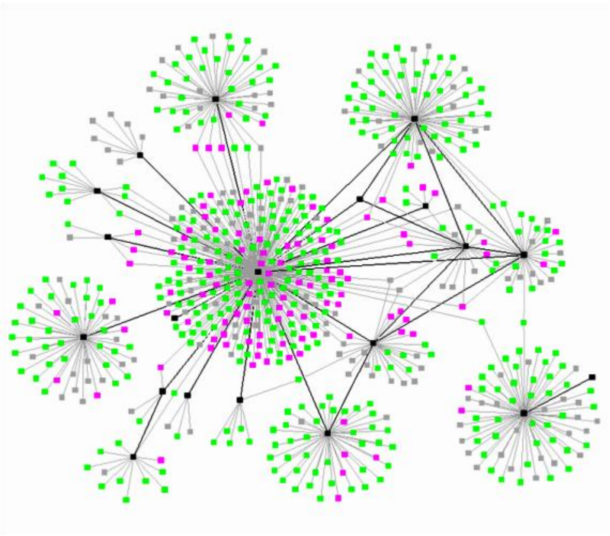
Acknowledgements

- Yu Zhang
- Jaeok Park
- William Zame

Emergence of socio-computation systems

- Socio-computational systems allow individuals and organizations to get connected and build relationships.
- Rapid expansion of social cloud computing, social networks, online labor markets, P2P networks, multi-user mobile communication etc.





Socio-computational systems = collection of self-interested, learning agents (people, machines, software …)

Designing socio-computational systems

- **Goals**
  - ***Design*** networks, systems and protocols that maximize the designer's utility by inducing compliance by agents
  - Designer's utility = social welfare/fairness/revenue maximization etc.

- **Who is the designer?**
- **Challenges**
  - Self-interested, rational, heterogeneous users
  - Large-scale
  - Ongoing interactions
  - Robustness

Where are we coming from and where are we going?

Classical System Design

- Nodes: Cooperative
- System designer has a high degree of control: prescribes decision rules for nodes
- Systems assume compliance
- Social and individual goals coincide, e.g. utility maximization
- Truthful information revelation assumed
- Mostly single-agent learning, prescriptive

Next-generation System Design

- Agents: Self-interested, strategic
- System designer can control only a playground on which agents interact, but the agents choose how to play
- System compliance not guaranteed - Strategy-proof protocols needed
- Social and individual goals in conflict, e.g. system collapse
- Agents may lie/hide information
- Multi-agent learning

Designing socio-computational systems

New Theoretical Foundations

Strategic design

Application Domains

- **Online trading markets**
- **P2P networks**
- **Multimedia networks & systems**
- **Cyber-security**
- **Social cloud computing**
- **Wireless, cognitive, mobile, mission-critical networks**
- **Network Economics**
- **Energy policy/EVs**

How is this different than Game Theory?

Game Theory

- **Focus: Analysis, Behavioral understanding**
- **Example: Repeated games**
  - Folk Theorems
  - Monitoring (given)
  - Reputation – types
  - Social norms
  - Review strategies
  - Cheap talk

Strategic Design

- **Focus: Design**
- **Example: Repeated interactions**
  - Optimal design given constraints (signaling, information, memory, physical limitations etc. etc.)
  - Optimality criteria are decided by the designer
  - Monitoring/Information feedback – design
  - Group protocols
  - Group reputation
  - Personal observations
  - Robustness
  - Tokens
  - Selection of partners

Group protocols

- Group protocols - rules for appropriate and inappropriate behaviors
  - Compliance
  - Rewards (present and future)
  - Punishments (present and future)
- We consider a group protocols using reputation.
  - Each peer is tagged a reputation label.

Formal Representation of a Group Protocol

- A group protocol is represented by $\kappa = (\Theta, \theta_0, \tau, \sigma)$
  - Θ : set of reputation labels
  - $\theta_0 \in \Theta$: initial reputation
  - $\tau : \Theta \times \Theta \times \mathcal{A} \rightarrow \Theta$: reputation update rule
    - $\tau(\theta, \tilde{\theta}, a_R)$ is the new reputation for a server with current reputation θ when it is matched with a client with reputation $\tilde{\theta}$ and its action is reported as a_R.
  - $\sigma : \Theta \times \Theta \rightarrow \mathcal{A}$: prescribed strategy
    - $\sigma(\theta, \tilde{\theta})$ is the approved action for a server with reputation θ that is matched with a client with reputation $\tilde{\theta}$.

reputation scheme

What do agents know? What choices do they have?

Design Problem

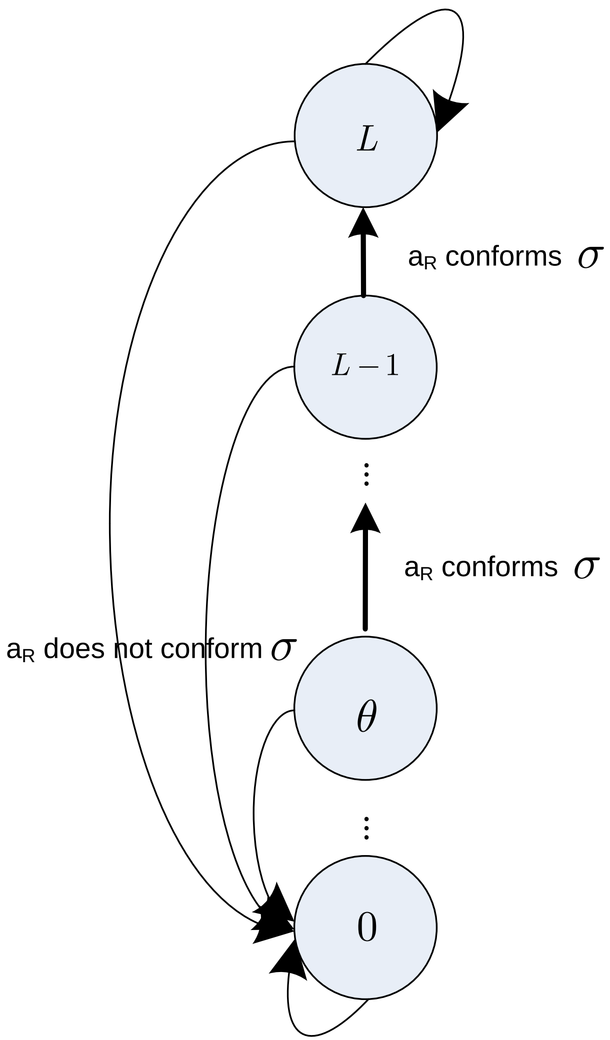
- The design problem can be expressed as

$$\begin{aligned} &\underset{\kappa}{\text{maximize}} \quad U_{\kappa} \; ? \\ &\text{subject to} \;\; \text{Sustainability} \end{aligned}$$

- An optimal group protocol is a group protocol that maximizes social welfare *among sustainable group protocols* (i.e. selfish agents want to follow the prescribed strategies).

Design Choice

- The design choice in the protocol design problem is a group protocol $(\Theta, \theta_0, \tau, \sigma)$.
- Starting point: simple designs
  - Impose restrictions on (Θ, θ_0, τ)
  - Θ is finite, i.e., $\Theta = \{0, 1, \ldots, L\}$ for some integer L.
  - Initial reputation is $\theta_0 = 0$
  - Punishments/Rewards – fixed
- Simple group protocol designs: (L, σ)
- Even the design of prescribed strategies can be restricted: e.g. focus on "threshold" strategies



Design of an exemplary networked community: Crowdsourcing platforms

- Numerous crowdsourcing platforms, such as Yelp, Yahoo! Answers and Amazon Mechanical Turk, can be viewed as socio-computational systems where small tasks (typically on the order of minutes or seconds) are performed in exchange for rewards awarded to the users who performed them.
- A task is described and posted by a requester together with an associated reward.
- Workers submit solutions to tasks, and the requester selects a subset of submissions (usually the first one that solves the task) and the selected workers are rewarded.

Setup

- There are more requesters than workers.
- The price for each task is q (flat-rate pricing) <- Initially
- Workers select the task to solve. Each worker selects one task she can solve with equal probability.
- Time is divided into periods, with each period being the typical length of time to solve a task.
- Each worker can only devote her effort to a single task in each period.
- Other system parameters:
- $\delta \in [0,1)$: time discount factor
- $\varepsilon \in [0,0.5]$: report error probability

Game Played by a Pair of Matched Users

- We assume that the requester always pays the same amount.
  - The worker receives μq .
  - The website charges $1-\mu \;\; q$ as the transaction fee.
- Actions:
  - Requester: no action to choose
  - Worker: $a \in \mathcal{A} = \;\; S, NS$
    - S: High level of effort
    - NS: Low level of effort
- Payoffs:
  - If the worker exerts a high level of effort, she incurs a cost *c* and the requester receives a benefit *b*.

	Worker	
	S	NS
Requester	$b-q, \quad \mu q - c$	$-q, \quad \mu q$

Incentives needed!

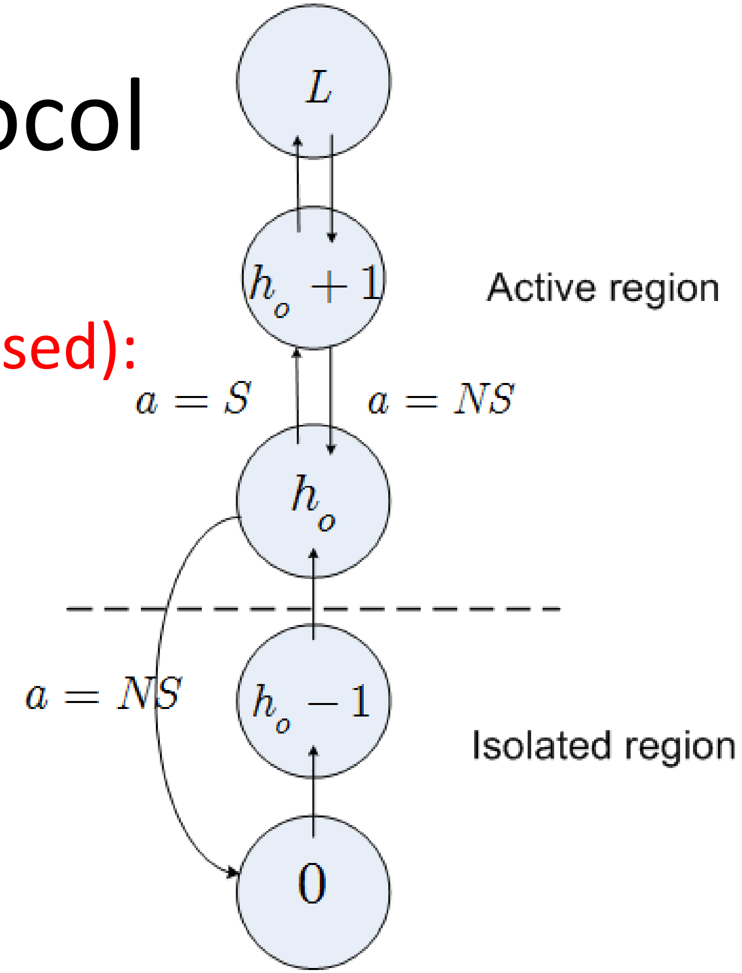
A “Simple” Group Protocol

- Prescribed protocol (threshold-based):

$$\sigma\;\theta = \begin{cases} S & if\;\; \theta \geq h_o \\ NS & otherwise \end{cases}$$

- Reputation scheme:

$$\tau\;\theta, a = \begin{cases} \min\;\; L, \theta + 1 & if\;\; a = S\;\; and\;\; \theta \geq h_o \\ \theta - 1 & if\;\; a = NS\;\; and\;\; \theta \geq h_o + 1 \\ 0 & if\;\; a = NS\;\; and\;\; \theta = h_o \\ \theta + 1 & if\;\; \theta < h_o \end{cases}$$



Users' Utilities and Social Welfare

- The expected period payoff of the worker complying with the group protocol:

$$v_{\kappa} \quad \theta \quad = \mu q - c, \ \text{if } \theta \geq h_{o}$$

$$v_{\kappa} \quad \theta \quad = 0, \ \text{if } \theta < h_{o}$$

Lemma: There exists a unique stationary distribution of reputations η_{κ}

- **Social welfare:** average period payoff of all workers and requesters

$$U_{\kappa} = \sum_{\theta < h_{o}} \eta_{\kappa} \quad \theta \quad v_{\kappa} \quad \theta \quad + \sum_{\theta \geq h_{o}} \eta_{\kappa} \quad \theta \quad\quad v_{\kappa} \quad \theta \quad + b - q$$

Design of the Group Protocol

The platform designer's problem:

$$\max_{L,h_o,\mu} \quad U_\kappa$$

subject to

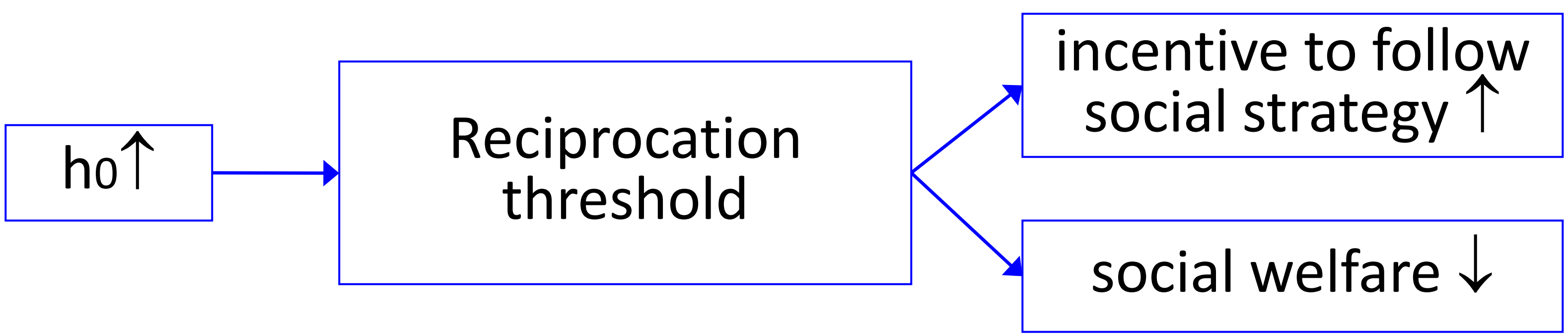
$$c \leq \delta \ 1 - 2\varepsilon \left[v_\kappa^\infty \ \min \ L, \theta + 1 \ - v_\kappa^\infty \ \theta - 1 \right], \text{ if } \theta \geq h_o + 1,$$

$$c \leq \delta \ 1 - 2\varepsilon \left[v_\kappa^\infty \ \min \ L, \theta + 1 \ - v_\kappa^\infty \ 0 \right], \text{ if } \theta = h_o.$$

Sustainability conditions

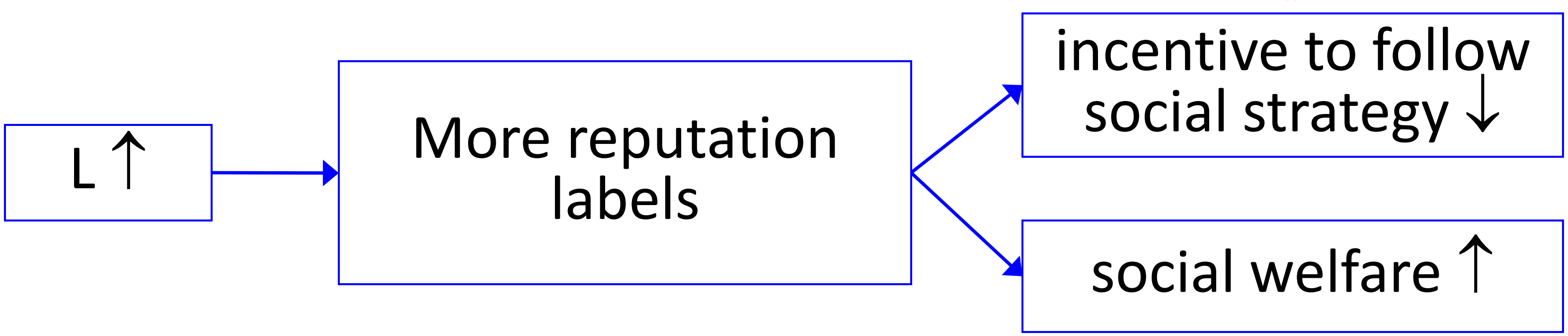
Resulting optimal design μ^*, L^*, h_0^*

- Given a group protocol κ, $\mu^* = 1$ provides optimal solution.
  - The website designer only has to choose the optimal group protocol when setting $\mu^* = 1$.
- Impact on social welfare:
  - U_κ monotonically increases with L and monotonically decreases with h_o.
- Impact on incentives:
  - Given q, c, δ, and ε, a group protocol $\kappa = \ \sigma, \tau$ can be sustained as an equilibrium if and only if
    - Its protocol threshold h_o is larger than a constant $\bar{h}\ \ q, c, \delta, \varepsilon$;
    - The highest reputation L is smaller than a constant $\bar{L}\ \ q, c, \delta, \varepsilon$.

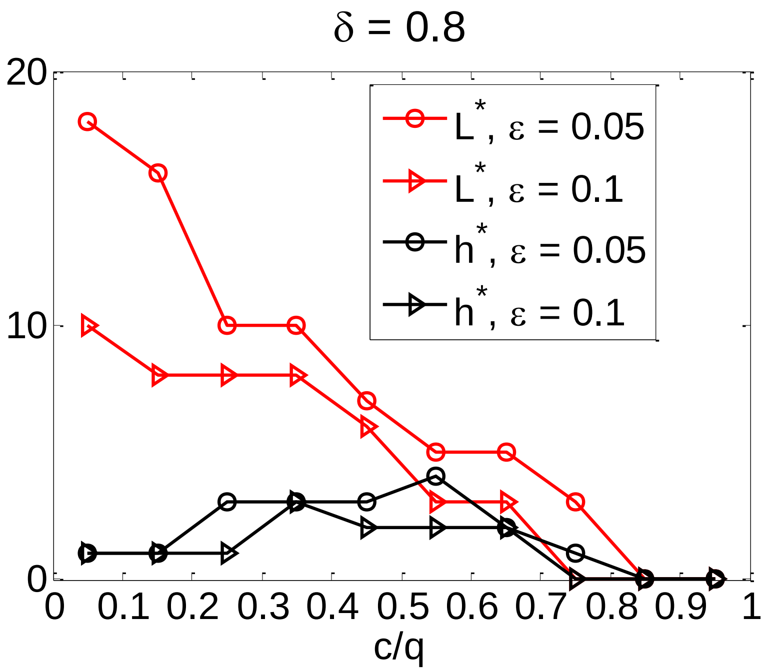


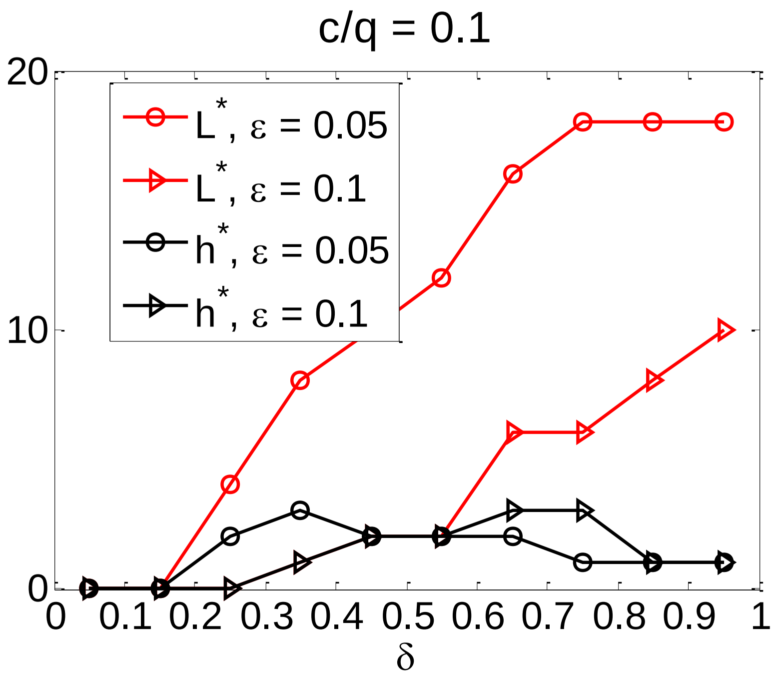
Resulting optimal design μ^*, L^*, h_0^*

- Given a group protocol κ, $\mu^* = 1$ provides optimal solution.
  - The website designer only has to choose the optimal group protocol when setting $\mu^* = 1$.
- Impact on social welfare:
  - U_κ monotonically increases with L and monotonically decreases with h_o.
- Impact on incentives:
  - Given q, c, δ, and ε, a group protocol $\kappa = \sigma, \tau$ can be sustained as an equilibrium if and only if
    - Its protocol threshold h_o is larger than a constant $\bar{h}\ q, c, \delta, \varepsilon$;
    - The highest reputation L is smaller than a constant $\bar{L}\ q, c, \delta, \varepsilon$.



Resulting optimal design μ^*, L^*, h_0^*





Platform wants to its maximize revenue

So far, the focus was on maximizing the social welfare.

Now:

The design problem changes significantly when the platform aims to maximize its own revenue.

New platform designer's problem:

$$\max_{L,h_o,\mu} \quad R_\kappa = \sum_{\theta \geq h_o} \eta_\kappa \;\; \theta \;\; 1-\mu \;\; q$$

subject to

$$c \leq \delta \;\; 1-2\varepsilon \left[v_\kappa^\infty \;\; \min \;\; L, \theta+1 \;\;\; - v_\kappa^\infty \;\; \theta - 1 \right], \text{ if } \theta \geq h_o + 1,$$

$$c \leq \delta \;\; 1-2\varepsilon \left[v_\kappa^\infty \;\; \min \;\; L, \theta+1 \;\;\; - v_\kappa^\infty \;\; 0 \right], \text{ if } \theta = h_o.$$

Different design emerges!

The design changes!

- Social welfare maximization
  - A large μ increases both the social welfare as well as the incentive of workers --- It should always be set to be 1.
- Revenue maximization
  - A large μ increases the incentive of workers but reduces the revenue of the website --- The tradeoff has to be considered.

- The optimal design $\mu^{\#}$ monotonically increases with the cost-to-price ratio c / q and monotonically decreases with the discount factor δ .
- $\lim\limits_{c/q \to 0} \mu^{\#} = 0$ and $\lim\limits_{\delta \to 1} \mu^{\#} = c / q$.

$$v_{\kappa} \quad \theta \ = \mu q - c, \ \text{if } \theta \geq h_{o}$$

- The optimal revenue $R^{\#}$ monotonically decreases with the cost-to-price ratio c / q and monotonically increases with the discount factor δ .

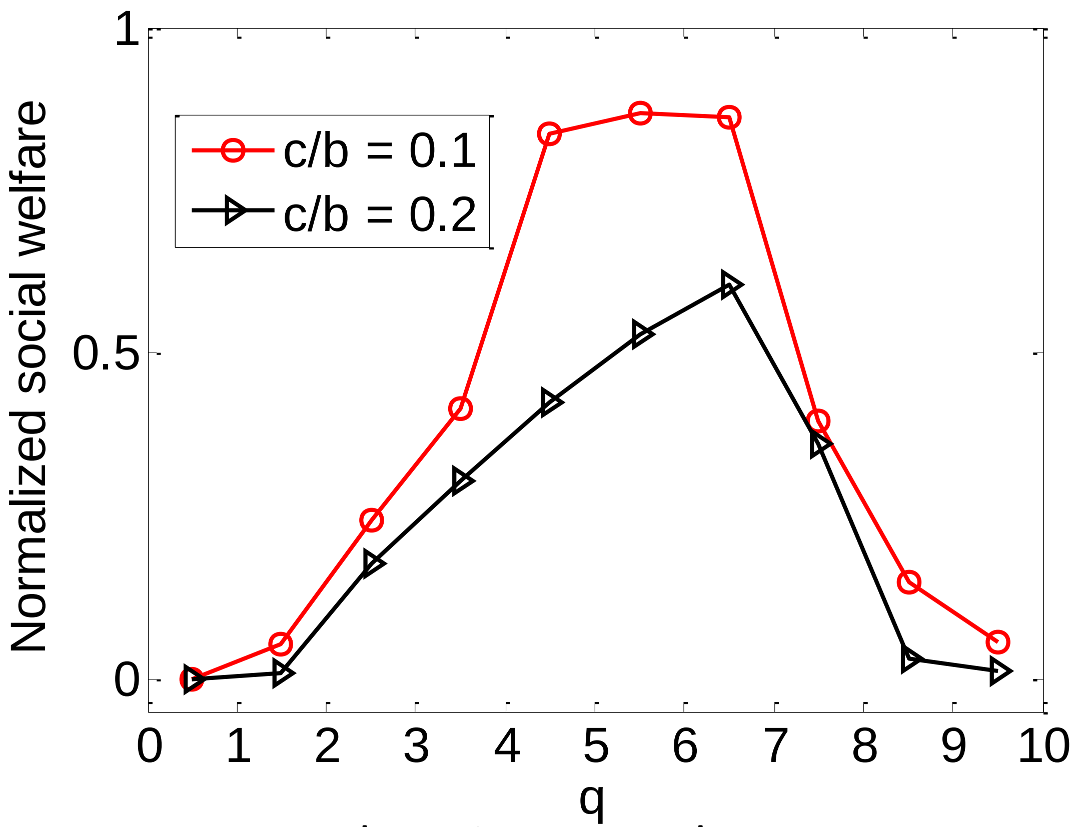
Different design if requesters are strategic

Next, we assume that requesters are also strategic in determining whether to make or not to make payments.

		Worker	
		S	NS
Requester	Pay	$b-\mu q\,,\; q-c$	$-q\,,\; q$
	No pay	$b\,,\; -c$	$0\,,\; 0$

In this case, the selection on the service price q will influence requesters' incentive and thus the social welfare, i.e. it becomes a *design parameter*.

Social Welfare vs. Service Price

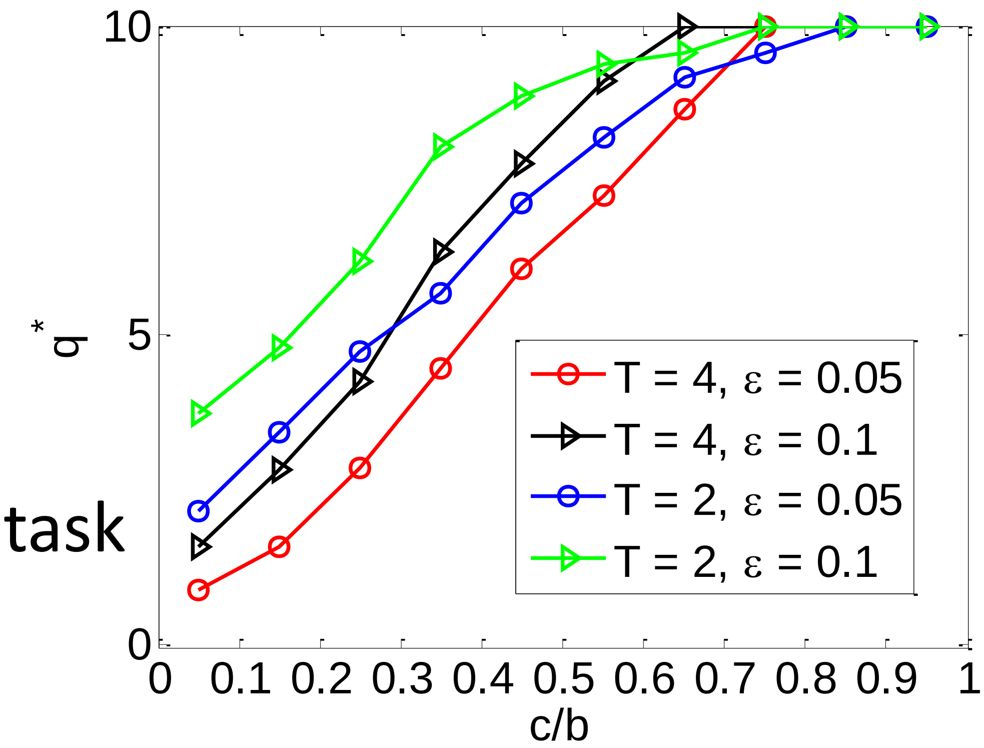


The price per task

- When q is small, workers' incentive increases against q and the social welfare increases.
- When q becomes large, requesters' incentive decreases against q and the social welfare decreases.

Optimal Service Price

Optimal price per task



- **T=population(requester)/population(worker)**
- A larger ε results in lower incentives for workers, which in turn requires a higher price to encourage their contributions.
- A larger T implies a lower frequency for requesters to interact with workers. Therefore, they will put less weight on their future utilities, and a smaller price is needed to encourage requesters' participations.

Findings

- Other "designed" communities:
  - P2P networks
  - Content/knowledge production
  - Other labor markets
- Other interesting results:
  - Design in the presence of altruistic users
  - Group protocol for "friends" vs. "passers-by"
  - Group protocols using tokens instead of reputations
- Engineer communities for which we can prove that "simple" designs are optimal
- "Robust" designs
- **Golden rule: Design matters!**

Reputations		Tokens
Central memory	Memory	No central memory (tokens as memory)
Reputation ↑	Rewards	Treasury ↑
Reputation ↓	Punishments	
High	Informational requirements	Low
Does not limit effectiveness of design	Impatience	Limits effectiveness of design (nobody chooses to build a large treasury)
Initial reputation	Whitewashing	Initial endowment

Part II:

Design of Dynamic Personal Reciprocation Policies

- Hyunggon Park and Mihaela van der Schaar, “A Framework for Foresighted Resource Reciprocation in P2P Networks,” *IEEE Trans. Multimedia*, vol. 11, no. 1, pp. 101-116, Jan. 2009.
- Hyunggon Park and Mihaela van der Schaar, “Evolution of Resource Reciprocation Strategies in P2P Networks,” *IEEE Trans. Signal Process.*, vol. 58, no. 3, pp. 1205-1218, Mar. 2010.
- Rafit Izhak-Ratzin, Hyunggon Park and Mihaela van der Schaar, “Reinforcement Learning in BitTorrent Systems,” *Infocom 2011.*

Part II: Dynamic P2P systems

- As before
  - Users interact repeatedly
  - Users are heterogeneous
  - Information is decentralized

- New
  - Choose ***partners*** and ***level*** of cooperation
  - Environment ***changing***

→ **No previous solutions for rigorously designing and evaluating protocols for P2P systems in dynamic environments**

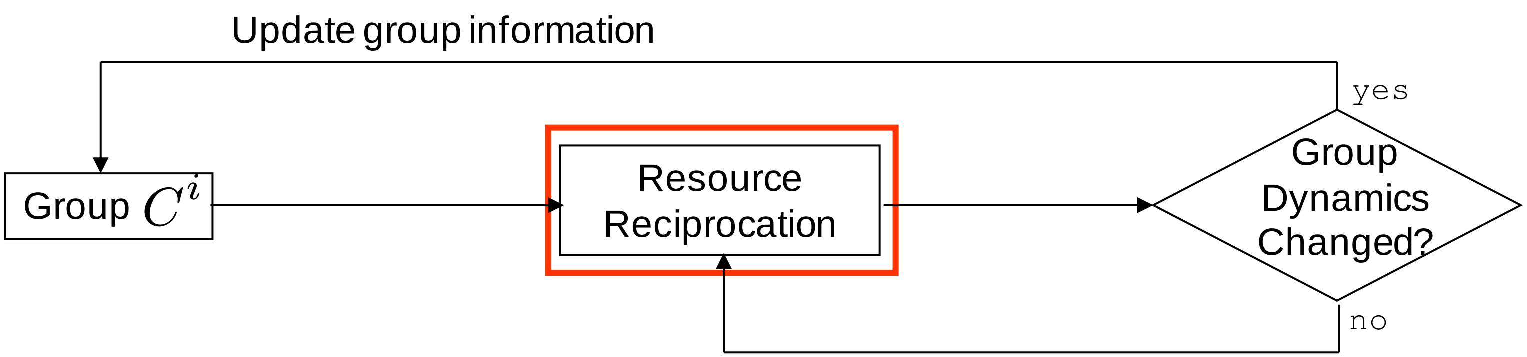
Our approach – central issues

a) What reciprocation policy (protocol) to adopt while environment is known and stationary?

b) How to change the policy when environment changes?

A) Markov strategies – use Markov Decision Processes (MDPs) to determine policies

B) Online learning –reinforcement learning or model-based

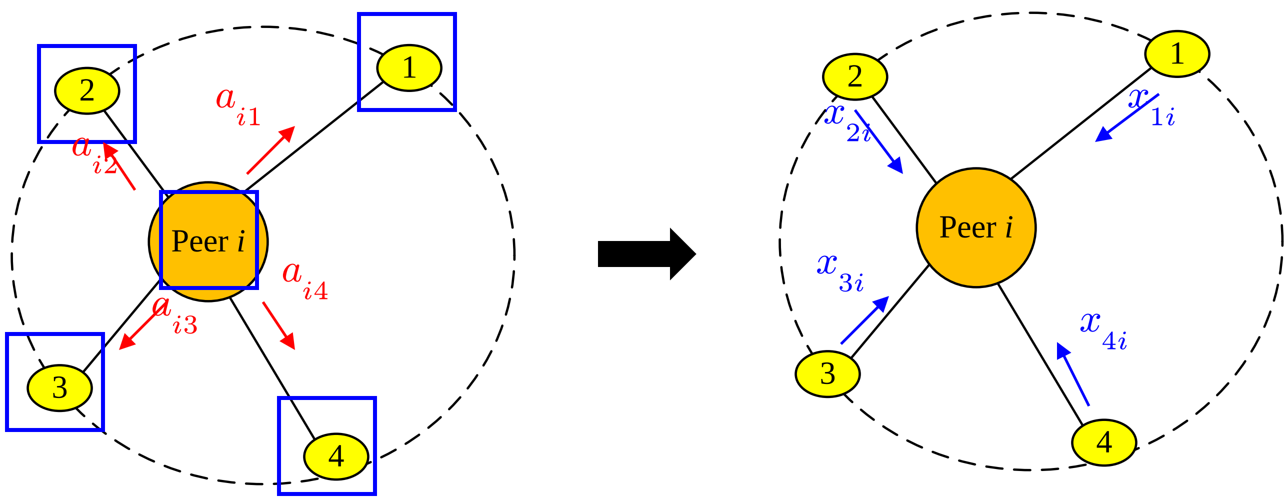


Resource Reciprocation

- A finite set of agents (peers)
- Actions: upload bandwidth allocations
- Policy: actions selected today are based on yesterday's reciprocation levels = states
- Utility: download rates, video quality, etc.
- Foresighted peers worry about long-term utility

State descriptions =>

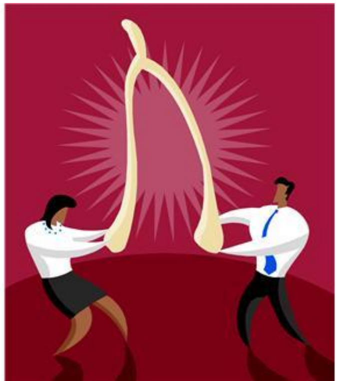
Peers' intelligence



! Policy determines optimal *level* of cooperation, unlike "all or nothing" solution in BitTorrent (Tit-For-Tat)

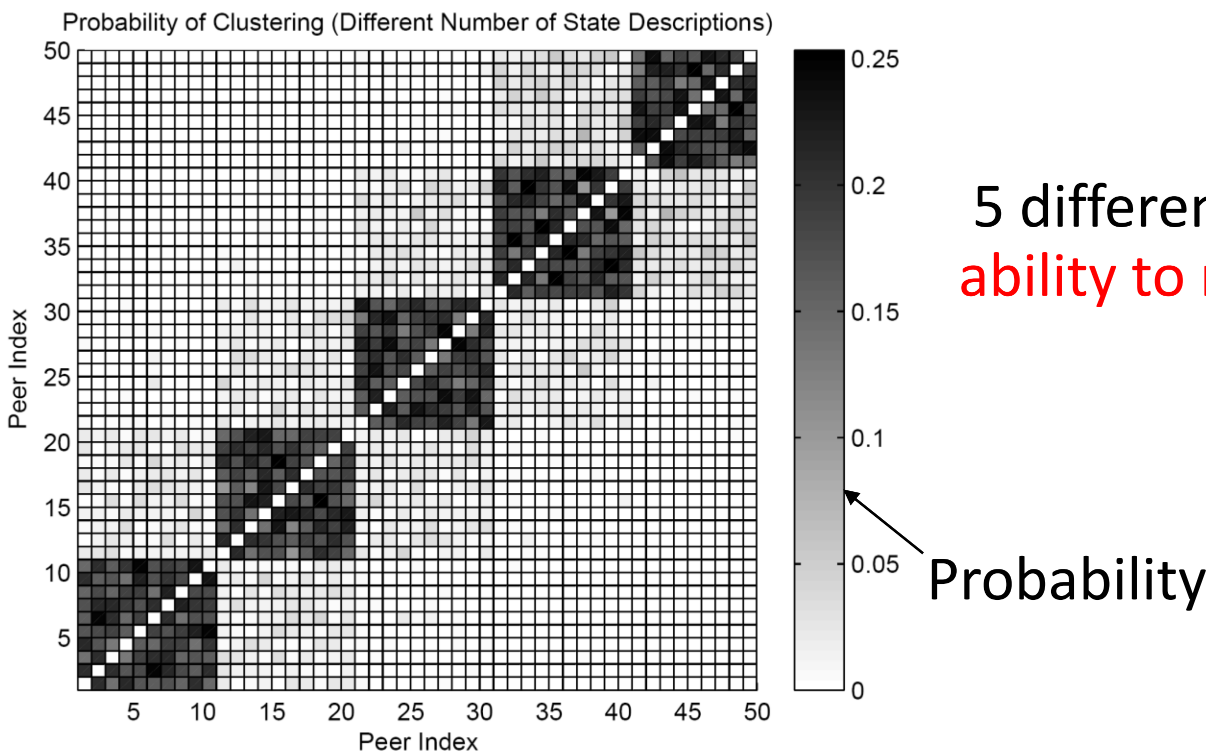
Discrimination among peers - How?

- We prove assortative matching
  - Richer peers (=peers with higher bandwidth) match with richer peers
    - Generosity prompts generosity
  - Smarter peers (= peers with more refined states) match with smarter peers
    - Careful monitoring prompts careful monitoring
    - Better to cooperate with smarter peers than to steal from stupid peers ☺



Clustering for Heterogeneous Peers

Different state refinement ability, same available bandwidth



5 different classes: different ability to refine states

→ Peers prefer to form a group with peers having similar ability to refine states

Implementation and real-world experiments in Planetlab (Infocom 2011)

Part III: Community Formation

Information production, sharing and consumption and link formation in networked communities

- Jaeok Park and Mihaela van der Schaar, “A Game Theoretic Analysis of Incentives in Content Production and Sharing over Peer-to-Peer Networks,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 4, no. 4, pp. 704-717, August 2010.
- Jaeok Park and Mihaela van der Schaar, “Content Pricing in Peer-to-Peer Networks,” *NetEcon '10.*
- Jaeok Park and Mihaela van der Schaar, “Pricing and Incentives in Peer-to-Peer Networks,” *INFOCOM 2010.*

Current EE/CS/Econ Literature		**Our research**
Fixed	Who produces?	Choice
Fixed	What/how much is produced?	Choice
Fixed	What/how much is shared?	Choice
Fixed	Who connects to whom?	Choice

Challenges

Research agenda

New Theoretical Foundations

Strategic design

Application Domains

- Online trading markets
- P2P networks
- Multimedia networks & systems
- Cyber-security
- Social cloud computing
- Wireless, cognitive, mobile, mission-critical networks
- Network Economics
- Energy policy/EVs

Our research: medianetlab.ee.ucla.edu