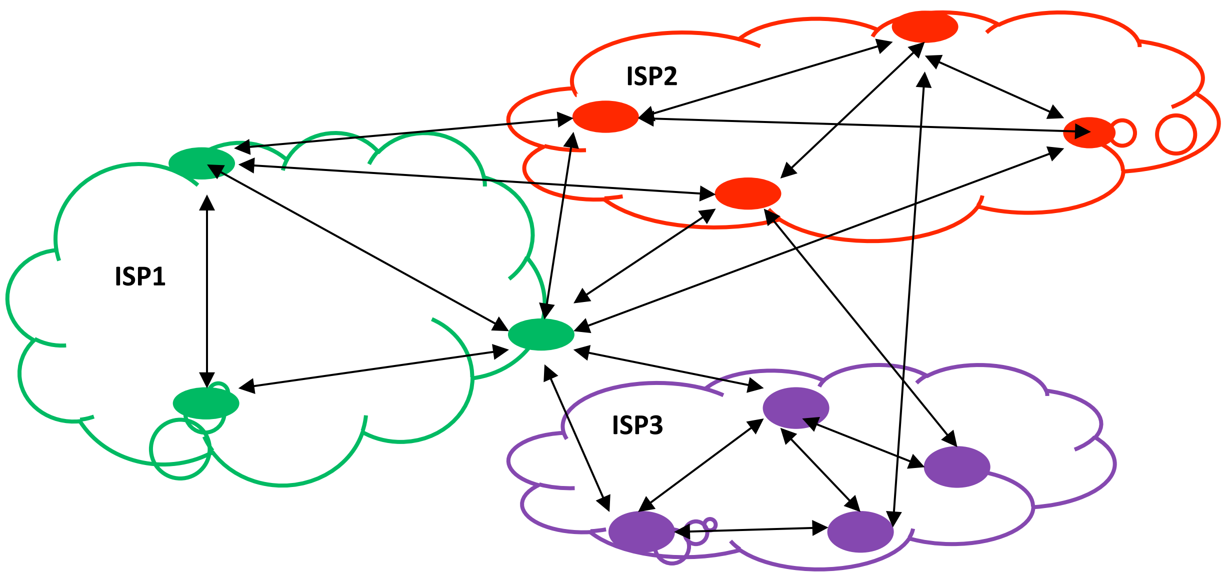
Designing ISP-Friendly P2P Using Game-based Control

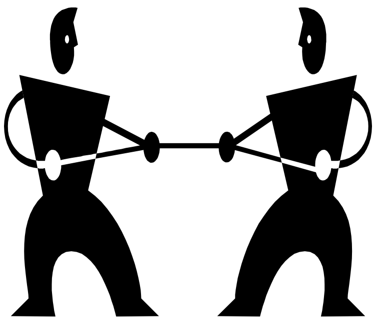
Srinivas Shakkottai
Texas A&M University

Problems with existing P2P

- Oblivious of ISP domains
- Can result in huge data flow across ISP boundaries
- Hence increased cost for an ISP



Problem Overview



- We need a P2P system that trades off transit price and delay
  - Price is reduced by localizing traffic within an ISP domain
  - Delay can be reduced by choosing the best peer, irrespective of the ISPs

Key Question: How to achieve the optimal point?

Related Work

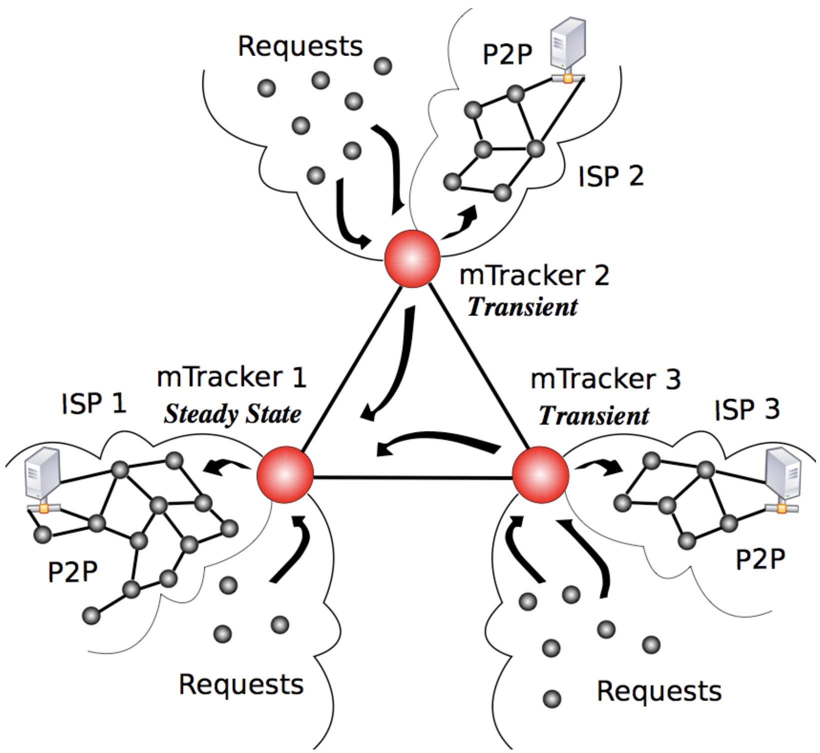
- V. Aggarwal, A. Feldmann, and C. Scheideler, Can ISPs and P2P users cooperate for improved performance?

  ACM Computer Communication Review, 37(3), July 2007.

- H. Xie, Y. R. Yang, A. Krishnamurthy, Y. Liu, and A. Silberschatz, P4P: Portal for P2P applications.

  In Proc. ACM SIGCOMM, Aug. 2008.

MultiTrack for BitTorrent-like P2P

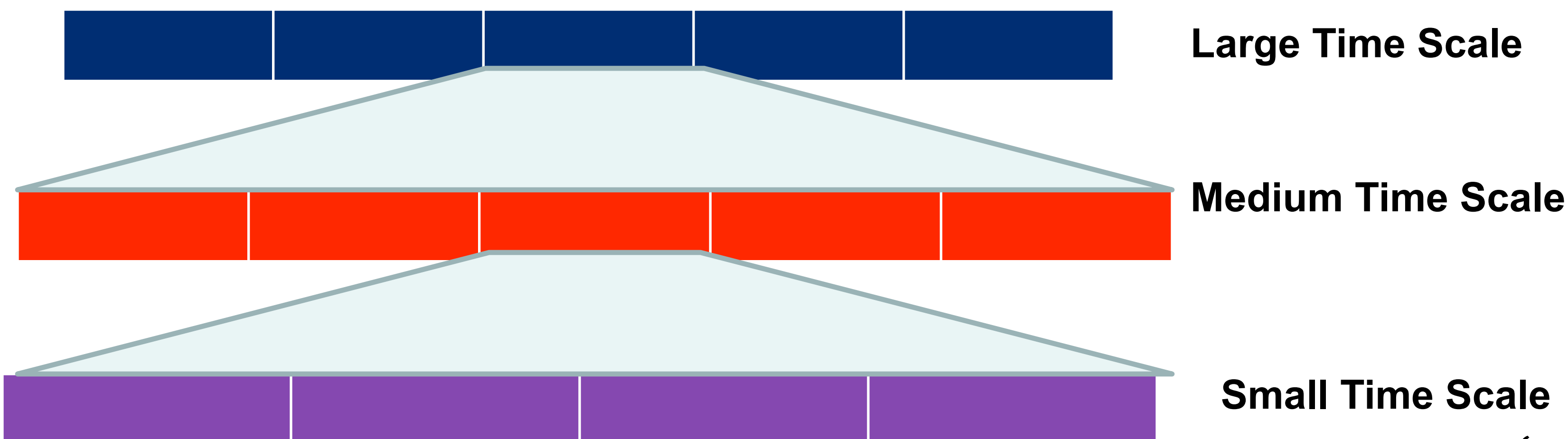


- **<u>Steady State</u>** : Load is less than the available capacity
- **<u>Transient State</u>** : Load is more than the available capacity

- Must split traffic taking into account both delay and cost.

Assumptions

- Capacity at mTracker i (or the peer swarm) is assumed to be C^i *users/time*
- New requests arrive at mTracker j in a Poisson process with parameter x_j *users/time*
- Delay is convex increasing in load.



Population Game

- A population game G, has Q non-atomic populations and for each population *j*:
  - A mass x_j ,
  - A strategy set $\mathcal{S}_j = \{1, ..., S_j\}$
  - A marginal payoff for each strategy $F_j^i(\mathbf{X}), i \in \mathcal{S}_j$ where $\mathbf{X}$ is the state of the system

- A **state** $\mathbf{X}$ (or a strategy distribution) is the way the population is partitioned into the different strategies available, $\mathbf{X} = \{\vec{x}_1, \vec{x}_2, ..., \vec{x}_Q\}$

$$\vec{x}_j = \{x_j^1, x_j^2, ...x_j^{S_j}\}$$ → **Sum is exactly x_j**

Dynamics

- Every player follows selfish dynamics, maximizing their own payoff.
- User strategies evolve with time as they adapt to the state.

Replicator Dynamics: *Rich become richer and poor become poorer*

$$\dot{x}_j^s = x_j^s \left(\underbrace{F_j^s(\mathbf{X})}_{\text{Payoff per unit}} - \underbrace{\frac{1}{x_j} \sum_{i=1}^{S_j} x_j^i F_j^i(\mathbf{X})}_{\text{Average payoff per unit}} \right)$$

Payoff per unit **Average payoff per unit**

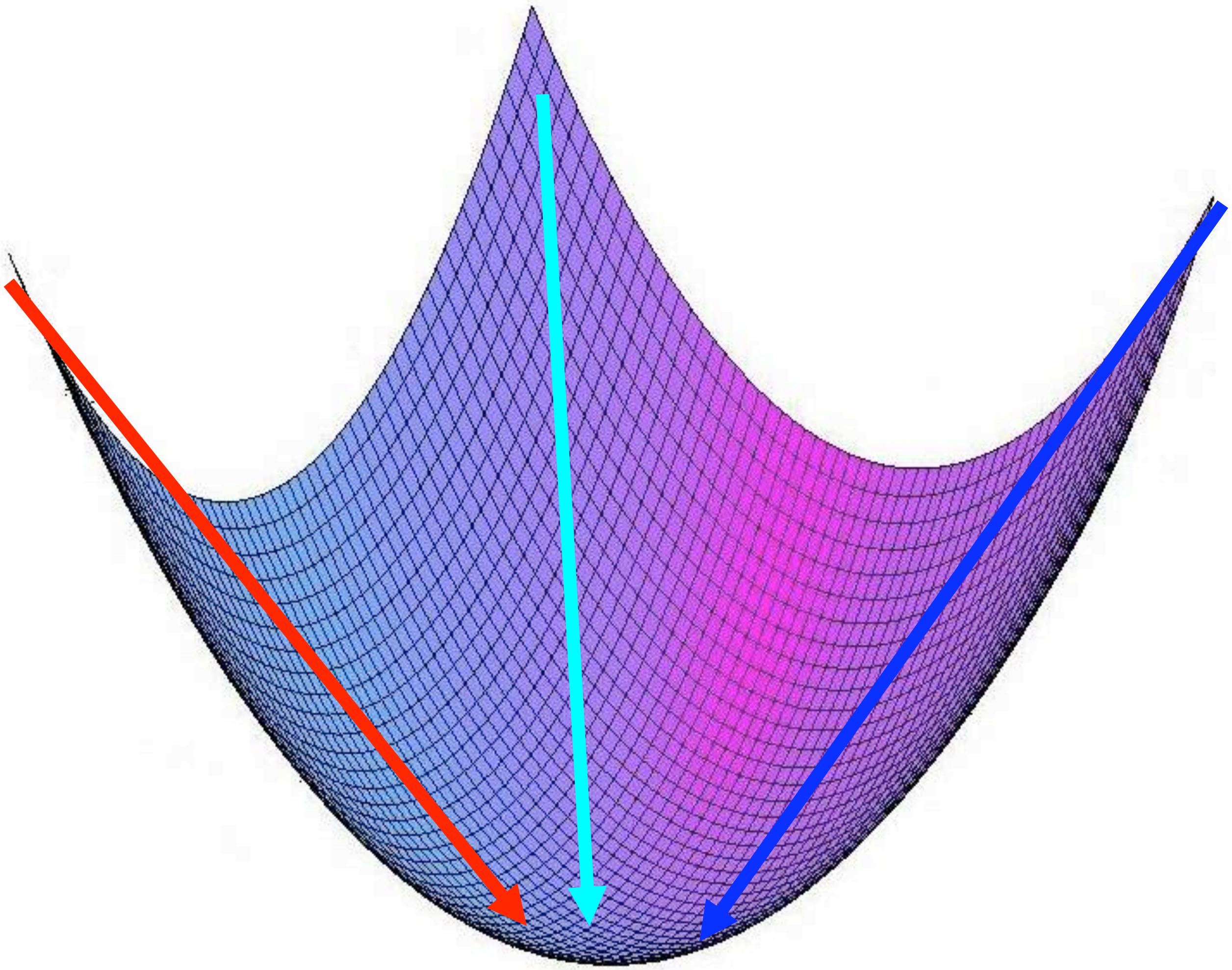
Marginal Payoff/Cost

- F_j^i *(X)* represents per unit payoff for mTracker *j* in forwarding request to strategy *i in state* **X** :
  - *Delay at mTracker i*
  - *Transit cost from mTracker j to mTracker i*
  - *Congestion cost at mTracker i*

$$F_j^i(\mathbf{X}) = \frac{1}{C^i - \sum_{l=1}^{Q} x_l^i} + p_j^i + \frac{\sum_{l=1}^{Q} x_l^i}{(C^i - \sum_{l=1}^{Q} x_l^i)^2}$$

Marginal delay | **transit cost** | **Congestion**

Lyapunov Function



Total System cost

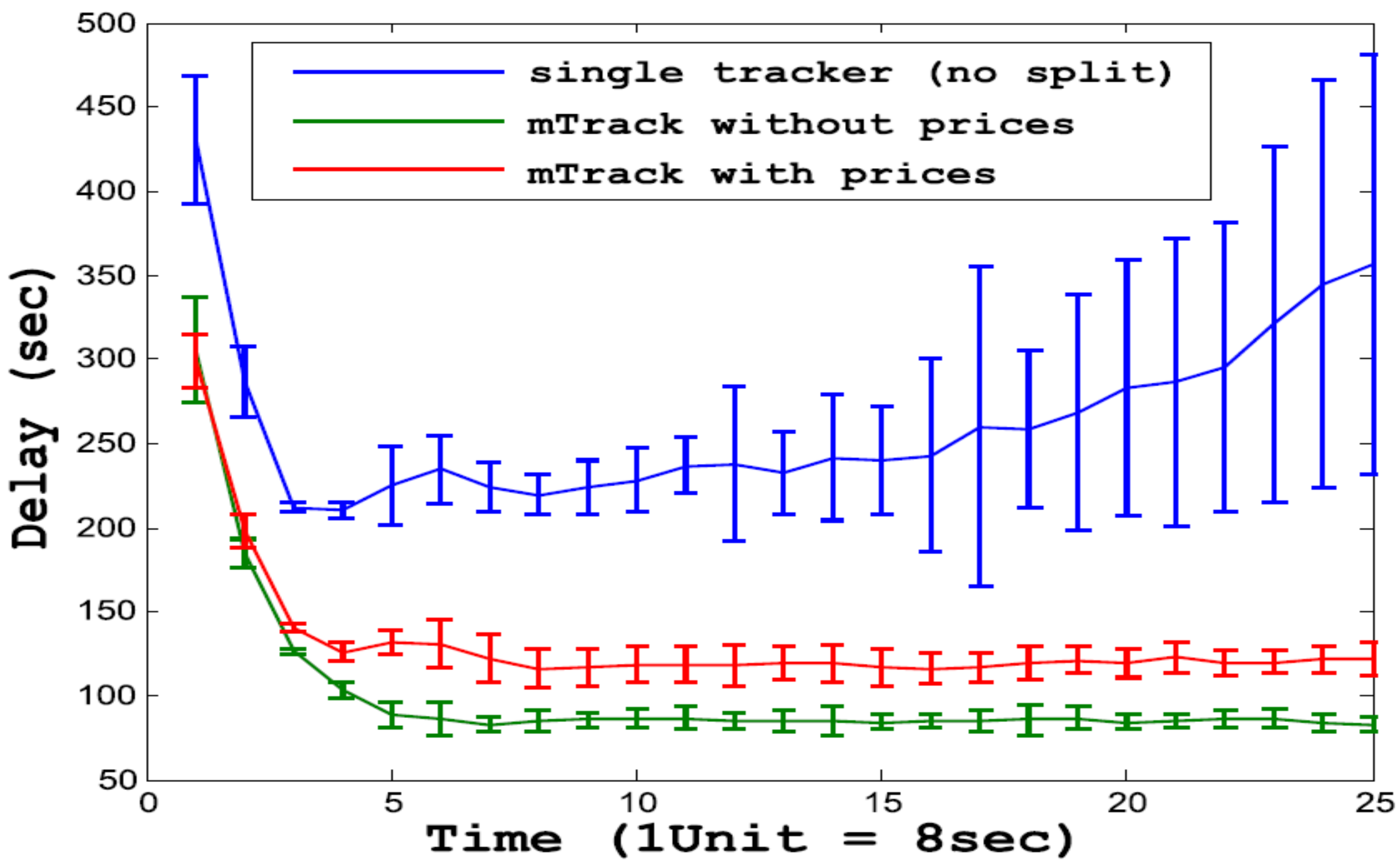
- The total cost of the system when in state X is:

$$\mathcal{C}(\mathbf{X}) = \sum_{i=1}^{Q} \left\{ \frac{\sum_{r=1}^{Q} x_r^i}{C^i - \sum_{l=1}^{Q} x_l^i} + \sum_{r=1}^{Q} p_r^i x_r^i \right\}$$

Delay **Transit cost**

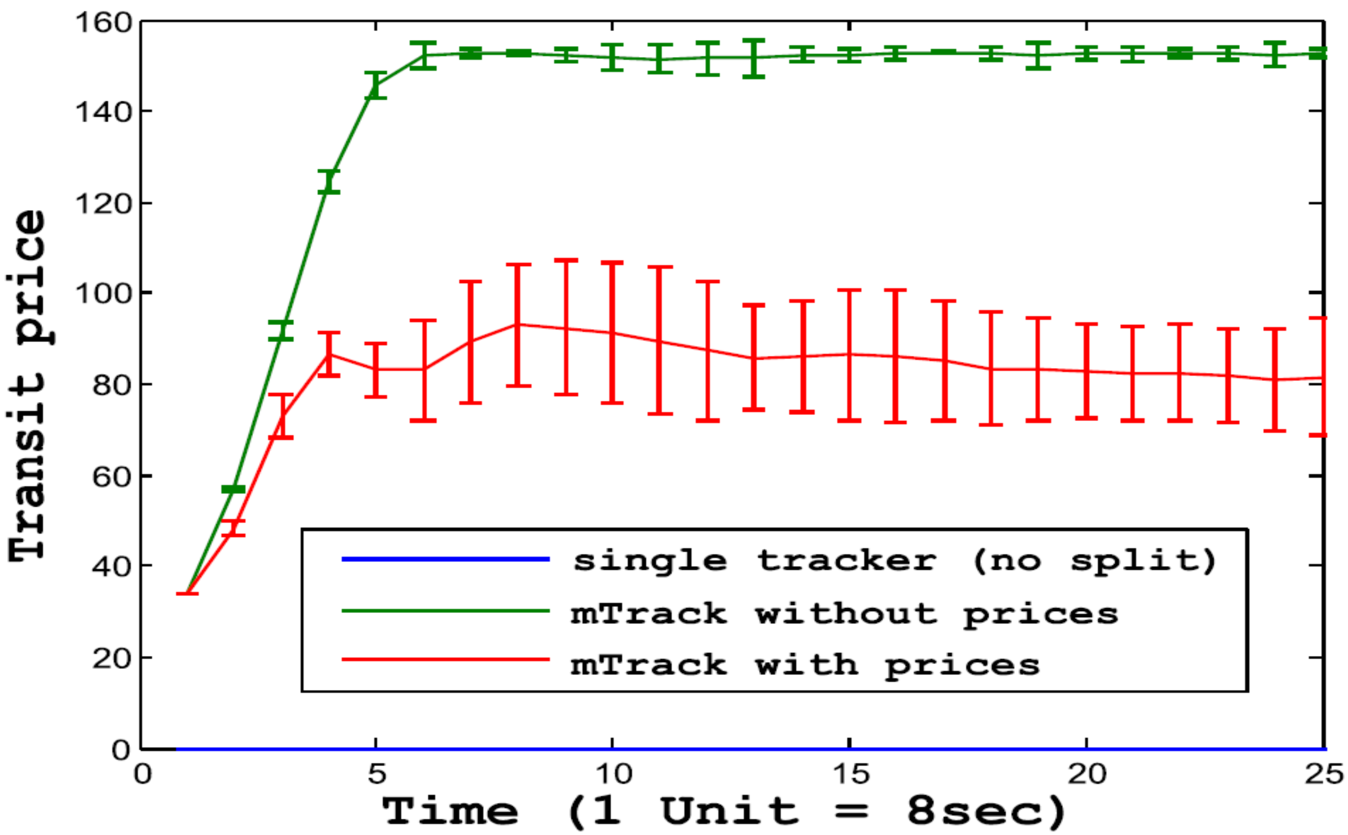
- We use $\mathcal{C}(\mathbf{X})$ as our Lyapunov function
- We prove that the system of ***mTrackers*** that uses negative replicator dynamics is *globally asymptotically stable.*

Delay

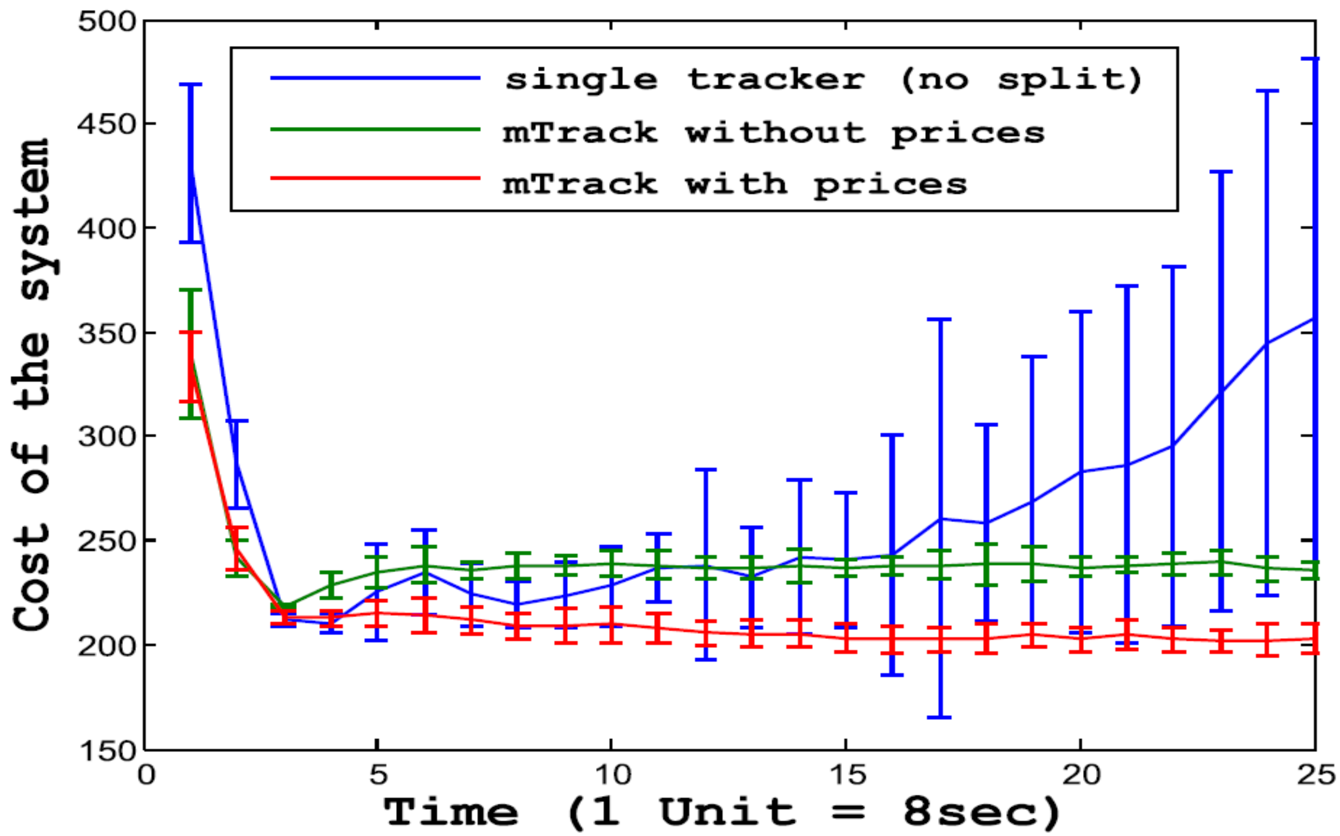


single tracker (no split)
mTrack without prices
mTrack with prices
Delay (sec)
Time (1Unit = 8sec)
500
450
400
350
300
250
200
150
100
50
0
5
10
15
20
25

Transit Cost



Total Cost (Delay + Transit)



Insights and ongoing work

- *Key insight:*
  It is possible to align incentives in terms of delay of a P2P user and the transit costs of an ISP.

- *Ongoing work:*
  Admission Control.
  Potential testbed.