On Selfish Routing In Internet-like Environments

Lili Qiu (Microsoft Research)
Yang Richard Yang (Yale University)
Yin Zhang (AT&T Labs – Research)
Scott Shenker (ICSI)

ACM SIGCOMM 2003

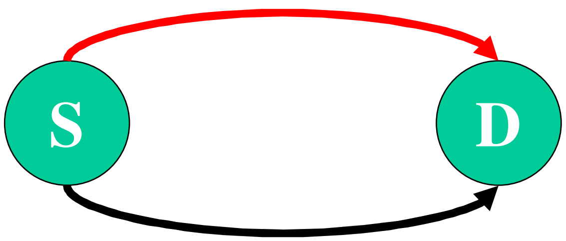
Selfish Routing

- IP routing is sub-optimal for user performance
  - Routing hierarchy and policy routing
  - Equipment failure and transient instability
  - Slow reaction (if any) to network congestion
- Autonomous routing: users pick their own routes
  - Source routing (e.g. Nimrod)
  - Overlay routing (e.g. Detour, RON)
- Autonomous routing is selfish by nature
  - End hosts or routing overlays greedily select routes
  - Optimize their own performance goals
  - … without considering system-wide criteria

Bad News

- Selfish routing can seriously degrade performance [Roughgarden & Tardos]

$L_1(y) = 1$



$L_0(x) = x^n$

Total load: $x + y = 1$

Mean latency: $x\, L_0(x) + y\, L_1(y)$

Worst-case ratio is unbounded

- Selfish source routing
  - All traffic through lower link
  - ➔ Mean latency = 1
- Latency optimal routing
  - To minimize mean latency, set $x = [1/(n+1)]^{1/n}$
  - ➔ Mean latency → 0 as n → ∞

Questions

- Selfish source routing
  - How does selfish source routing perform?
  - Are Internet environments among the worst cases?
- Selfish overlay routing
  - How does selfish overlay routing perform?
  - Does the reduced flexibility avoid the bad cases?
- Horizontal interactions
  - Does selfish traffic coexist well with other traffic?
  - Do selfish overlays coexist well with each other?
- Vertical interactions
  - Does selfish routing interact well with network traffic engineering?

Our Approach

- Game-theoretic approach with simulations
  - Equilibrium behavior
    - Apply game theory to compute traffic equilibria
    - Compare with global optima and default IP routing
  - Intra-domain environments
    - Compare against theoretical worst-case results
    - Realistic topologies, traffic demands, and latency functions
- Disclaimers
  - Lots of simplifications & assumptions
    - Necessary to limit the parameter space
  - Raise more questions than what we answer
    - Lots of ongoing and future work

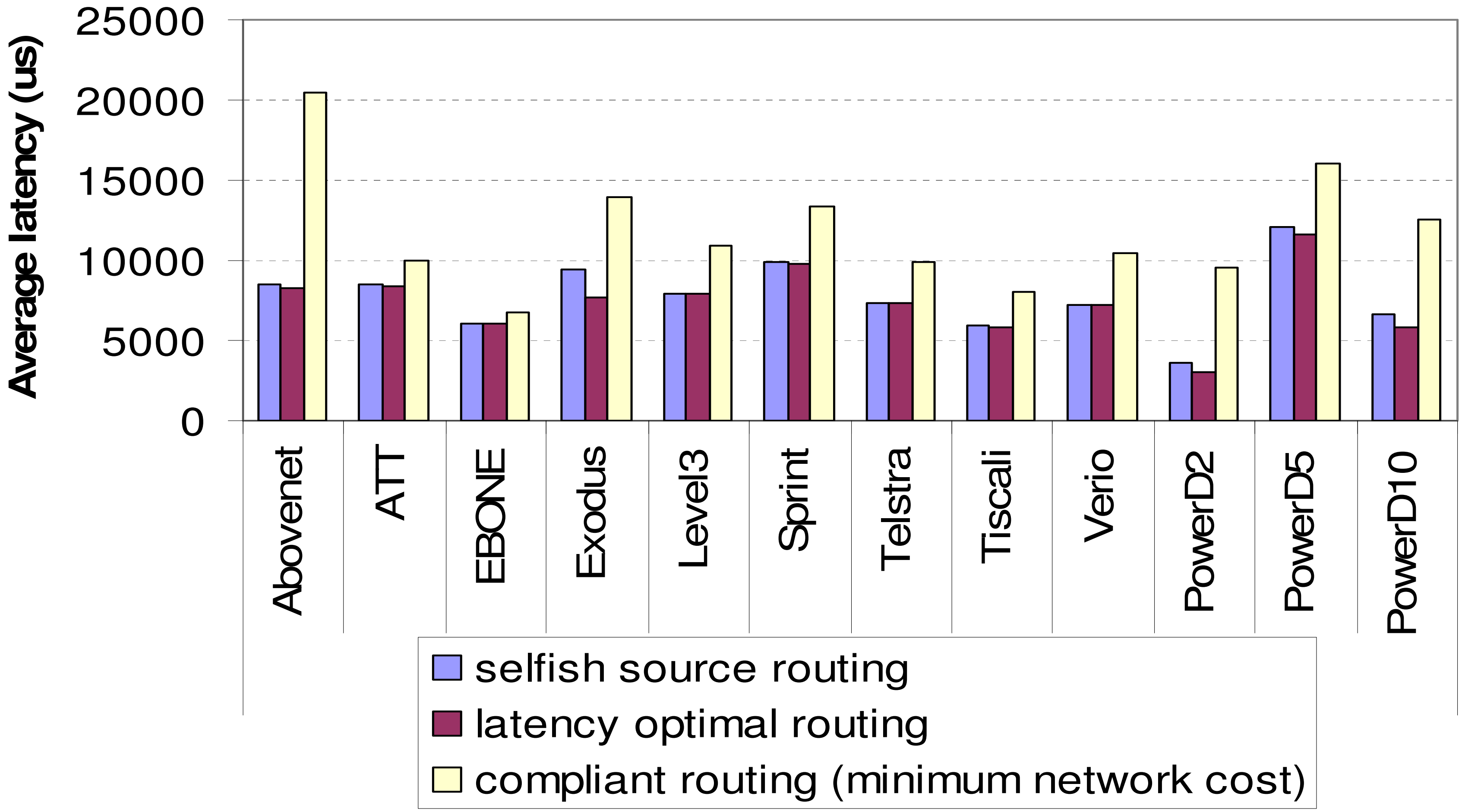
Routing Schemes

- Routing on the physical network
  - Source routing
  - Latency optimal routing
- Routing on an overlay (less flexible!)
  - Overlay source routing
  - Overlay latency optimal routing
- Compliant (i.e. default) routing: OSPF
  - Hop count, i.e. unit weight
  - Optimized weights, i.e. [FRT02]
  - Random weights

Internet-like Environments

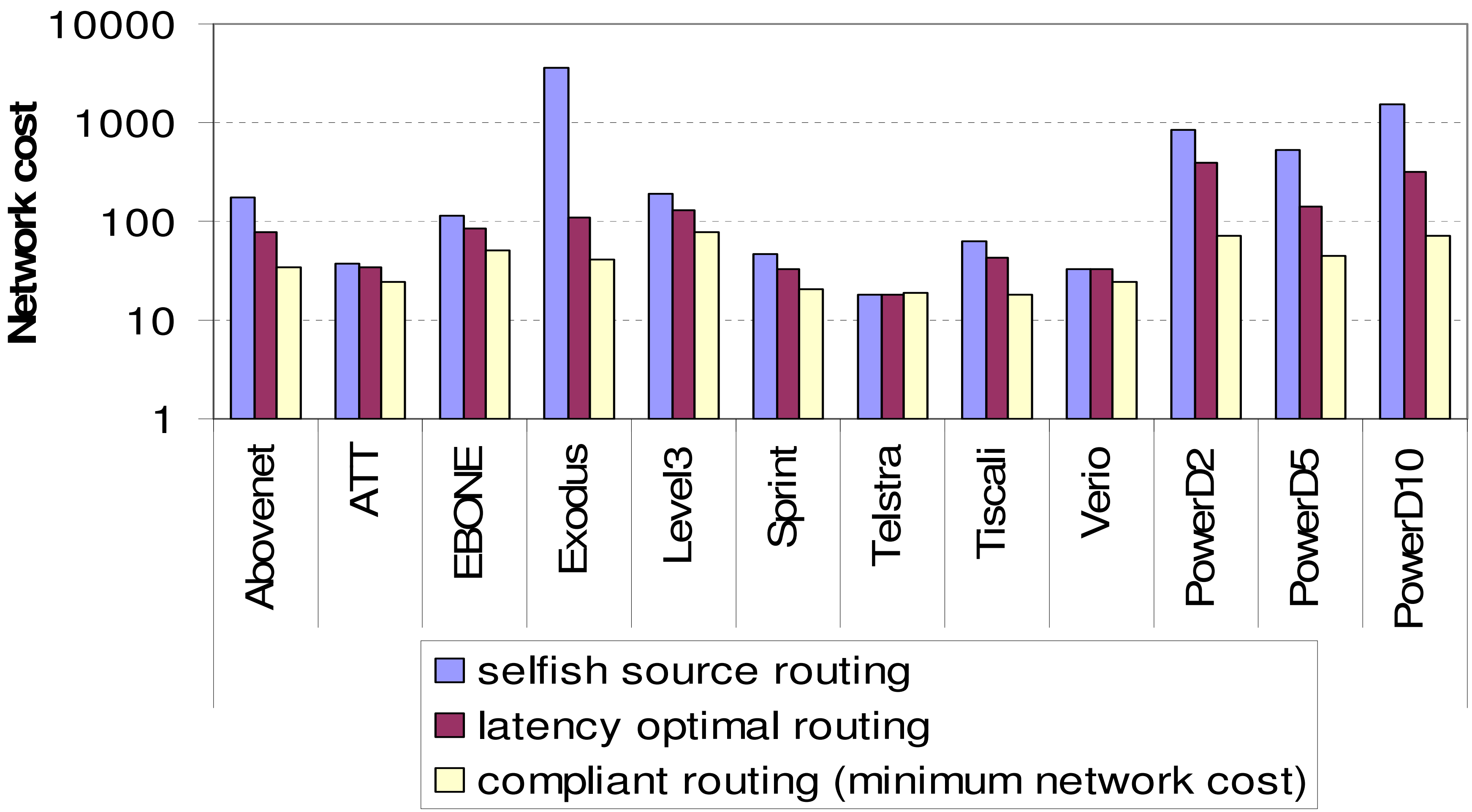
- Network topologies
  - Real tier-1 ISP, Rocketfuel, random power-law graphs
- Logical overlay topology
  - Fully connected mesh (i.e. clique)
- Traffic demands
  - Real and synthetic traffic demands
- Link latency functions
  - Queuing: M/M/1, M/D/1, P/M/1, P/D/1, and BPR
  - Propagation: fiber length or geographical distance
- Performance metrics
  - User: Average latency
  - System: Max link utilization, network cost [FRT02]

Source Routing: Average Latency



Good news: Internet-like environments are far from the worst cases for selfish source routing

Source Routing: Network Cost

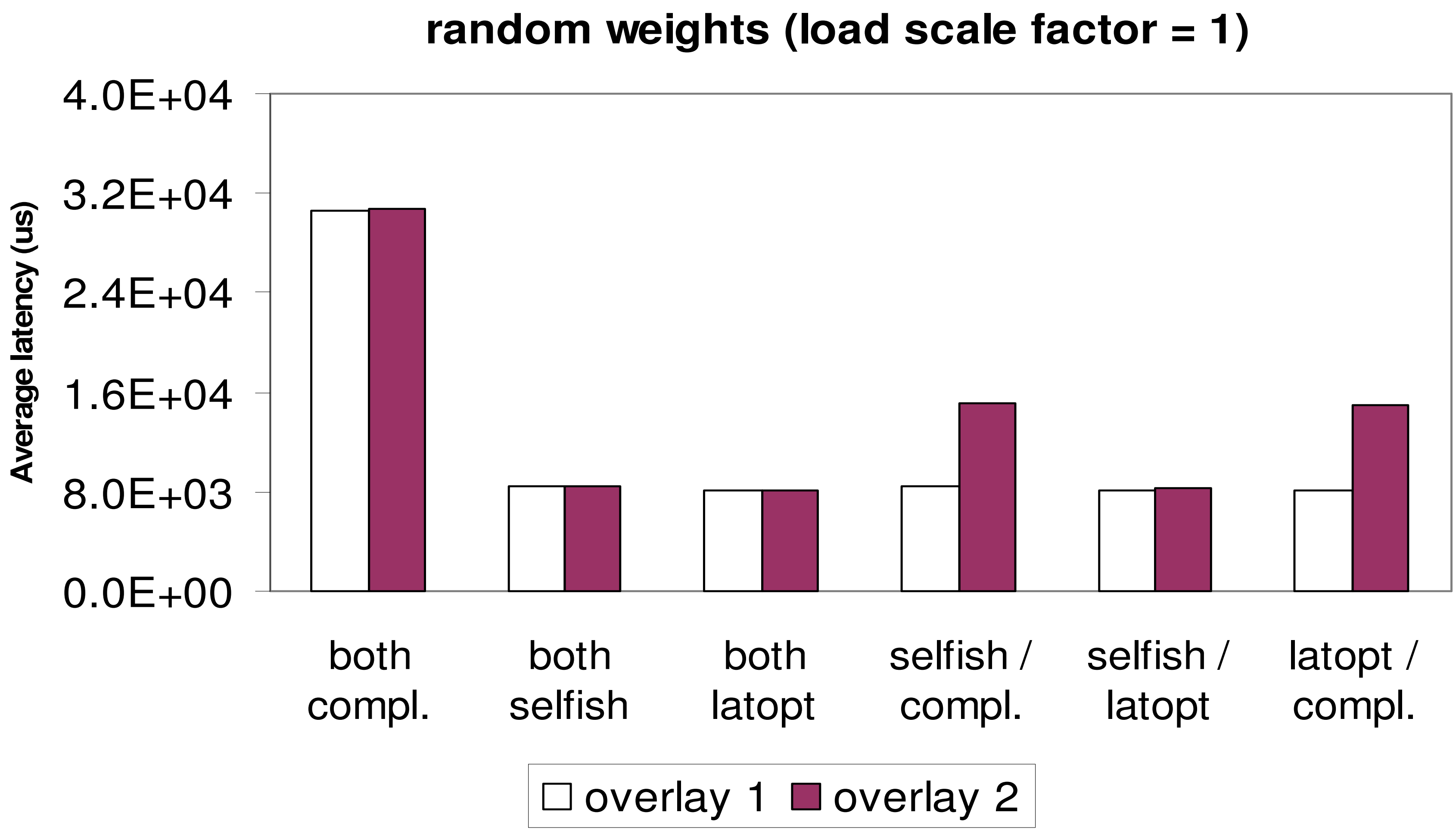


Bad news: Low latency comes at much higher network cost

Selfish *Overlay* Routing

- Similar results apply for overlay routing
  - Achieves close to optimal average latency
  - Low latency comes at higher network cost
- Even if overlay covers a fraction of nodes
  - Random coverage: 20-100% nodes
  - Edge coverage: edge nodes only

Horizontal Interactions

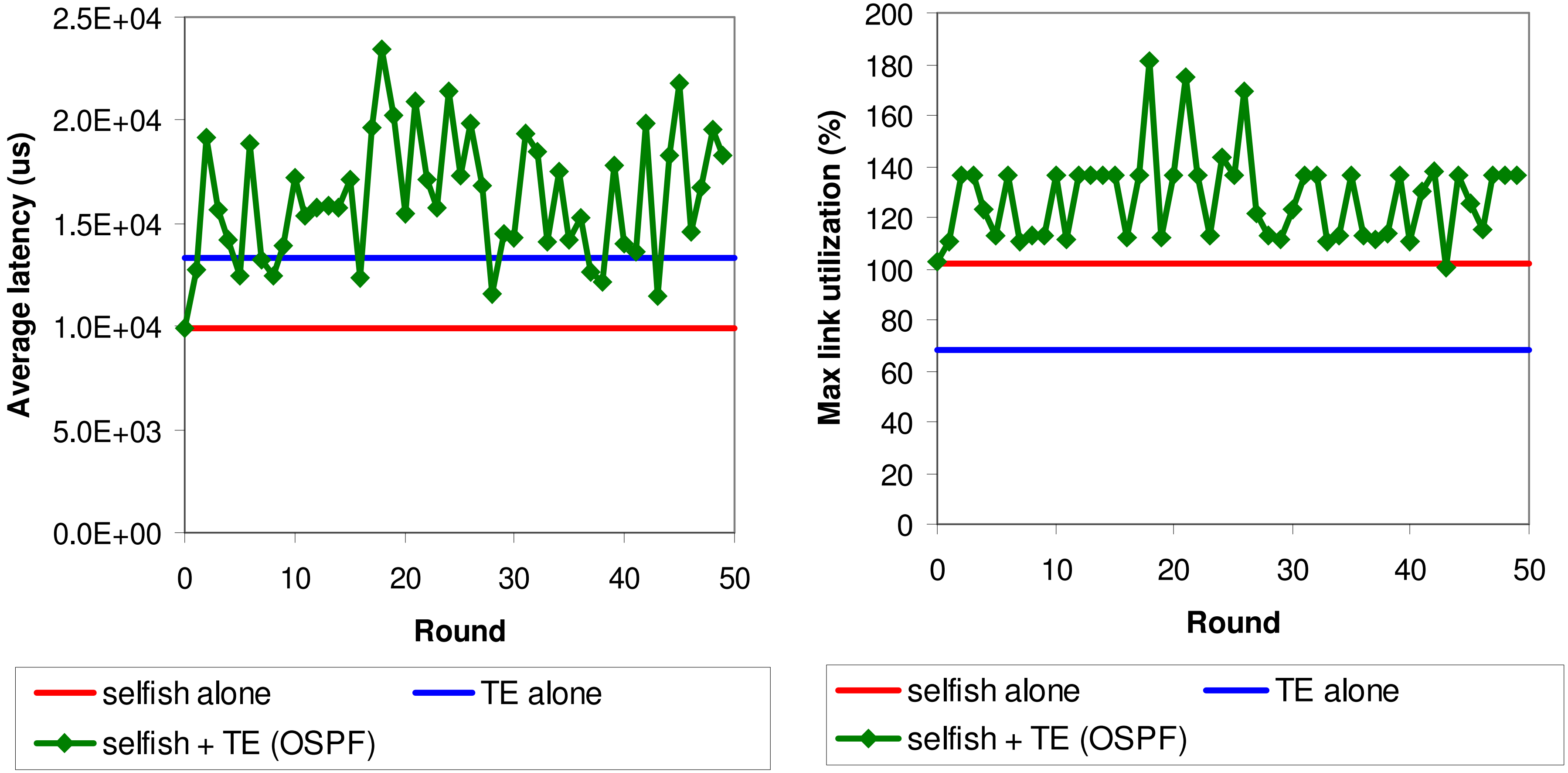


Different routing schemes coexist well without hurting each other.
With bad weights, selfish overlay also improves compliant traffic.

Vertical Interactions

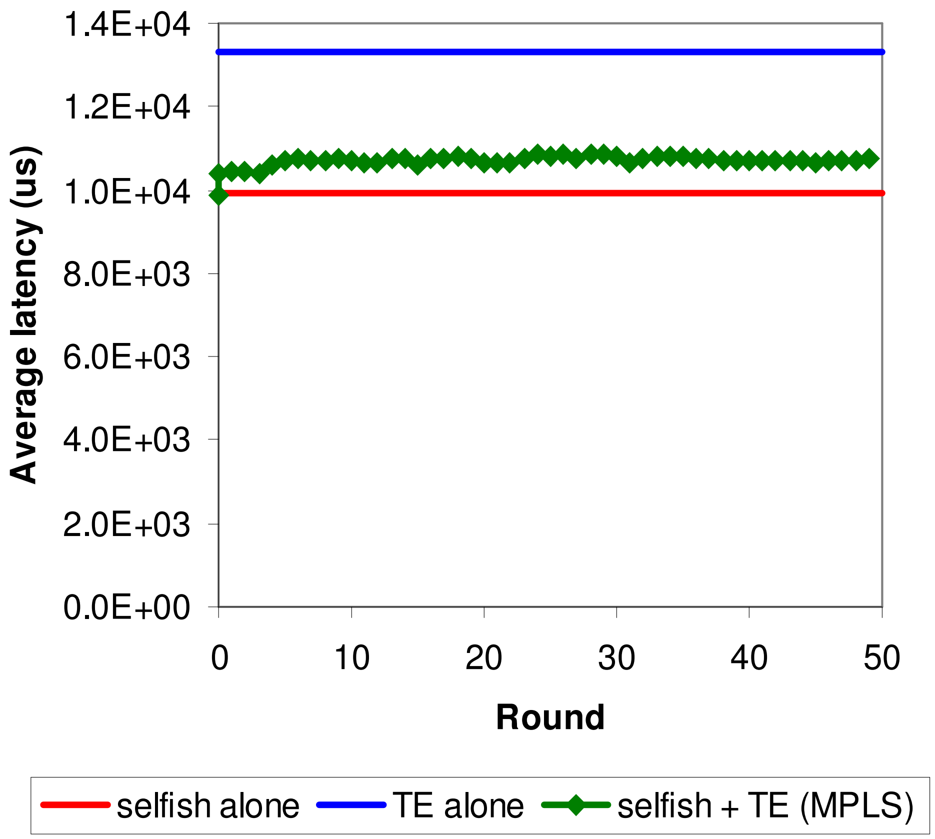
- An iterative process between two players
  - Traffic engineering: minimize network cost
    - current traffic pattern → new routing matrix
  - Selfish overlays: minimize user latency
    - current routing matrix → new traffic pattern
- Question:
  - Does system reach a state with both low latency and low network cost?
- Short answer:
  - Depends on how much control underlay has

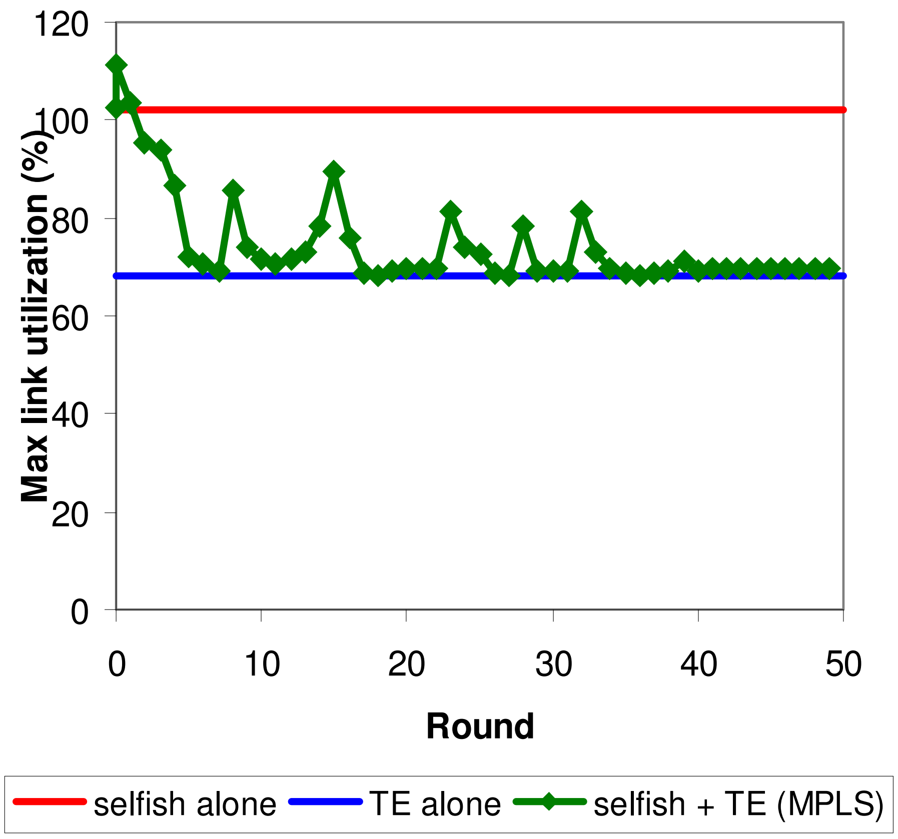
Selfish Overlays vs. OSPF Optimizer



OSPF optimizer interacts poorly with selfish overlays because it only has very coarse-grained control.

Selfish Overlays vs. MPLS Optimizer





MPLS optimizer interacts with
selfish overlays much more effectively.

Conclusions

- Contributions
  - Important questions on selfish routing
  - Simulations that partially answer questions
- Main findings on selfish routing
  - Near-optimal latency in Internet-like environments
    - In sharp contrast with the theoretical worst cases
  - Coexists well with other overlays & regular IP traffic
    - Background traffic may even benefit in some cases
  - Big interactions with network traffic engineering
    - Tension between optimizing user latency vs. network load

Lots of Future Work

- Extensions
  - Multi-domain IP networks
  - Different overlay topologies
  - Alternative selfish-routing objectives
- Study dynamics of selfish routing
  - How are traffic equilibria reached?
- Improve interactions
  - Between selfish routing & traffic engineering
  - Between competing overlay networks

Thank you!

Computing Traffic Equilibrium of Selfish Routing

- Computing traffic equilibrium of non-overlay traffic
  - Use the linear approximation algorithm
    - *A* variant of the Frank-Wolfe algorithm, which is a gradient-based line search algorithm
- Computing traffic equilibrium of selfish overlay routing
  - Construct a logical overlay network
  - Use Jacob's relaxation algorithm on top of Sheffi's diagonalization method for asymmetric logical networks
  - Use modified linear approximation algo. in symmetric case
- Computing traffic equilibrium of multiple overlays
  - Use a relaxation framework
    - In each round, each overlay computes its best response by fixing the other overlays' traffic; then the best response and the previous state are merged using decreasing relaxation factors.