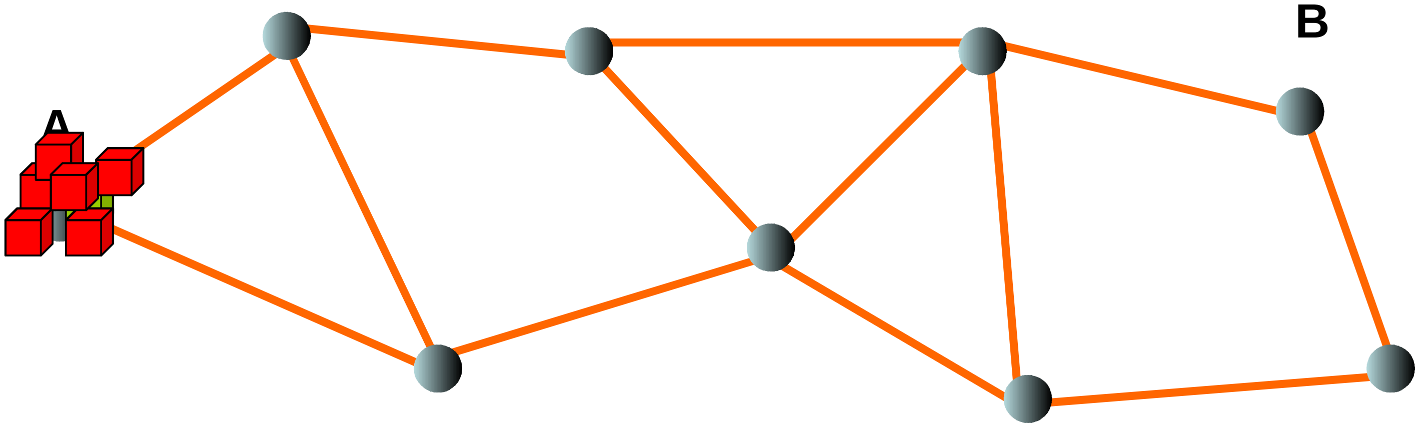
Routing in Wireless and Adversarial Networks

Christian Scheideler

Institut für Informatik

Technische Universität München

Routing



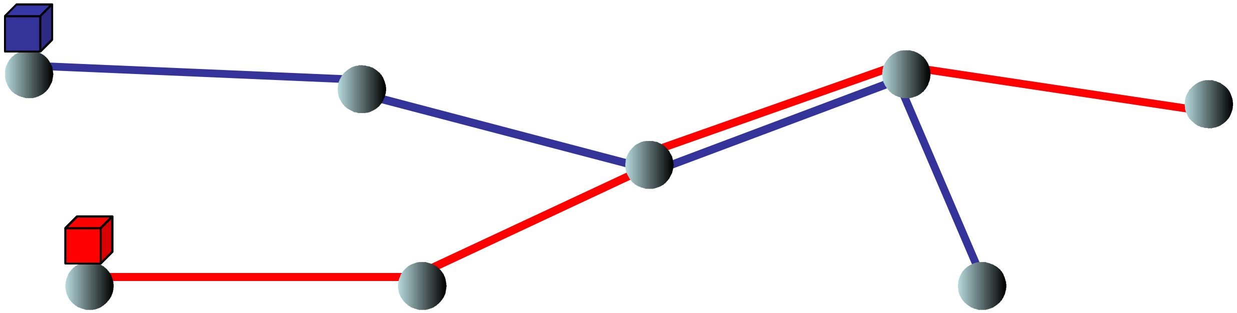
- Path selection
- Scheduling
- Admission control

Classical Routing Theory

Given a path collection with

- congestion C (max. number of paths over edge) and
- dilation D (max. length of a path)

find (near-)optimal schedule for packets.



Classical Routing Theory

Leighton, Maggs, Rao 88:

There is a schedule with O(C+D) runtime.

Also for non-uniform edges [Feige & S 98]

Since then many randomized online protocols with runtime ~O(C+D) w.h.p.

Basic techniques: random delays or ranks

Classical Routing Theory

Extensions faulty and wireless networks.

Adler & S 98:

- $G=(V,E)$ with probabilities $p:E$! $[0,1]$
- $H=(V,E)$ with latencies $l(e)=1/p(e)$
- Valid routing schedule of length T for H can be simulated in G in time $O(T \log L + L \log n)$, w.h.p.; L: max. latency

Scheduling

Classical model: batch-like scheduling

More relevant models:

- Stochastic injection models (packets are continuously injected using Poisson distribution or Markov chains)
- Adversarial queueing theory (introduced by Borodin et al. 96)

Adversarial Queueing Theory

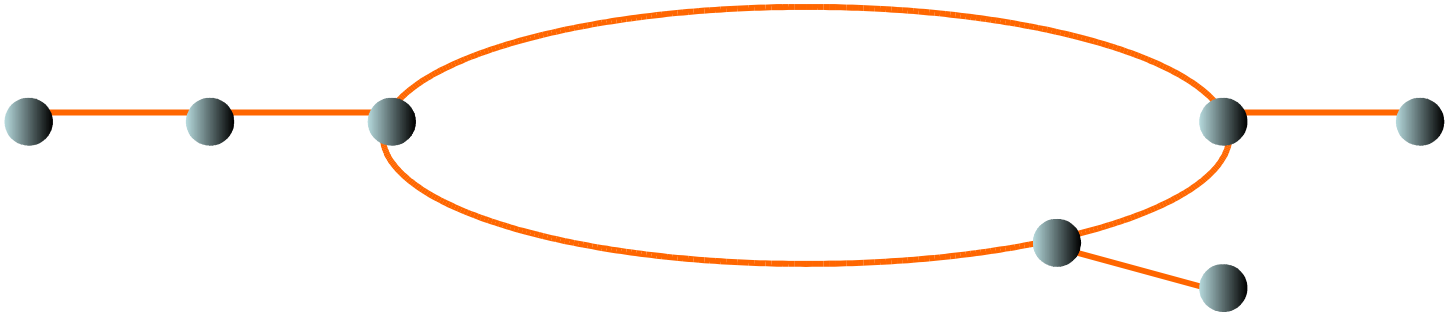
Basic model:

- Static network $G=(V,E)$
- (w,λ)-bounded adversary continuously injects packets subject to the condition that for all edges e and all time intervals of length w, it injects at most λw packets with paths containing e
- All packets have to be delivered ($\lambda<=1$)

Adversarial Queueing Theory

Basic results:

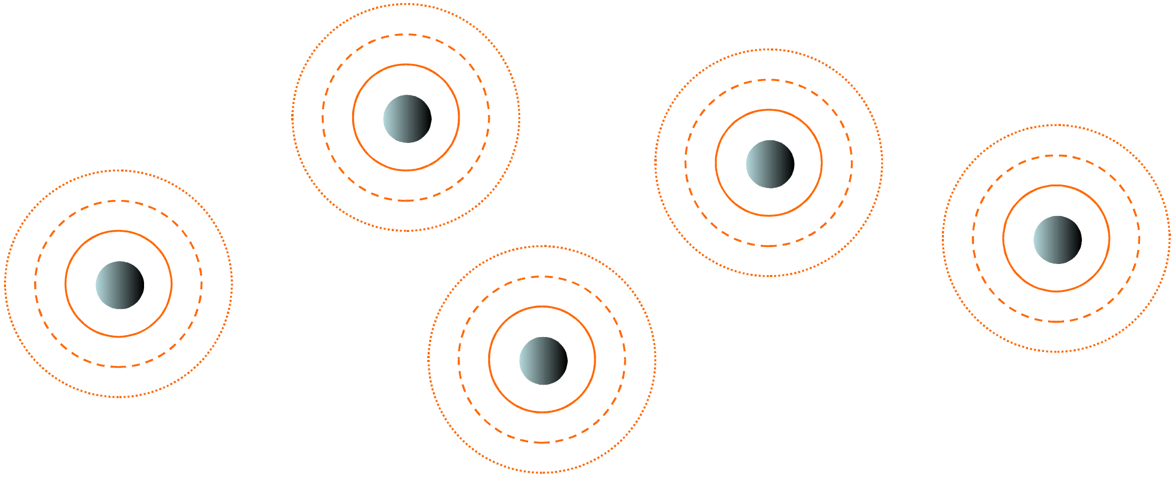
- Universal stability and instability of various queueing disciplines (FIFO, SIS, LIS, NTO,…)
- Universal stability of networks



Adversarial Queueing Theory

Networks with time-varying channels:

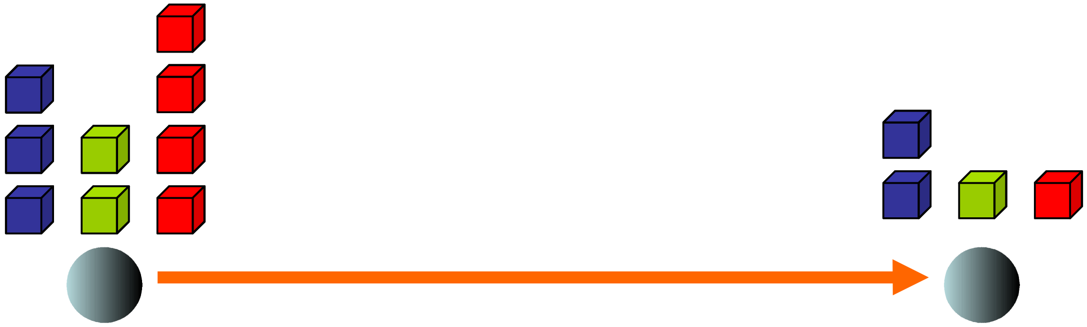
- Packet injections and edges under adversarial control
- Andrews and Zhang 04: Variant of NTO is universally stable in this model



Adversarial Routing Theory

Paths are not given to system:

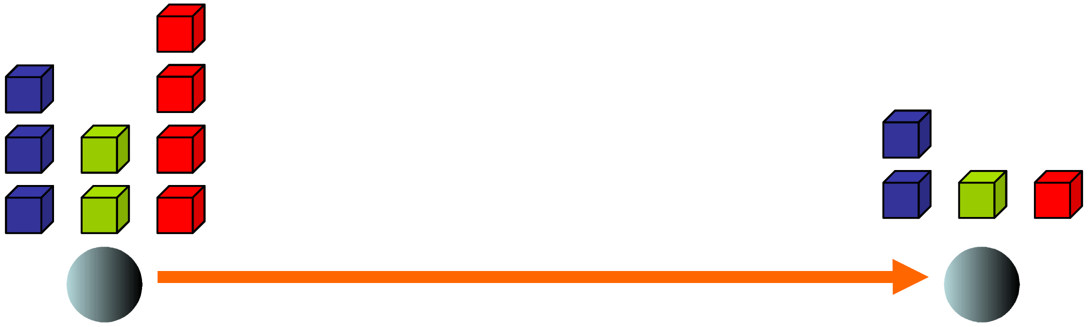
- Aiello, Kushilevitz, Ostrovsky, Rosen ’98: local load balancing techniques can be used to keep queues bounded



Adversarial Routing Theory

Paths are not given to system:

- Awerbuch, Brinkmann & S ’03:
  local load balancing technique with bounded queues also handles admission, works even for adversarial networks



Adversarial Routing Theory

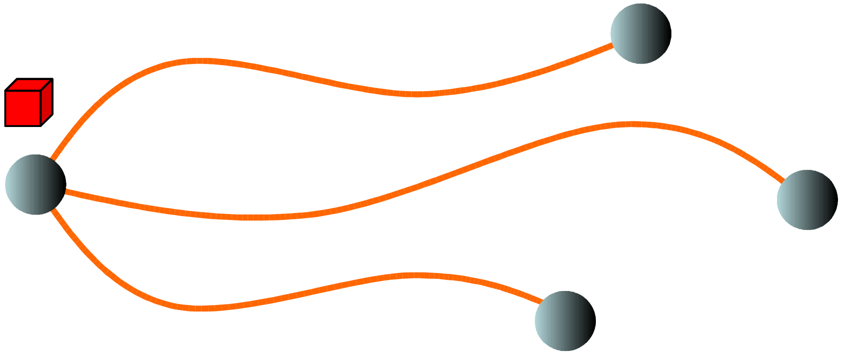
Paths are not given to system:

- Awerbuch, Brinkmann & S ’03:
  load balancing technique with $O(L/\varepsilon)$ times buffer space of OPT is $(1+\varepsilon)$-competitive w.r.t. throughput; L: max path length

Path Selection

Problems:

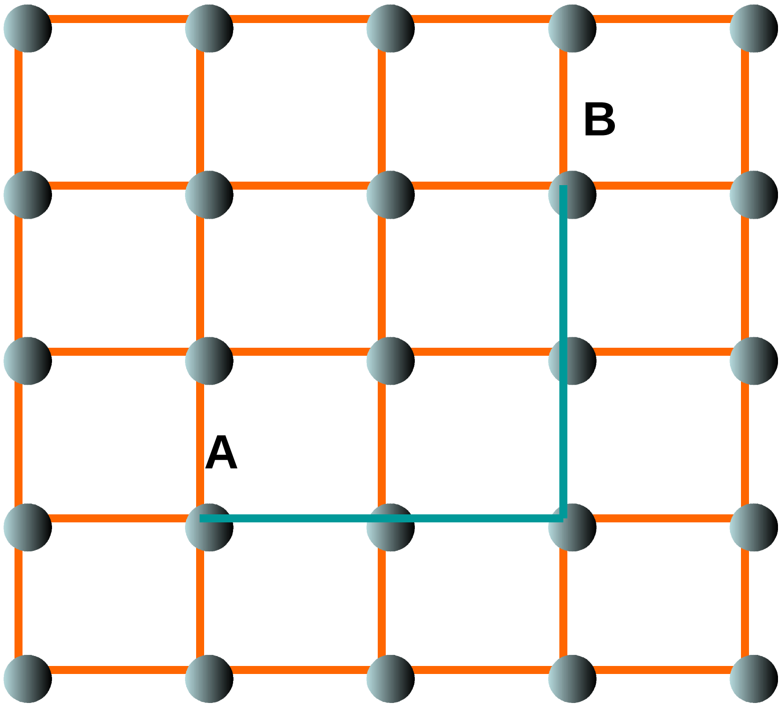
- packet-based paths: slow delivery
- destination-based paths: congestion



Better: source-based path selection
(MPLS: Multiprotocol Label Switching)

Path Selection

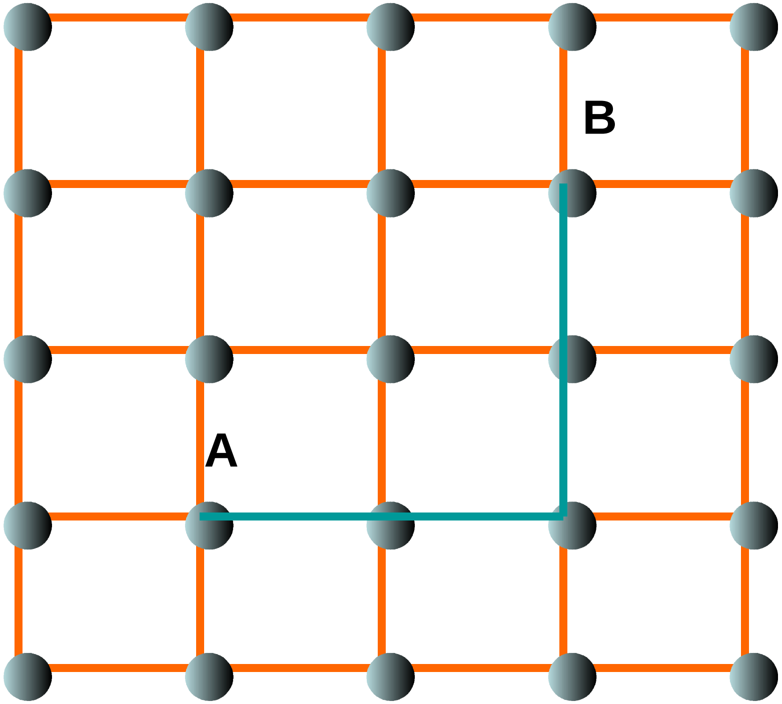
Classical work: path selection strategies for specific networks (n£n-mesh)



x-y routing

Path Selection

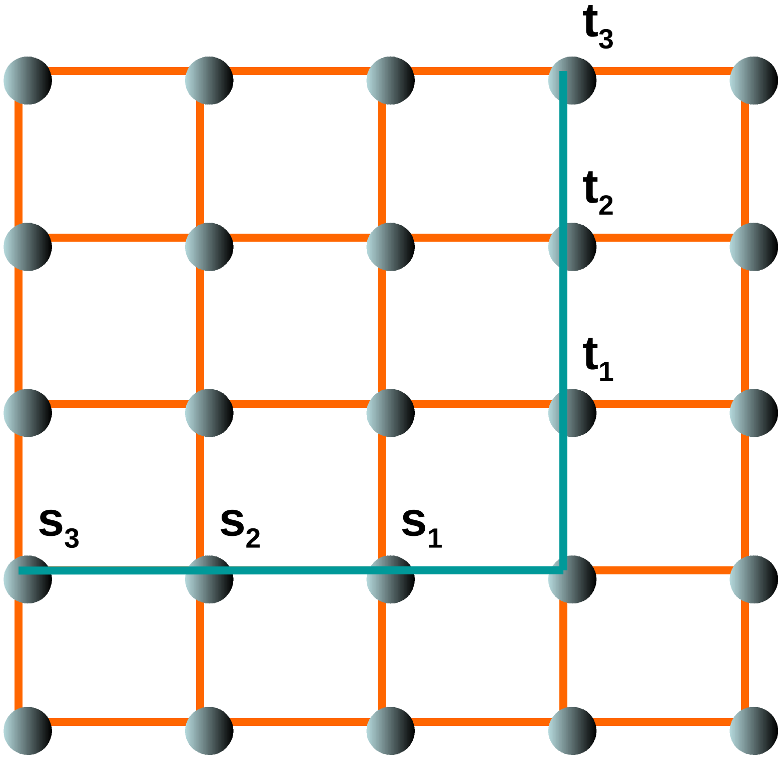
x-y routing: ~worst-case optimal congestion and dilation for permutation routing



x-y routing

Path Selection

x-y routing: far from optimal in general

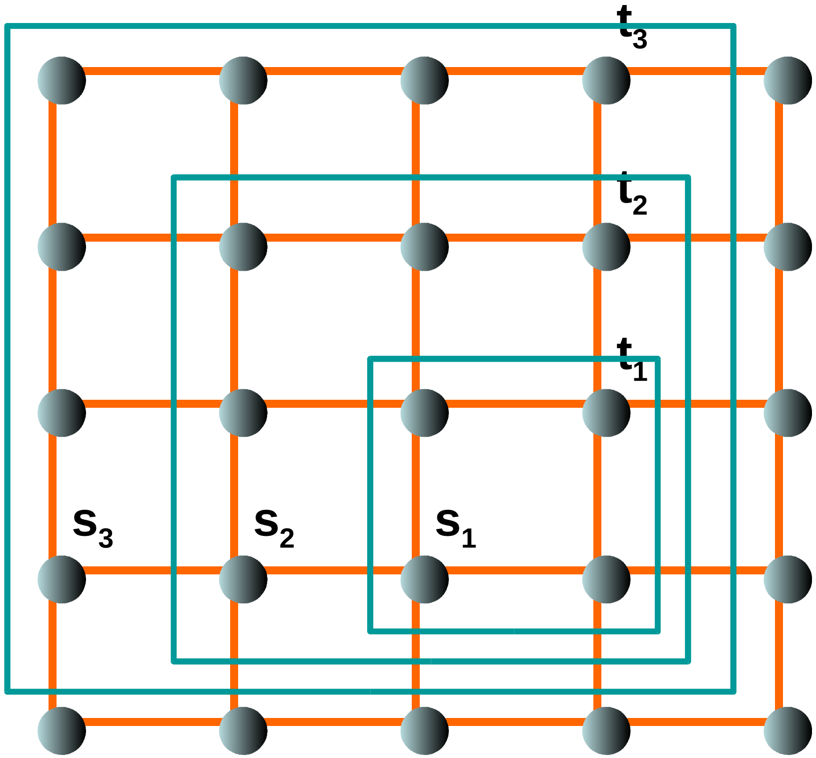


x-y routing

Path Selection

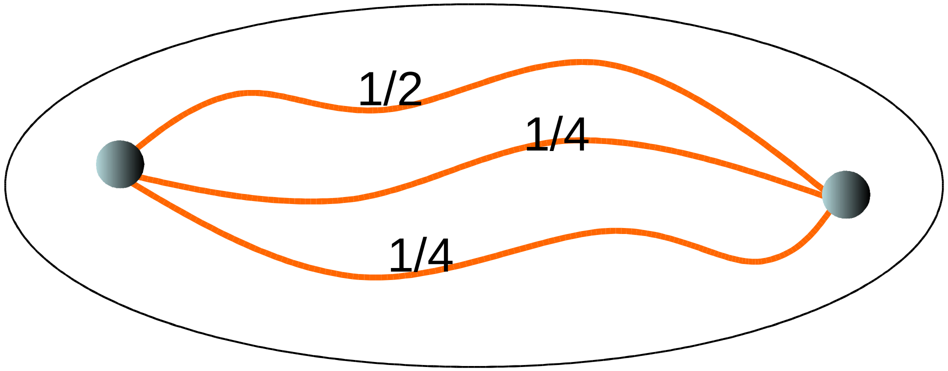
Trick: use hierarchical randomized routing.

$\Theta(\log n)$-competitive for any problem



Oblivious Path Selection

Räcke 02: For any network with edge capacities, path collections for random path selection can be set up for every source-destination pair s.t. the expected congestion of routing any routing problem is $O(\log^3 n)$-competitive.



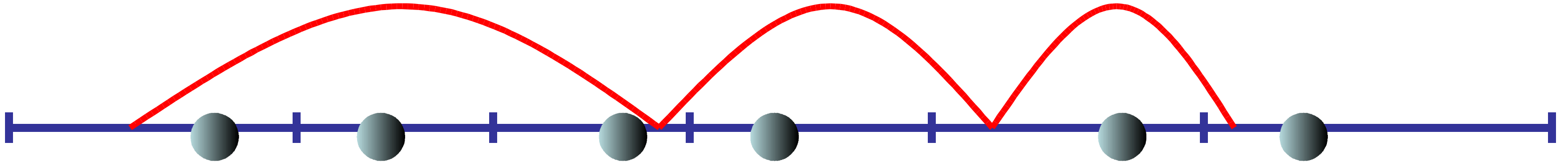
Best bound [HHR03]: $\sim O(\log^2 n)$

Oblivious Path Selection

Also works well for certain dynamic networks for peer-to-peer systems.

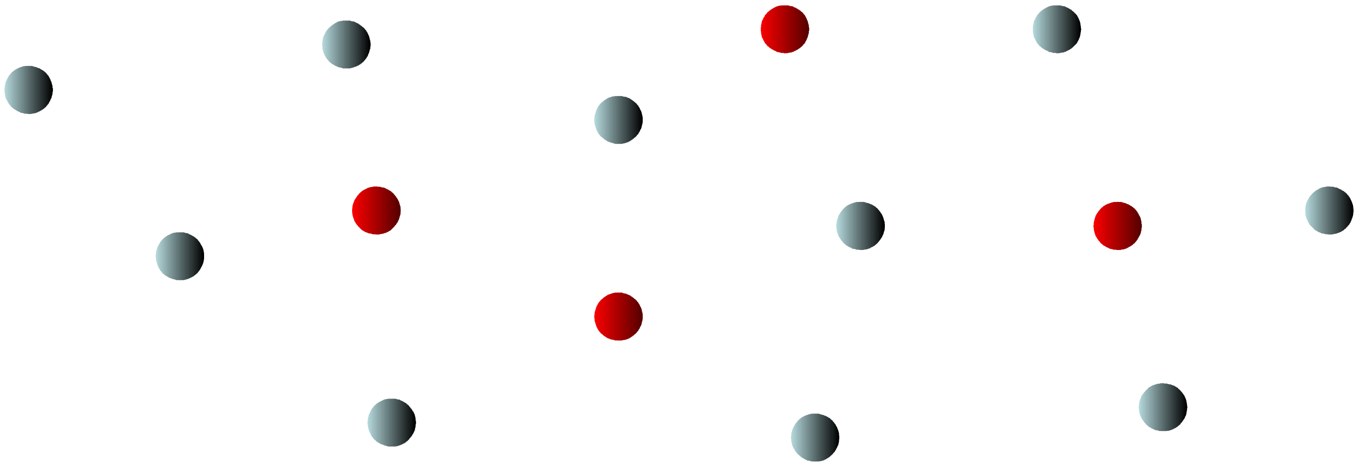
Trick: continuous-discrete approach

- route in virtual space
- nodes partition virtual space among them



Oblivious Path Selection

Does not work well for wireless, unknown or adversarial networks (e.g., unstructured P2P systems with adversarial presence)



Adaptive Path Selection

Basic Idea: Garg & Könemann 98

Multicommodity flow problem: collection of commodities (source, dest., demand)

- Solution 1: use LP
- Solution 2: combinatorial approach
  (path packing using primal-dual approach)

Garg-Könemann Framework

Problem: MCF (maximum concurrent flow problem), i.e., given commodities with demands d_i, find flows of value d_i for commodities s.t. $\max_e f_e/c_e$ minimized

Goal: find $(1+\varepsilon)$-approximate solution via path packing

Garg-Könemann Framework

Initially, $f_e^i=0$ for all commodities i and edges e

Algorithm runs in $T=\ln m/\varepsilon^2$ phases, routes a flow of d_i/T for each commodity i in each phase

A phase consists of several steps

In each step, flows augmented simultaneously subject to two constraints:

- $(1+\varepsilon)$-shortest paths constraint, using edge lengths $l_e = m^{cong_e/\varepsilon}/c_e$ with $cong_e = f_e/c_e$
- step-size constraint: $\Delta l_e <= \varepsilon\ l_e$ (which implies $\Delta f_e <= \varepsilon^2\ c_e/\ln m$)

Garg-Könemann Framework

Original Garg-Könemann approach:

- Route commodities in round-robin fashion, one commodity per step
) #steps depends linearly on #commidities

Awerbuch, Khandekar and Rao 07:

- Route commodities simultaneously in each step using capacities $c_e^i = \varepsilon^2 f_e^i/\log m$ for comm i
) multiplicative-increase strategy, faster conv

Garg-Könemann Framework

Awerbuch, Khandekar and Rao 07:
runtime $O(L \log^3 m \log k)$
L: max flow length, k: #commodities

- L small (hypercube): fast convergence
- L always boundable by expansion of net (flow shortening lemma [Kolman & S 02])

Oblivious vs. Adaptive

Congestion for arbitrary routing problems in hypercubic networks:

- Oblivious path selection:
  $\Theta(\log n)$-competitive, paths instantly, update of path system complicated
- Adaptive path selection:
  $(1+\varepsilon)$-competitive, paths in polylog comm rounds, continuous updates easy

Adaptive Path Selection

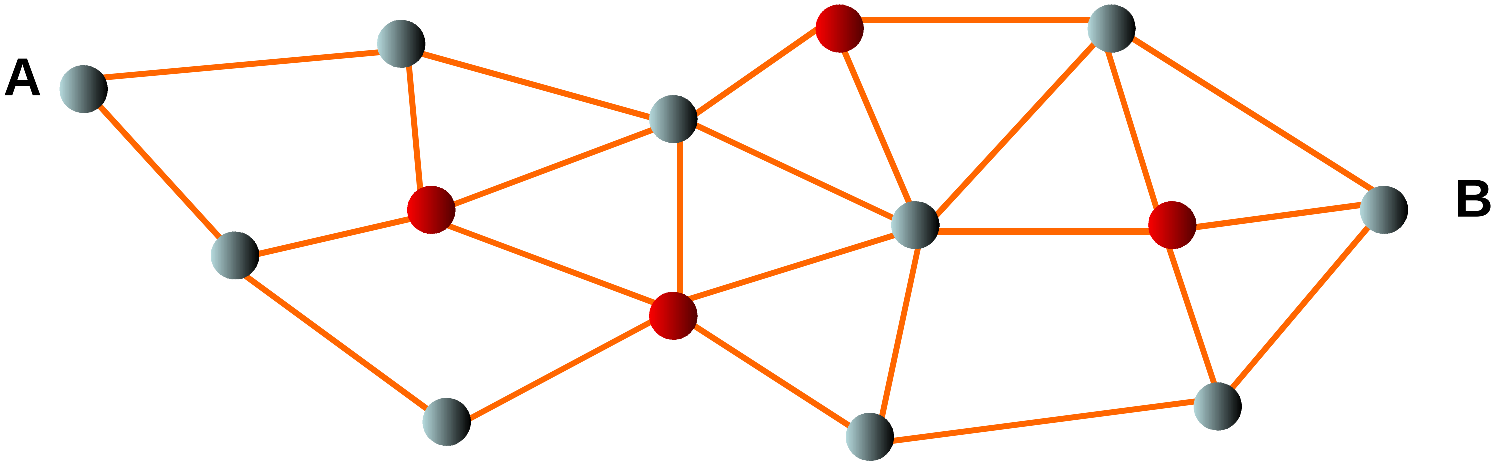
Problem: previous approaches not stateless resp. self-stabilizing

Awerbuch and Khandekar 07:

- Adaptive path selection strategy that only needs to know current state
- Fast convergence through greedy strategy based on multiplicative increase, additive decrease

Adversarial Path Selection

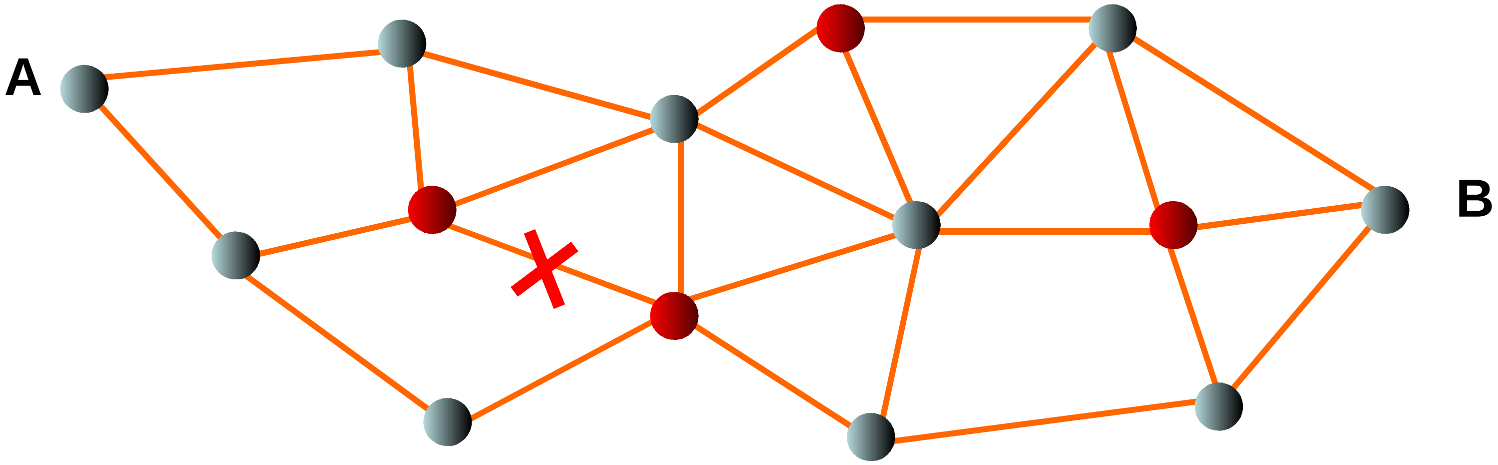
Scenario I: Adversaries part of network, but path along honest nodes available



Adversarial Path Selection

Basic approach: A fixes a path from A to B.

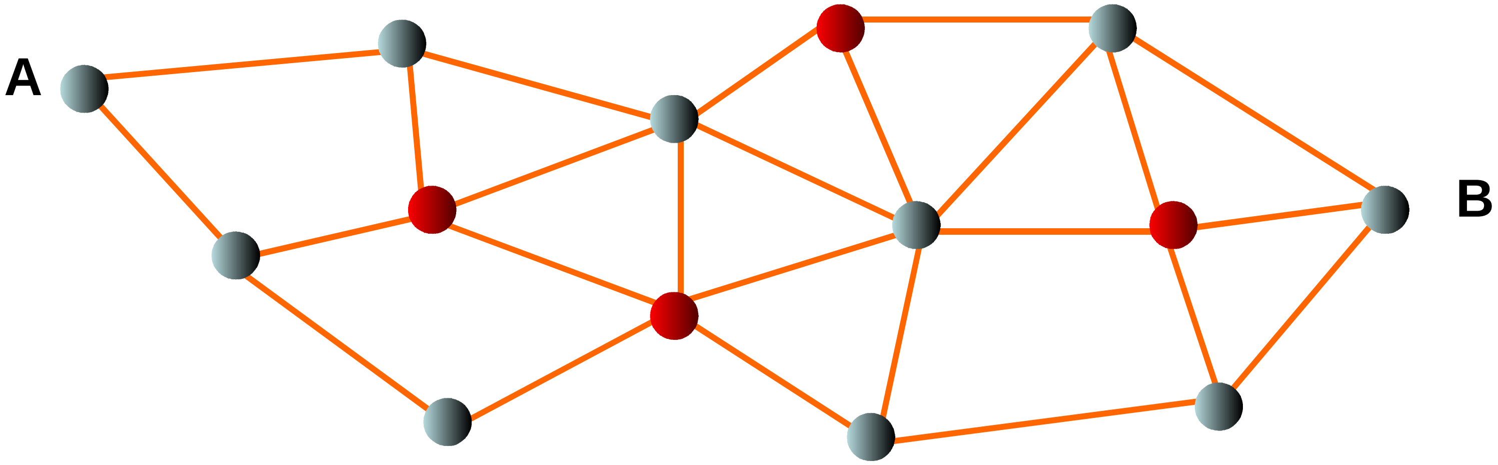
Path does not work: A identifies bad edge.



Adversarial Path Selection

Identification of bad edge:

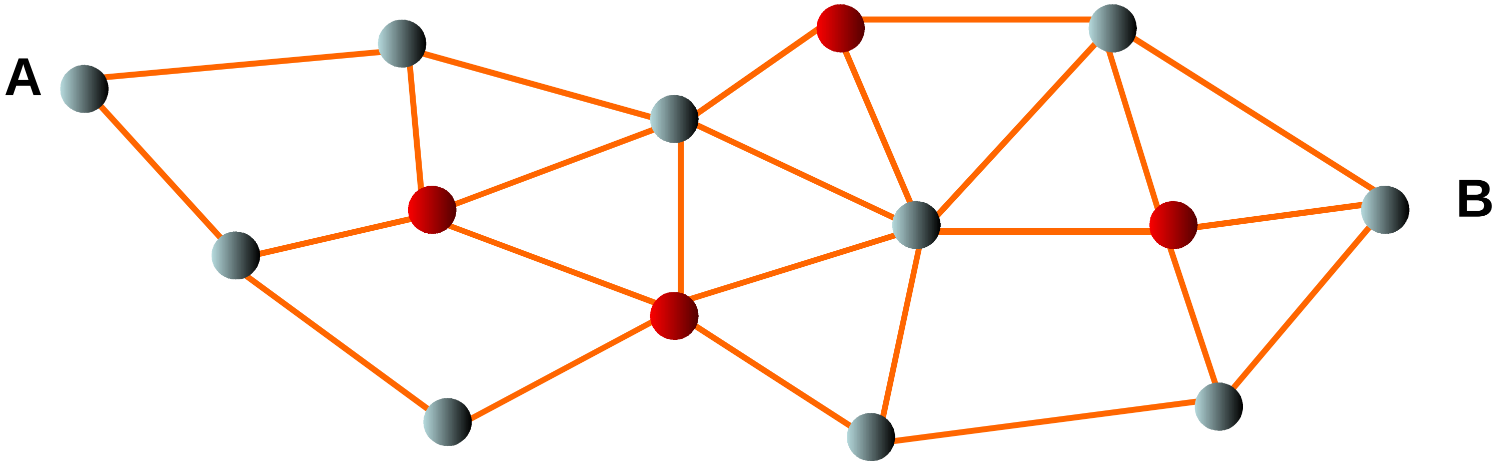
Acknowledgements via binary search



Adversarial Path Selection

Maximum number of attempts: m (# edges)

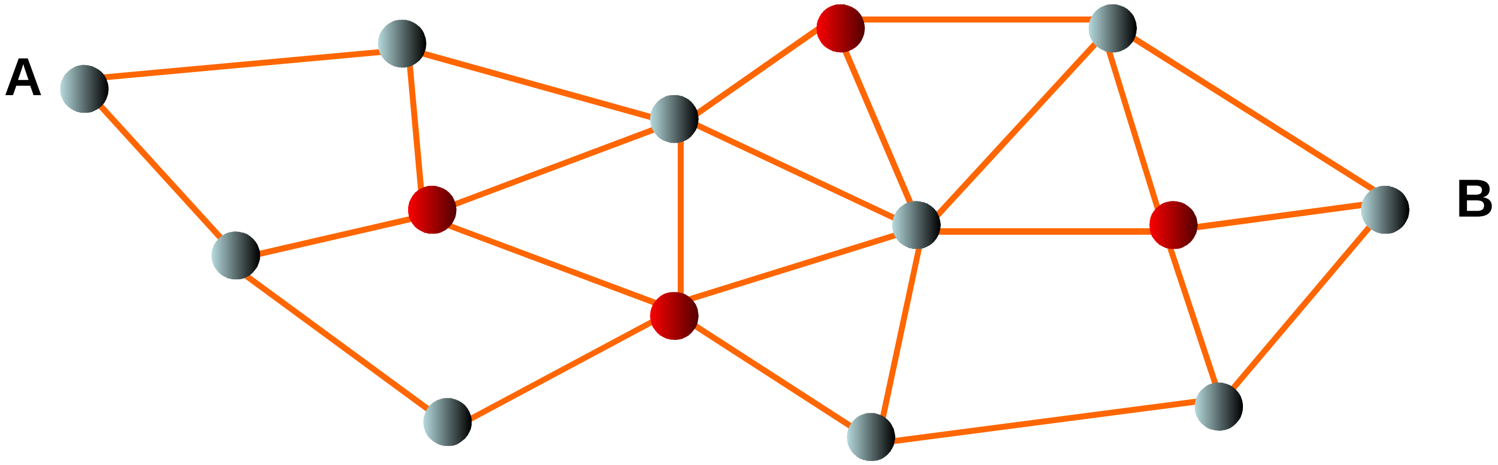
Either successful or edge killed.



Adversarial Path Selection

Improvement: use recommendations

If neighbor knows better, suggests a diff path



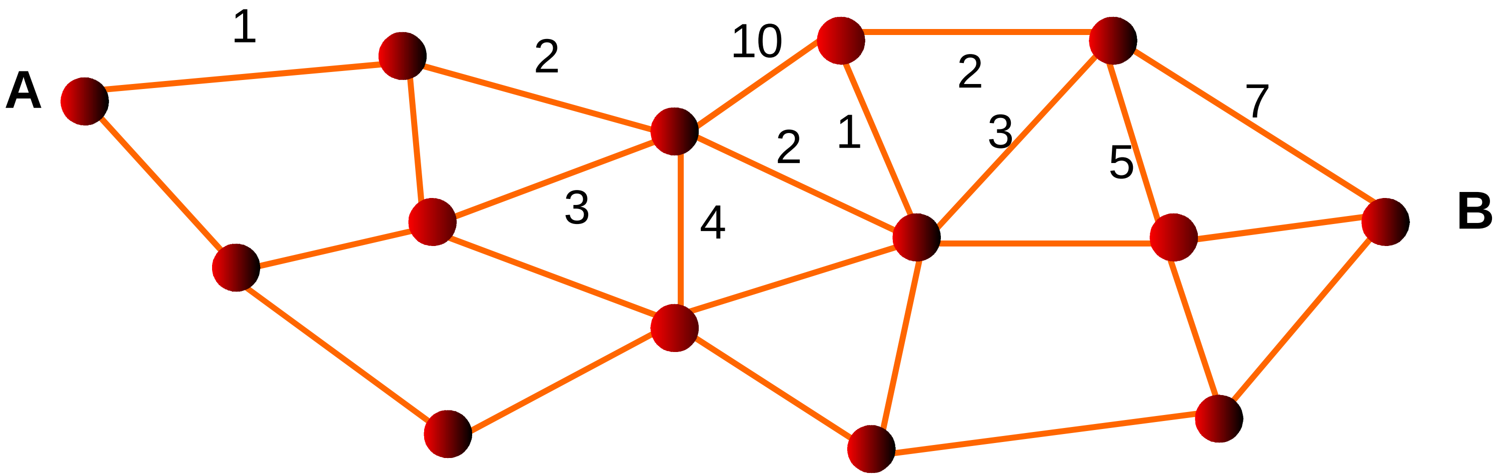
! collaborative learning

Adversarial Path Selection

Scenario II: All nodes adversarial.

Awerbuch and Kleinberg 04:

Learns best static path in hindsight



Adversarial path selection

Model:

- There is a set S of static strategies (paths)
- Algorithm A interacts with adversary for T steps
- In each step j, the adversary picks a cost function c_j:S ! IR and A picks a random strategy x_j 2 S
- Only cost of chosen strategy revealed to A
- The regret of the algorithm A is defined as

$$R(A) = E[\sum_j c_j(x_j) – \min_{x\ 2\ S} \sum_j c_j(x)]$$

Adversarial Path Selection

Awerbuch and Kleinberg:

- Regret of $O(T^{2/3}\ C\ m^{5/3})$ against oblivious adversary
  C: maximum cost difference, m: #edges

- Regret of $O(T^{2/3}\ C^{7/3}\ m^{1/3})$ against adaptive adversary

Regret does not depend on $|S|$!

Adversarial Path Selection

Otto von Bismarck:

Fools learn from experience; wise men learn from the experience of others.

Only collaborative learning result due to Awerbuch and Hayes 07, who study the dynamic regret for $|S|=2$:

$$R(A) = \mathrm{avg}_a\, E[\textstyle\sum_j c_j(x_j) - \sum_j \min_{x\ 2\ S} c_j(x)]$$

Adversarial Path Selection

Awerbuch and Hayes 07:

- N agents, n of which are honest
- In each round, agents make decisions in a fixed order, report the costs incurred
- Costs are either 0 or 1
- Dynamic regret: $O(\log N^2 + T/n)$

  $\log^2 N$: rounds to figure out whom to trust

  T/n: just one mistake per round

Adversarial Path Selection

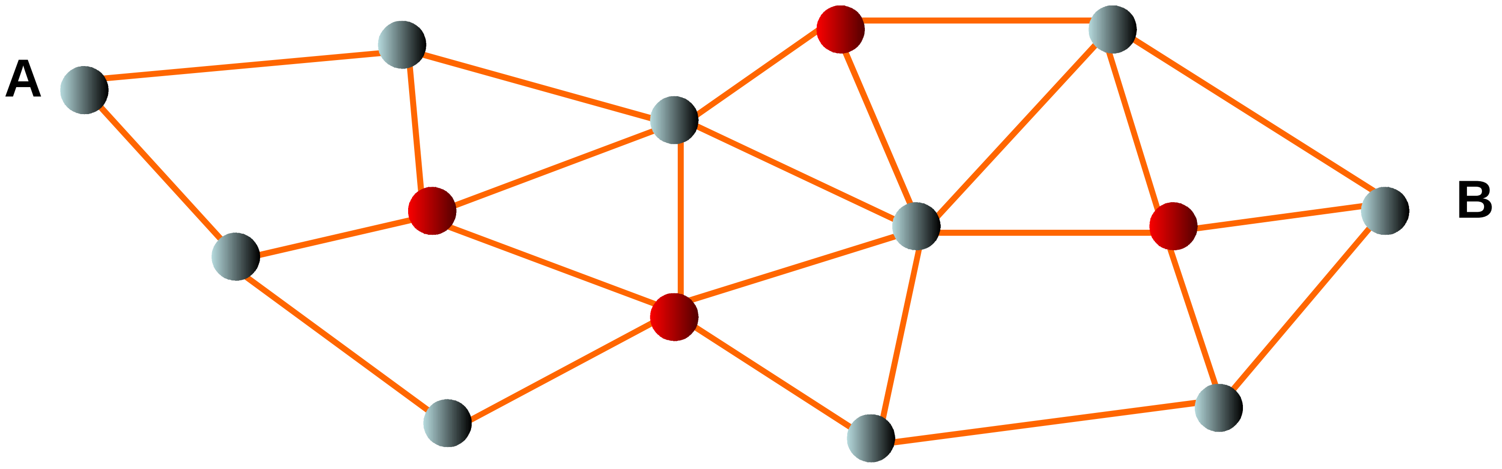
Scenario III: Network topology unknown but position of destination known

- Geometric spanners (wireless networks)
- Navigable graphs (small world) pioneered by Kleinberg 96

How to design self-stabilizing processes?

Adversarial Path Selection

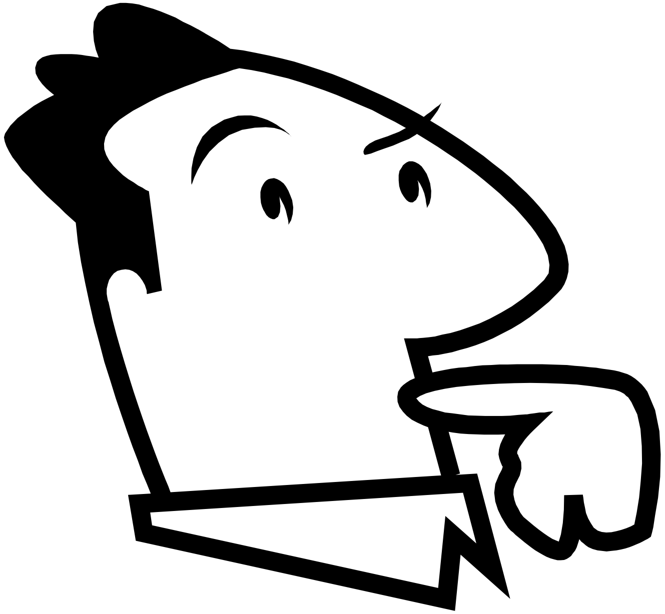
Scenario IV: Network topology unknown and position of destination unknown
! discovery via flooding



Open Problems

- Scheduling: non-uniform problems

- Path selection: many open problems left
- Collaborative learning approaches particularly interesting



Questions?