Distributed Algorithms

Rajmohan Rajaraman
Northeastern University, Boston

May 2012

Outline

- Randomization in distributed computing
  - Maximal independent set
- The local balancing paradigm
  - Multi-commodity flow and routing
- Information dissemination in dynamic networks
  - Network coding based approach

Randomization in Distributed Computing

- A number of distributed computing tasks need "local algorithms" to compute globally good solutions
  - Maximal independent set
  - Minimum dominating set
- Easy best-possible greedy algorithms
  - Inherently sequential
  - Place an ordering on the nodes
- How to compute in a distributed setting?
  - At the cost of a small factor, can work with an approximate order
  - Challenge: Break symmetry among competing nodes [Luby 86]

Maximal Independent Set

- Given an undirected graph G = (V,E), find a maximal set S such that no two vertices in S are adjacent in G
  - Easy to compute using a sequential algorithm
  - Difficult to make it distributed
- Model: In each step, each node can send an O(log(n)) bit message to each neighbor
- Theorem [Luby 86]: There exists a distributed MIS algorithm that completes in O(log(n)) rounds whp

Minimum Dominating Set

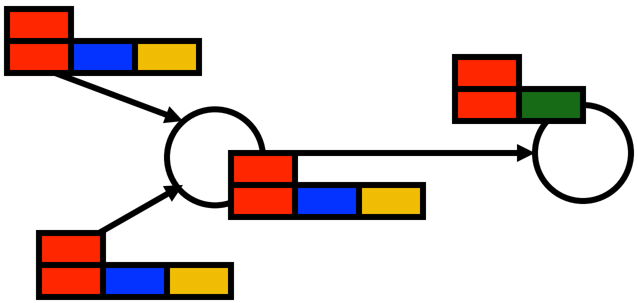
- Given an undirected graph G = (V,E), find the smallest S such that every vertex is either in S or adjacent to some vertex in S
  - Special case of set cover
  - A greedy algorithm yields an O(log(n)) approximation
  - Difficult to make it distributed
- Theorem [Jia-R-Suel 01]: There exists a distributed MDS algorithm that achieves an O(log(n)) approximation in $O(\log^2(n))$ rounds whp

Multi-commodity Flow and Routing

- Given an directed capacitated graph G = (V,E), with k source destination pairs (s_i,t_i), each with demand d_i
  - Goal: Find flow paths for the demands
  - Can be solved using LP and rounding, if necessary
  - Difficult to make it distributed
- Routing problem:
  - The demands represent rates
  - Packets arriving at sources according to rates need to be routed to corresponding destinations

Routing Through Local Balancing

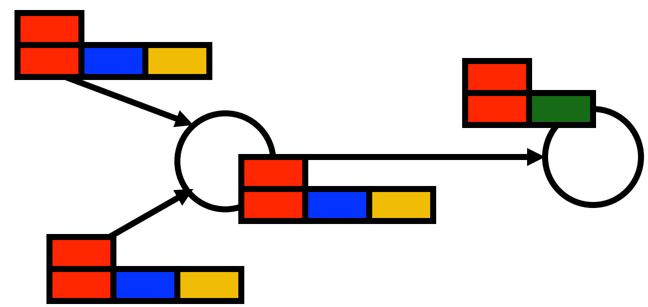
- Multi-commodity flow [Awerbuch-Leighton 94]
- Queue for each flow's packets at head and tail of each edge
- In each step:
  - New packets arrive at sources
  - Packet(s) transmitted along each edge using local balancing
  - Packets absorbed at destinations
  - Queues balanced at each node
- Local balancing through potentials
  - Packets sent along edge to maximize potential drop, subject to capacity
- Queues balanced at each node by simply distributing packets evenly



$\varphi_k(q) = \exp(\varepsilon q/(8Ld_k))$
L = longest path length
d_k= demand for flow k

Routing Through Local Balancing

- Edge capacities can be dynamically and adversarially changing
- If there exists a feasible flow that can route d_k flow for all k:
  - This routing algorithm will route (1-eps) d_k for all k
- Crux of the argument:
  - Destination is a sink and the source is constantly injecting new flow
  - Gradient in the direction of the sink
  - As long as feasible flow paths exist, there are paths with potential drop
- Follow-up work has looked at packet delays and multicast problems [Awerbuch-Brinkman-Scheideler 03, Jia-R-Scheideler 03]



$\varphi_k(q) = \exp(\varepsilon q/(8Ld_k))$
L = longest path length
d_k= demand for flow k

Information Dissemination

- Consider a highly dynamic network G in which each of n nodes has a message to share with all
- What is the best way to perform the all-to-all dissemination?
- Model: In each step, each node can broadcast an equivalent of O(1) messages to its neighbors
- Static network: Can complete in O(n) steps
- Dynamic network: Still open!

Network Coding

- A clever approach based on network coding seems very promising
- Idea: Instead of sending out individual messages, send random linear combinations
- If receiver receives enough such combinations, can reconstruct the original set of messages
- Theorem [Haeupler 11, Haeupler-Karger 11]: If message sizes are Omega(n) bits, then network-coding based gossip completes in O(n) steps whp
- Provably beats "flow-based" dissemination in certain models [Dutta-Pandurangan-R-Sun 12]

Take Away Messages

- Randomization:
  - Very effective technique for breaking symmetry
  - Can beat deterministic algorithms in certain distributed computing models
  - Also effective for thwarting adversaries
- Local balancing:
  - Useful for load balancing, flow, and routing problems
  - Potential function analysis effective to show
- Network coding:
  - Provably more effective than flow-based dissemination
  - Extent of separation still open