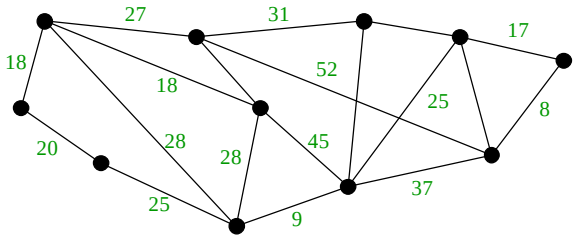
A Performance Comparison of Competitive On-line Routing and State-Dependent Routing

Matthew Andrews
Bell Labs

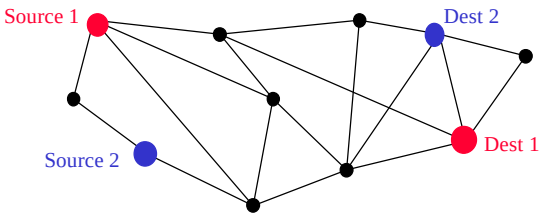
Joint with
Lisa Zhang - Bell Labs
Bill Aiello - Bellcore
Sandeep Bhatt - Bellcore
K. R. Krishnan - Bellcore

The Problem



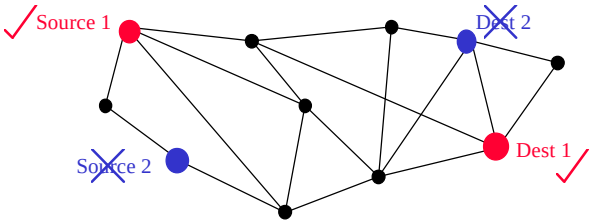
- Communication network
- Capacities on links

Call Requests



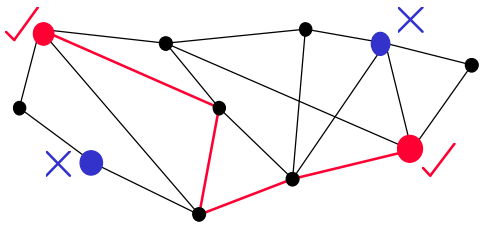
- Call request - source, destination, unit bandwidth
- Arrival times and holding times unknown
  - May have knowledge of distributions

Admission Control



- Decide which calls to accept

Routing

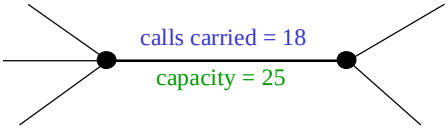


- For accepted request, select path from source to destination

Goal

- Maximize bandwidth routed
- Here ≡ maximize number of accepted requests

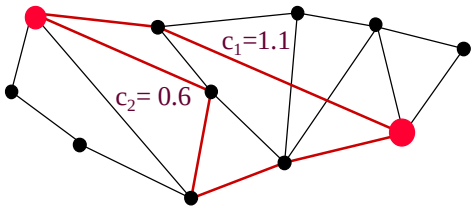
Algorithm Framework



- Define cost for each edge

  cost = f (edge utilization, edge capacity)

Algorithm Framework



- If there is path with cost < 1 then:
  - Accept request. Route on least-cost path.

  Else
  - Reject request.

Competitive On-line Routing (COL)

- Developed by Plotkin et al.

$$cost_e = \mu^{(n_e / c_e) - 1}$$

n_e = calls carried on edge e, c_e = capacity of edge e

- Does not assume traffic information

- Motivated by competitive analysis
  - Variant of COL is O(log n) competitive (n = #edges)

    Off-line alg accepts < O(log n) (# calls accepted by COL)

State-Dependent Routing (SDR)

- Developed by Krishnan and Ott

$$cost_e = \frac{B(c_e, \lambda_e)}{B(n_e, \lambda_e)}$$

- Assumes Poisson arrivals & exponential holding times
- B : Erlang-B formula. $B(c, \lambda) = \frac{\lambda^c}{c!} \Big/ \sum_{i=0}^{c} \frac{\lambda^i}{i!}$
- λ_e : Offered load to edge e under state-independent routing

- Motivated by Markov decision analysis
  - "Accept request if increase in # future rejections < 1"

SDR-ADAPT

- Calculating λ_e requires solution of large non-linear program
- Algorithm SDR-ADAPT attempts to learn λ_e

N_e = average occupancy of edge e
B_e = fraction of time that edge e is full
$\tilde{\lambda}_e = N_e / (1 - B_e)$

Previous Simulation Work

- COL: Compared favorably with GREEDY [GKPR]
  (GREEDY - accept request if there is available bandwidth)
- SDR: Compared favorably with :-
  Dynamic Non-Hierarchical Routing
  "Least-Loaded Rule"
- No direct comparison between COL and SDR

Aims of our simulations

- What is the benefit of having traffic information?
  (COL vs. SDR)

- What is the benefit of learning about the traffic?
  (COL vs. SDR-ADAPT)

Networks

- K5: Complete 5-node network. Uniform capacities and uniform requests between node pairs

- 16A: Complete 16-node networks with non-uniform
  16B: capacities and requests

- 38A: Dense 38-node network
  38B: Sparse 38-node network

- 15A: Dense 15-node networks
  15B:

- 66A: Sparse 66-node networks
  66B:

Results

- For complete 5-node network with uniform capacities and requests:-

  Algorithms have similar performance

Results

- For more complex networks and traffic patterns:-
  - SDR often noticeably better than COL, SDR-ADAPT
  - SDR-ADAPT compares favorably with COL in majority of cases

Choosing μ

- Recall that for COL:

$$cost_e = \mu^{(n_e / c_e) - 1}$$

- Question:
  - Is μ-value important to performance of COL?
- Answer:
  - Sometimes
- Open question:
  - Is there systematic method for choosing good μ?

Conclusion

- Knowledge of traffic does improve performance
- If traffic is unknown, then worthwhile to learn it
- If cost curve has "correct" shape, then good performance without knowledge of traffic