Kirk Pruhs

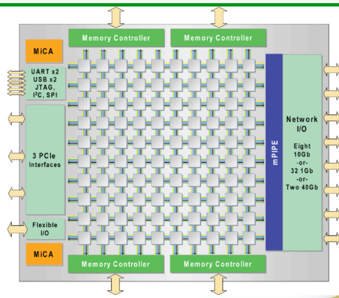
Hallucination Helps: Energy Efficient Virtual Circuit Routing

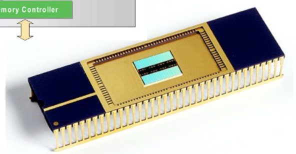
Workshop of
Flexible Network Design
July 2013

Covered Papers

- Past:
  - [WAOA2012] Anupam Gupta, Ravishankar Krishnaswamy, Kirk Pruhs: Online Primal-Dual for Non-linear Optimization with Applications to Speed Scaling. Workshop on Approximation and Online Algorithms, 2012: 173-186
  - [MedAlg2012] Nikhil Bansal, Anupam Gupta, Ravishankar Krishnaswamy, Viswanath Nagarajan, Kirk Pruhs, Cliff Stein: Multicast Routing for Energy Minimization Using Speed Scaling. Mediterranean Conference on Algorithms 2012: 37-51
- Present
  - Antonios Antoniadis, Sungjin Im, Ravishankar Krishnaswamy, Benjamin Moseley, Viswanath Nagarajan, Kirk Pruhs, Cliff Stein, Hallucination Helps: Energy Efficient Virtual Circuit Routing, submitted to the ACM-SIAM Symposium on Discrete Algorithms 2014.
- Future
  - Ravishankar Krishnaswamy, Viswanath Nagarajan, Kirk Pruhs, Cliff Stein, No title as of yet, hand written notes

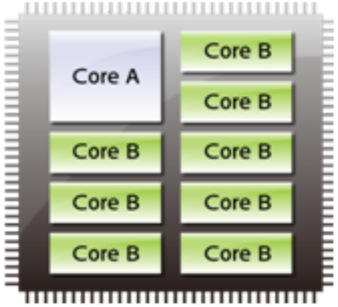
Green Computing Revolution





- Essentially all IT is being redesigned with energy efficiency as a first order resource

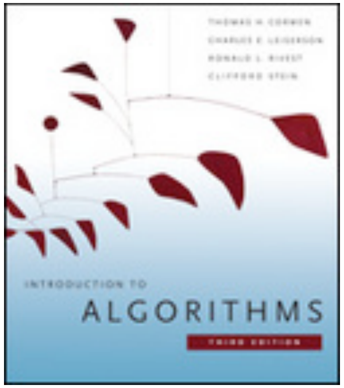
- Examples
  - Emergence of multicore chips
  - Emergence of Solid State Disks (SSD)
  - Speed Scalable processors and maybe disks
  - Power heterogeneous architectures
  - Google, Microsoft, etc. completely redesigning data centers and their management for energy efficiency
  - Cisco redesigning routers for energy efficiency

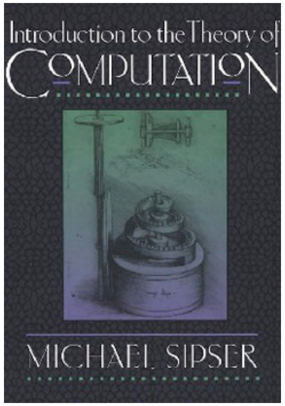


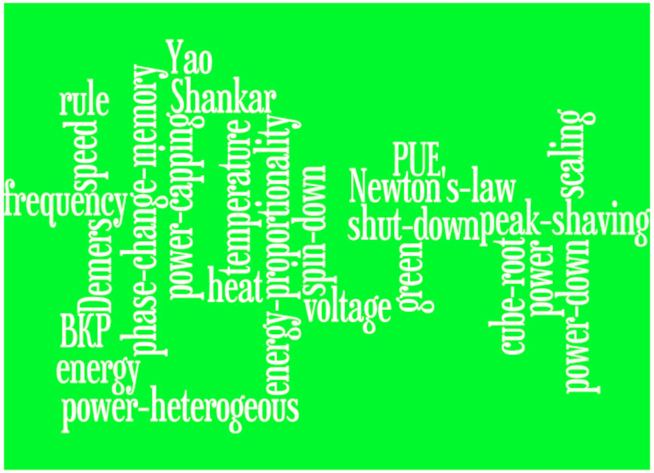
"What matters most to the computer designers at Google is not speed, but power, low power, because data centers can consume as much electricity as a city."--- Eric Schmidt, Former CEO Google

Long Term Goal Sales Pitch

- Build a theory of energy as a computational resource that allows software engineers to reason abstractly about power, energy and temperature as effectively as they can currently abstractly reason about time and space



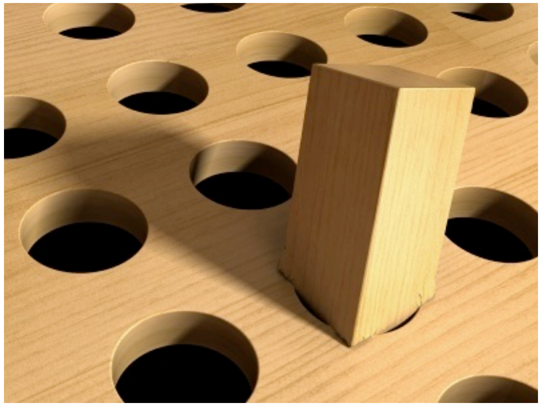




But …

- However, it seems that due to the fact that the physics of energy is quite different than that of time and space
  - e.g. there is no energy hierarchy theorem

- We need different models to study energy as a computational resource than we use for time and space



Current State of the Theory: Green Computing Algorithmics

- Algorithmic principles for managing resources with different energy characteristics using some mechanism to achieve an energy related objective

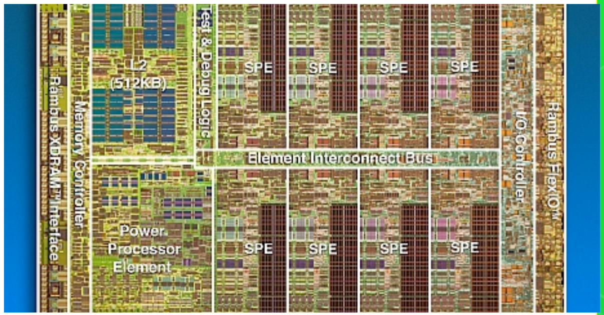
Energy Efficient Network Routing Research Program: This Talk

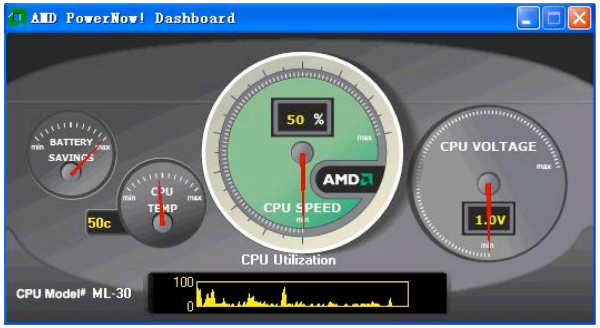
- Power Management Mechanism
  - Intra-device power heterogeneity
  - Inter-device power heterogeneity
    - Future work
  - Shutdown

- Network Routing Paradigm
  - Datagram packet routing
    - hard?
  - Virtual circuit routing

Dominant Energy Management Mechanisms

- 1. Power heterogeneity
  - Physics fact: Higher performance comes at a cost of energy efficiency
  - Management: Use higher performance mode/device when the increased energy per unit computation/communication gives sufficient performance returns
  - Heterogeneity can be intra-device or inter-device

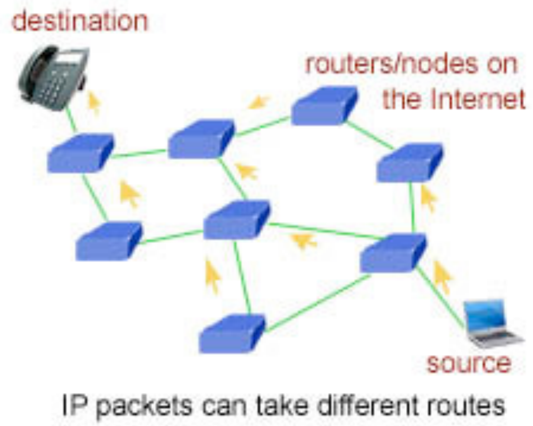




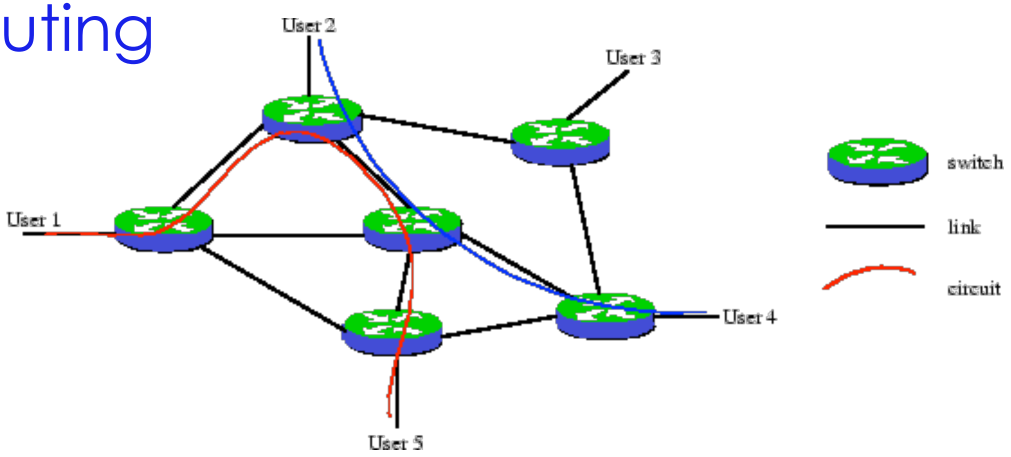
- 2. Shutdown

Network Routing Paradigms

- Datagram packet routing

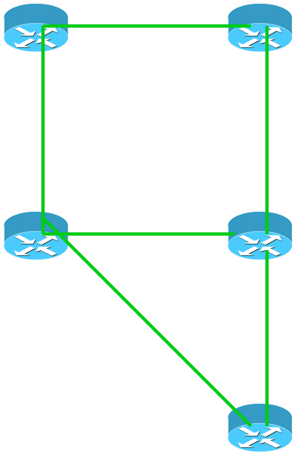


- Virtual circuit routing

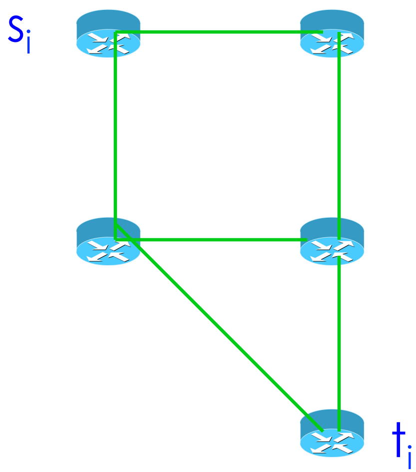


Virtual Circuit Model

- Network = undirected multigraph

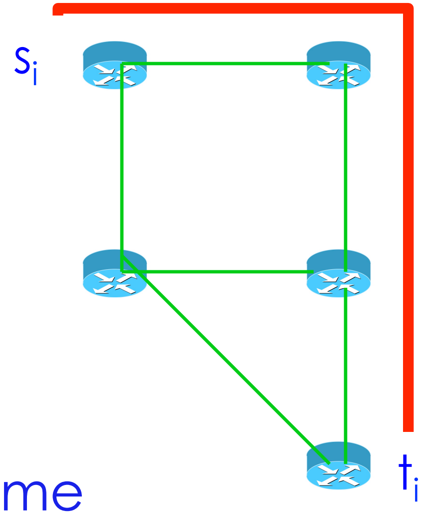


Virtual Circuit Model



- Network = undirected multigraph
- Input:
  - Requests for connections arrive over time
  - Request i consists of:
    - Source node s_i
    - Destination node t_i
    - Load (without great loss of generality assume unit loads for this talk)

Virtual Circuit Model

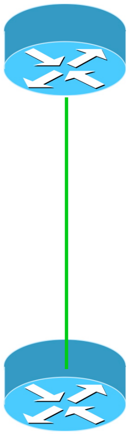


- Network = undirected multigraph
- Input:
  - Requests for connections arrive over time
  - Request i consists of:
    - Source node s_i
    - Destination node t_i
    - Load (without great loss of generality assume unit loads for this talk)
- Output: In response to request i, a (s_i, t_i) path must be specified

Standard Energy Model

- Power = static power + dynamic power
  - = σ + speed$^{\alpha}$
    - Speed in [0, ∞)
    - α in [1.1, 3]
  - = σ + load$^{\alpha}$

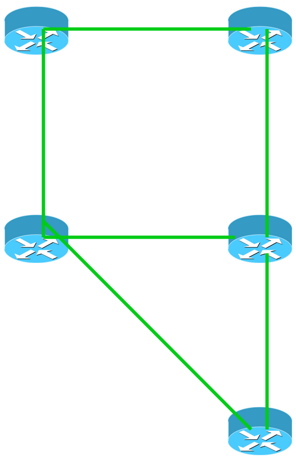
- A shutdown device uses no power



- For reasons of mathematical tractability, assume power management happens on edges
  - Later we'll say something about about the case where power management happens on nodes

Energy Efficient Routing Problem [AAZ2010]

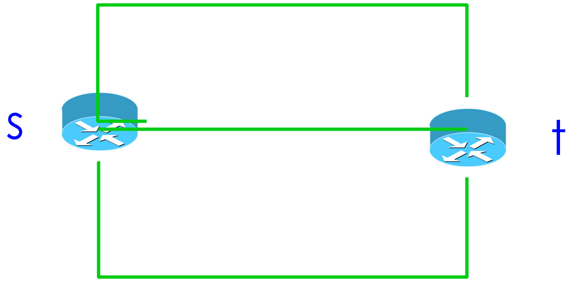
- Feasible Solution: A routing of all requests
- Objective: Minimize aggregate power over all edges
  - $= \Sigma_{\text{edges e powered on}} (\sigma + \text{load}(e)^{\alpha})$



$\sigma + \text{load}^{\alpha}$

Warm-up Problem 1:

- Assume static power $\sigma = 0$
- Assume all $s_i = s$
- Assume all $t_i = t$

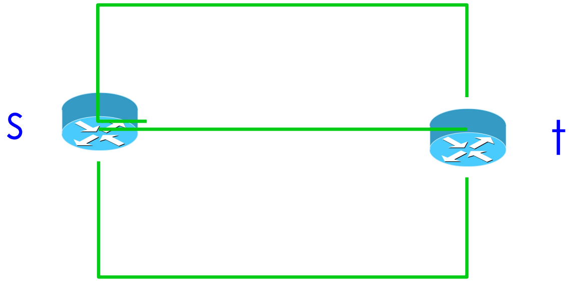


- Question: What is the optimal solution for this network?

$\sigma + \text{load}^{\alpha}$

Warm-up Problem 1:

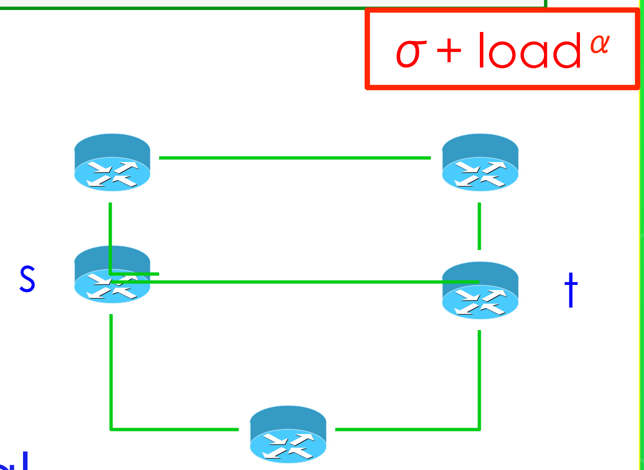
- Assume static power $\sigma = 0$
- Assume all $s_i = s$
- Assume all $t_i = t$



- Question: What is the optimal solution for this network?
- Answer: Put 1/3 of the paths on each edge

Warm-up Problem 2:

- Assume static power $\sigma = 0$
- Assume all $s_i = s$
- Assume all $t_i = t$

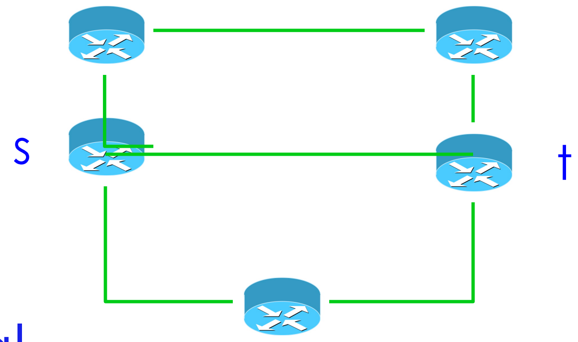


- Question: What is the optimal solution for this network?

Warm-up Problem 2:

$\sigma + \text{load}^{\alpha}$

- Assume static power $\sigma = 0$
- Assume all $s_i = s$
- Assume all $t_i = t$

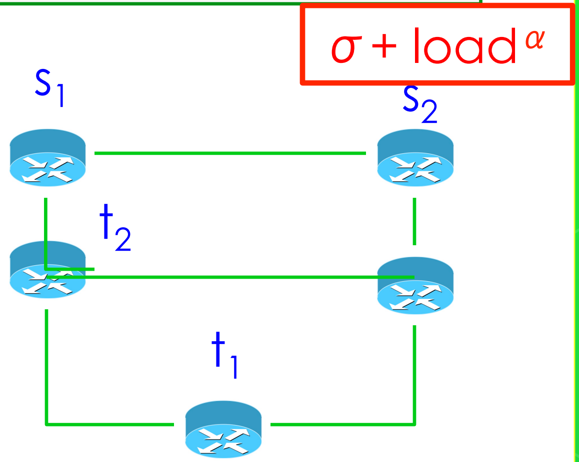


- Question: What is the optimal solution for this network?

- Answer: The aggregate power on each of the three s-t paths should be identical

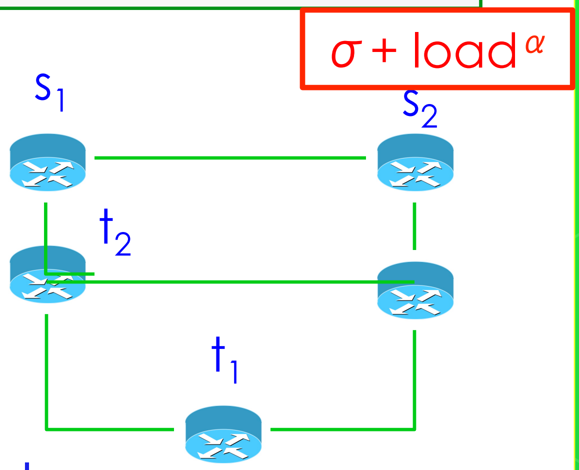
Warm-up Problem 3:

- Assume static power $\sigma = 0$
- Question: What is the obvious online algorithm?



Warm-up Problem 3:

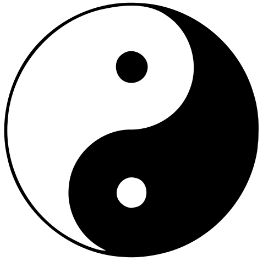
- Assume static power $\sigma = 0$



- Question: What is the obvious online algorithm?

- Answer: Greedy Water filling = route each request in such a way that the aggregate increase in power is minimized (shortest path computation)

- Theorem [AAFPW1997] Water filling is O(1)-competitive
- Proof[WAOA2012] : convex programming duality

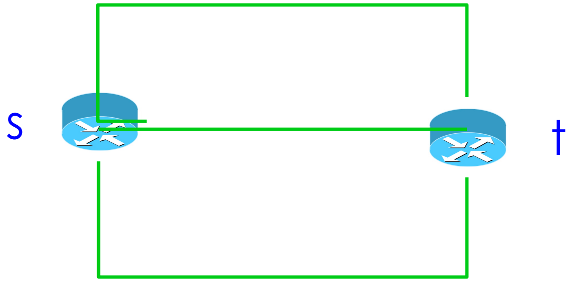


Warm-up Problem 4:

$\sigma + \text{load}^{\alpha}$

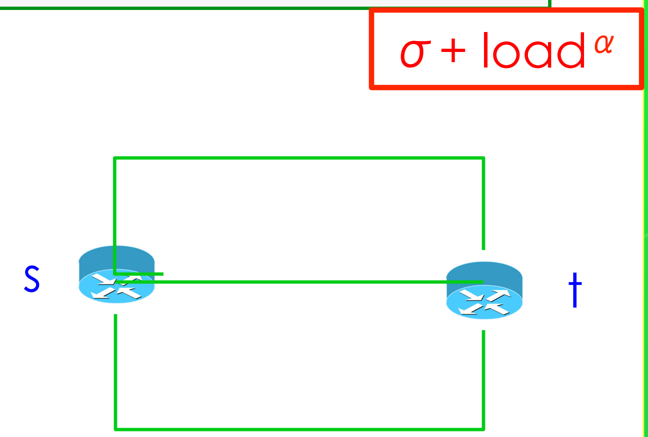
- Assume static power $\sigma = \infty$
- Assume all $s_i = s$
- Assume all $t_i = t$

- Question: What is the optimal solution for this network?



Warm-up Problem 4:

- Assume static power $\sigma = \infty$
- Assume all $s_i = s$
- Assume all $t_i = t$



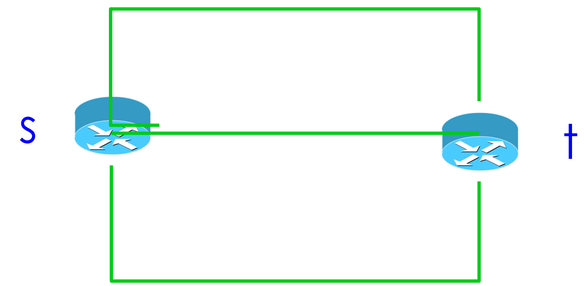
- Question: What is the optimal solution for this network?

- Answer: Use only 1 edge, and shutdown the rest

$\sigma + \text{load}^{\alpha}$

Warm-up Problem 5:

- k requests
- Many parallel edges
- Assume all $s_i = s$
- Assume all $t_i = t$

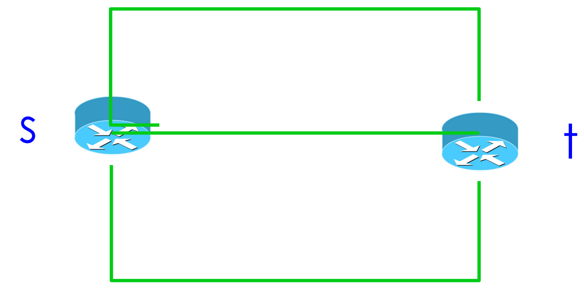


- Question: What is the optimal solution for this network?

$\sigma + \text{load}^{\alpha}$

Warm-up Problem 5:

- k requests
- Many parallel edges
- Assume all $s_i = s$
- Assume all $t_i = t$



- Question: What is the optimal solution for this network?
- Answer: Have q paths on each of k/q edges, and shutdown the rest of the edges
  - $q = \sigma^{1/\alpha}$ is load at which dynamic power = static power

Literature

- Theorem: Online greedy water filling algorithm is O(1)-competitive when static power is zero (or shutdown isn't possible)

- Theorem[AAZ2010]: There is a poly-time poly-log-approximation algorithm
  - Algorithm complicated and uses big hammers
  - The poly in poly-log-approximation is sufficiently large that it is not explicitly calculated
  - Hard to o(log $^{1/4}$ k) approximate using standard complexity theoretic assumptions

- Theorem[MedAlg2012]: On instances with a single source,
  - a poly-time O(1)-approximation using grouping and min-cost flow
  - an O(log $^{2\alpha+1}$ k)-competitive online algorithm

Results in Our SODA Submission

- Theorem: There is a poly-time $O(\log^{\alpha} k)$-approximation algorithm
  - Combination of simple combinatorial algorithms
  - Analysis follows directly from flow-cut gap for multi-commodity flow (the only hammer)
  - The poly in poly-log-approximation is small

- Theorem: An $\tilde{O}(\log^{3\alpha+1} k)$-competitive online algorithm
  - Natural minimal extension of offline algorithm
  - However, one part of the analysis is more involved than in offline case

Online/Offline Algorithm

- Power-on a Steiner forest to guarantee minimal connectivity
- Hallucination:
  - Sparsification: With probability Θ (log k)/q each request pair hallucinates its demand is q
  - Water filling algorithm is used to route this "hallucinated flow"
  - The "hallucinated" edges used to route hallucinated flow are powered on
- Water filling algorithm is used to route flow on the "powered on" edges

Offline Analysis

- Theorem: The algorithm is $O(\log^{\alpha} k)$-approximate
  - Proof: Both the static power and the dynamic power are $O(\log^{\alpha} k)$*Opt

Offline Analysis: Static Power

- Lemma: The static power for the Steiner forest edges is O(1)*Opt
- Lemma: The static power for the hallucinated edges is $O(\log^{\alpha} k)$ * Opt
  - Proof: The water filling that is used to route hallucinated flow is O(1)-competitive and sparsification doesn't increase the expected cost for Opt on any edge by more than $O(\log^{\alpha} k)$ factor on

Offline Analysis: Dynamic Power: via Congestion (1)

- Capacification: Each Steiner edge is given capacity q log k and each hallucinated edge is given capacity q

- Defn: Sparsity of a cut Q = capacity of edges in Q / demand across Q

- Lemma: The sparsity of every cut is Ω (log k)
  - Proof:
    - Tree edges have enough capacity if there is low demand across a cut.
    - Hallucinated edges have enough capacity if there is high demand across a cut.
    - Union bound

Offline Analysis: Dynamic Power: via Congestion (2)

- Corollary: There is a O(1)-congestion routing
  - Proof: flow-cut gap for multi-commodity flow

- Lemma: The dynamic power used by the algorithm is O($\log^{\alpha}$k) *Opt
  - Proof: There is a routing with O(1)-congestion, and hence flow O(q log k) on every powered-on edge, and the water filling algorithm that is used to route actual flow is O(1)-competitive

Recall Algorithm

- Power-on a Steiner forest to guarantee minimal connectivity
- Hallucination:
  - Sparsification: With probability Θ (log k)/q each request pair hallucinates its demand is q
  - Water filling algorithm is used to route this hallucinated flow
  - The “hallucinated” edges used to route hallucinated flow are powered on
- Water filling algorithm is used to route flow on the “powered on” edges

Online Analysis: Dynamic Power

- Lemma: The water filling algorithm is O(1)-approximate against any priority routing
  - Priority routing = each path only routes along edges powered on by the online algorithm by the time that request arrived
  - Proof: Same analysis as in [WAOA2012]

- Essentially all the technical difficulty: Need to prove that Opt can't greatly profit from powering on edges before online does
  - To mimic the analysis in the offline case, we need to show that there is a low congestion priority routing

Strategy to Show Low Congestion Priority Routing

Big

small

Optimal Dual Sparsest Priority Cut "ILP" Value

Argue this is big as in the offline case

Argue this "integrality gap" is small

Optimal Dual Sparsest Priority Cut LP Value

=

Optimal Primal Maximum Priority Flow LP Value

gap is obviously small

Optimal Primal Maximum Priority Flow ILP Value

Maximum Priority Multicommodity Flow LP

$$
\begin{aligned}
&\max \gamma && \text{(PriorityFlowLP)}\\
&\text{s.t.} \sum_{\boxed{p \in \mathcal{P}_i}} f(p) \geq \gamma && \forall i \in [k] \quad (2)\\
&\sum_{p | e \in p} f(p) \leq 1 && \forall e \in G(k) \quad (3)\\
&0 \leq f(p) \leq 1 && \forall i \in [k], \forall p \in \mathcal{P}_i \quad (4)
\end{aligned}
$$

- f(p) = flow routed on path p
- Priority component: $\mathcal{P}_i$ = priority (s_i, t_i) paths
- Objective: Fractionally route as large of a fraction of each unit demand as possible
- wlog capacities are 1 by duplicating edges

Dual LP: Sparsest Priority Cut

$$
\begin{aligned}
&\min \sum_{e \in G(k)} d_e && \text{(SparsestPriorityCutLP)} \\
&\text{s.t.} \quad \sum_{i=1}^{k} \eta_i \geq 1 && (5) \\
&\sum_{e \in p} d_e \geq \eta_i \qquad \boxed{\forall p \in \mathcal{P}_i} \forall i \in [k] && (6) \\
&d_e \geq 0 \qquad \forall e \in G(k) && (7) \\
&\eta_i \geq 0 \qquad \forall i \in [k] && (8)
\end{aligned}
$$

- Defn: (s_i, t_i) are priority cut by edges Q if removing Q makes them disconnected at time i
- Defn: Priority sparsity of cut Q = |Q|/ (number of requests priority separated by Q

- "ILP" = sparsest priority cut problem
  - d_e = 1/(number of priority cut requests) if e is in Q, and 0 otherwise
  - η_i = 1/(number of priority cut requests) if request i is cut, and 0 otherwise

Crux of Offline Analysis

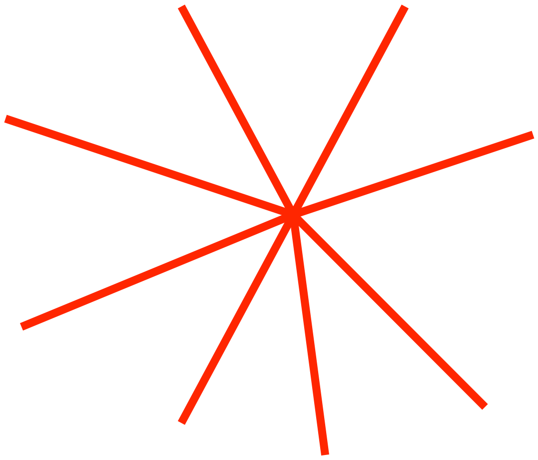
- Lemma: The priority sparsity of every cut is $\Omega(1)$
  - Proof: Slightly more involved than offline case
- Lemma: The priority cut "integrality gap" is $O(\log^2 k \log\log k)$
  - Comment: No good reason to think this is tight
  - Proof:
    - Geometric scaling to make η_i variables equal at a cost of a log k factor (reduction to multicut)
    - Use region growing approach to get "integral" d_e variables losing a $O(\log k \log\log k)$
- Corollary: There is a priority flow that has congestion $O(\log^2 k \log\log k)$
- Theorem: The online algorithm is $\tilde{O}(\log^{3\alpha+1} k)$-competitive

Say something about:

- Future paper
  - Ravishankar Krishnaswamy, Viswanath Nagarajan, Kirk Pruhs, Cliff Stein, No title as of yet, hand written notes
  - Assumes power management is on the vertices/router instead of the edges/links
    - In practice it seems more likely that power management will be more prevalent in routers than in links

What Power Management at Vertices is Different/Harder Mathematically/ Algorithmically

- Assume flow of 1 emanating from each leaf of a star-shaped Steiner tree, and you have to aggregate flows into groups of size q
  - There is a low edge congestion routing
  - There is not a low vertex congestion routing
- Upshot: The algorithm has to pick a Steiner forest so that the resulting vertex congestion is minimal



Steiner Tree

Main Result

- Theorem: There is a poly-time poly-log-approximate algorithm for energy efficient routing if the power management happens at the nodes

Research Agenda for This Fall

- Question 1: Can we obtain an analysis that doesn’t go via congestion?

- Question 2: Can we extend our results to the case that there is inter-device power heterogeneity?

Thanks to all my collaborators!
Thanks for listening!