Algorithmic aspects of communication networks	**Lecture 13**
OR 739 Spring 1997	**February 19, 1997**

Recall from last lecture that we have:

$$c(e) \geq 0$$
$$P_{s,t} \qquad \text{path from } s \text{ to } t \text{ used in solution}$$
$$f(e) = \text{number of paths on edge } e$$
$$\lambda = \max_e f(e)$$

Algorithm:

λ_0 at start of phase
$\ln a = 2\frac{1+\epsilon}{\lambda\epsilon}\ln(\frac{m}{\epsilon})$
$c(e) = a^{f(e)}$ update whenever we reroute
while phase is not over
For all s, t pairs
if $\sum_{e\in P_{s,t}} c(e) - dist_c(s,t) > \epsilon \sum_{e\in P_{s,t}} c(e) + \frac{\epsilon}{k}\lambda \sum_e c(e)$ then reroute,
where k = number of $s-t$ pairs

Theorem 1 *Suppose $\lambda \geq 8\epsilon^{-2}\ln(\frac{m}{\epsilon})$. If no rerouting happens in an iteration of the* While Loop *then $\lambda \leq \frac{1+\epsilon}{1-3\epsilon}\lambda^*$.*

Measure of Progress: $\Phi = \sum_e c(e) = \sum_e a^{f(e)}$

Theorem 2 *Rerouting a path decreases Φ by at least $\Delta\Phi \geq \frac{\log(\frac{m}{\epsilon})}{k}\Phi$.*

Proof:

Let P current path from s to t
Let Q shortest path from s to t
Let $A = P \setminus Q$ and $B = Q \setminus P$

$$\begin{aligned}\Phi &= \sum_{e\in A} c(e)(1-\frac{1}{a}) + \sum_{e\in B} c(e)(1-a) \\ &= (\sum_{e\in A} c(e) - \sum_{e\in B} c(e))(1-\frac{1}{a}) + \sum_{e\in B} c(e)(1-a+1-\frac{1}{a})\end{aligned}$$

$$= \quad (\sum_{e\in P} c(e) - \sum_{e\in Q} c(e))(1-\frac{1}{a}) + \sum_{e\in B} c(e)(1-a+1-\frac{1}{a})$$

$$\geq \quad (\epsilon \sum_{e\in P} c(e) + \frac{\epsilon}{k}\lambda \sum_{e} c(e))(1-\frac{1}{a}) + \sum_{e\in B} c(e) \underbrace{\overbrace{(1-a+1-\frac{1}{a})}^{\geq -2(\ln a)^2}}_{=0 \quad \text{if we cheat}} \qquad (*)$$

We have: $\ln a = 2\frac{1+\epsilon}{\lambda_0 \epsilon}\ln(\frac{m}{\epsilon}) \leq 2\frac{1+\epsilon}{\epsilon 8\epsilon^{-2}\ln(\frac{m}{\epsilon})}\ln(\frac{m}{\epsilon}) = \frac{1+\epsilon}{4}\epsilon$

"little cheat:" $a = e^{\ln a} \sim 1 + \ln a$

Thus, if we cheat a little:

$$\begin{aligned}
(*) \quad &\geq \quad \frac{\epsilon}{k}\lambda\Phi(1-\frac{1}{a}) \\
&\sim \quad \frac{\epsilon}{k}\lambda\Phi\ln a \\
&\geq \quad \frac{\epsilon}{k}\lambda\Phi 2\frac{1+\epsilon}{\epsilon\lambda_0}\ln(\frac{m}{\epsilon}) \\
&\underset{\text{phase didn't end}}{\geq} \quad \frac{\epsilon}{k}\Phi\frac{1+\epsilon}{\epsilon}\ln(\frac{m}{\epsilon})
\end{aligned}$$

If we did things correctly, we would have to use: $a \leq 1 + \ln a + (\ln a)^2$ and we would have needed the $\epsilon\sum_{e\in P} c(e)$ term in $(*)$ to balance $\sum_{e\in B} c(e)(1-a+1-\frac{1}{a})$.

□

Main Loop:

- start with any solution
- new phase starts if λ decreases by a factor of 2
- algorithm ends if either
  1. no rerouting
  2. $\lambda \leq 8\epsilon^{-2}\log(\frac{m}{\epsilon})$

Running Time:

- number of reroutings per phase:

  1 rerouting: $\Phi \longrightarrow \Phi(1-\frac{\ln(\frac{m}{\epsilon})}{k})$

  recall: $(1-\frac{1}{r})^r \sim \frac{1}{e} \approx \frac{1}{2.7}$

  So $\frac{k}{\ln(\frac{m}{\epsilon})}$ reroutings: $\Phi \longrightarrow \frac{\Phi}{2.7}$

  Also: $m \leq \Phi \leq m a^{\lambda_0}$

  $\Longrightarrow$ number of reroutings: $\frac{k}{\ln(\frac{m}{\epsilon})}\lambda_0 \ln a = \mathcal{O}(\epsilon^{-1}k)$

- number of phases:

  $\epsilon^{-2}\ln(\frac{m}{\epsilon}) \leq \lambda \leq k$

  $\Longrightarrow$ number of phases $\leq \log k$

Thus the total number of reroutings: $\mathcal{O}(\epsilon^{-1} k \log k)$

Improvements:

- $\mathcal{O}(k \log k)$ reroutings are enough (gradually decrease ϵ)
- our algorithm needs $\mathcal{O}(\epsilon^{-1} k^2 \log k)$ shortest path computations ($k\times$total number or reroutings $= k\mathcal{O}(\epsilon^{-1} k \log k)$, where the worst case is if each iteration reroutes one of the k paths).

  Radzik showed that by terminating if Φ does not decrease enough during a phase, $\mathcal{O}(k \log k)$ shortest paths computations are enough