An Approximation Algorithm for QoS Routing with Two Additive Constraints

Ronghui Hou, King-Shan Lui, Ka-Cheong Leung
Dept. of Electrical and Electronic Engineering
University of Hong Kong, Hong Kong

Fred Baker
Cisco Research Center
Outline

• Introduction
  – Precomputing the supported QoS
  – Sampling approximation method
• The approximation error produced by the sampling approximation method.
• Two-dimensional sampling mechanism.
• Simulation and Conclusion
Our focus

- Each link is associated with two additive metrics.
- Precomputing the supported QoS between two nodes.
- Different than the most current works
  - Given a request, finding a feasible path for this request.
  - Cannot support QoS in the Internet.
Routing with two constraints

- Finding a feasible path from E to D.
- Cannot find the best path from B to D.
  - (3,4); (5,3)

The aggregated topology

The original topology
Challenge!

- Precomputing the supported QoS between any two border nodes in a domain.
Non-dominated paths

- Six paths: (4,7), (5,6), (5,10), (7,5), (7,9), (8,4)
- Non-dominated paths: (4,7), (5,6), (7,5), (8,4)
- Representative points.
Our contributions

• NP-complete problem.

• **Propose** a new approximation method, two-dimensional sampling mechanism.

• **First study** to analyze the approximation error.

• **Formally prove** that our mechanism produces **smaller** approximation error than the existing algorithms.
QoS with integral constraints

The minimum delay with a cost constraint
Procedure for computing the supported QoS

- Hop-by-hop
- Polynomial-time
  - Depends on the metric of link
Sampling approximation method with real-number metrics

Uniform: $\delta, 2\delta, 3\delta, \ldots$ \hfill (Xin TON 2002)
Logarithmic: $1, 1+\delta, (1+\delta)^2 \ldots$ \hfill (Orda TON 2003)

**Uniform:** $\delta = 0.4$
Sampling approximation method

Approximation error?

![Diagram showing cost and delay with data points (4,7), (5,6), (7,5), (8,4).]
Cost-deviation

• Cost-deviation: $\Delta c = c - c_0$

Uniform: $\Delta c \leq h \delta$

Logarithmic: $\Delta c \leq ((1 + \delta)^h - 1)c_0$

Property: accumulated with $h$

Technique: induction
Setting the sampling parameter

- **Lemma**: Given a predefined $\varepsilon$, when $\delta \leq \varepsilon / 2N$, where $N$ is the number of nodes, the cost deviation of $c_0$ is less than $\varepsilon c_0$. 
Computing the error

![Diagram showing a 2D graph with axes labeled 'Delay' and 'Cost'. The graph includes points labeled as $p_1$, $p_2$, $p_3$, $p_4$, and $p_5$. The points are connected with lines indicating the relationship between delay and cost.]

The graph illustrates the relationship between delay and cost, with points $p_1$, $p_2$, $p_3$, $p_4$, and $p_5$ representing different scenarios. The lines connect these points, showing how delay changes with cost.
The approximation error

- **Approximation error**
  - Uniform sampling $\varepsilon \cdot UB$
  - Logarithmic sampling $\varepsilon \cdot (UB)^2$

- **Computational overhead**
  - Uniform sampling $O(N \cdot UB/\varepsilon)$.
  - Logarithmic sampling $O(N \cdot \log(UB)/\varepsilon)$.

The approximation error depends on UB
Two-dimensional sampling

- The main philosophy:
  - Bound both the cost-deviation and delay-deviation.
- Technique: sample both cost and delay
Computing the approximation error of our approach
Comparing errors

- Uniform sampling
  - $N\varepsilon^2$ (two-dimensional)
  - $\varepsilon N B$ (cost-sampling)
- Logarithmic sampling
  - $N\varepsilon^2 UB^2$ (two-dimensional)
  - $\varepsilon UB^2$ (cost-sampling)
  - $N\varepsilon$ should be less than 1.
Simulation Experiments

- Waxman-model based network topology.
  - 100 domains
  - 50-node or 100-node domains
- Two evaluation metrics
  - Region-deviation probability
    - The ratio of the approximation error to the actual feasible region.
  - The computational overhead
    - The average number of samples.
Simulation results

- Logarithmic sampling
  - 100-node domain
Conclusion and Future works

• Studied the problem of computing the supported QoS with two additive constraints.
• Extend to more than two additive constraints.
• Study the impact of the number of constraints on the performance of approximation algorithms.
Thank you!