Optimizing Cost and Performance for Multihoming

Lili Qiu
Microsoft Research

Joint Work with
D. K. Goldenberg, H. Xie, Y. R. Yang, Y. Zhang

HP Labs, Palo Alto
July 30, 2004
Motivation

Multihoming is a popular way to connect to Internet

Smart routing
Actively control how to distribute traffic among multiple providers

Potential benefits
- Improve performance
- Improve reliability
- Reduce cost
Related Work

Techniques for implementing multihoming
- BGP peering, DNS-based, NAT-based
- Complementary to our work

Performance evaluation
- Multihoming can potentially improve performance by 25% or more [Akella03]
- Multihoming can potentially achieve performance improvement similar to overlay routing [Akella04]
- Not clear how to achieve this in practice

Smart routing algorithms
- Commercial products [RouteScience, Internap, Proficient, ...]
- Hash-based load balancing [Cao01, Guo04]
Outline

Goal
- Design effective smart routing algorithms to achieve the potential benefit of multihoming

Questions
- How to assign traffic to multiple ISPs to optimize cost?
- How to assign traffic to multiple ISPs to optimize cost and performance?
- What are the global effects of smart routing?
**Network Model**

Network performance metric
- Latency (also an indicator for reliability)
- Easy to extend to alternative metrics

ISP charging models
- Cost = \( C(x) + C' \)
  - \( C' \): a fixed subscription cost
  - \( C \): a piece-wise linear non-decreasing function mapping \( x \) to cost
  - \( x \): charging volume
    - Percentile-based charging
    - Total volume based charging
Percentile-based Charging

Volume

Interval

95%*N

N
Why cost optimization?

- Previous work focus on performance only

Optimizing performance alone could result in high cost!
**Problem Specification: Cost Optimization**

**Inputs**
- \( K \): \# ISPs
- \( C_k \): cost function of ISP \( k \)
- \( q_k \): charging percentile of ISP \( k \)
- \( v(i) \): total traffic volume during interval \( i \)
- \( v(f,i) \): traffic volume of flow \( f \) during interval \( i \)

**Outputs**
- \( p_k \): charging volume of ISP \( k \) (e.g., 95-th percentile)
- \( T_k[i] \): traffic assigned to ISP \( k \) during interval \( i \)

**Goal:** find \( T_k[i] \) that minimizes \( \sum c_k(p_k) \)
- Offline fractional vs. online integral
An Extreme Case

- 95-th percentile-based charging
- 20 ISPs
- Optimal assignment
  - Each ISP serve 5% intervals
  - 95-th percentile traffic = 0
  - Cost = 0
Observations

• Each ISP can serve peak intervals for free

• Maximize the benefit of peak intervals
  - Burst ISP takes highest possible traffic
  - Maximize # peak intervals
    • Maximal # peak intervals is reached when ISPs’ peak intervals don’t overlap
Observations (Cont.)

• Let $V_0$ denote the sum of all ISPs’ charging volume
  - $V_0$ is boundary between peak vs. no peak
• Minimize cost $\iff$ minimize $V_0$
• $V_0 \geq 1-\sum(1-q_k)$ percentile traffic
Observation (Cont.)

4 ISPs, 95-th percentile charging

Volume

Non peak

Interval

80  95  100
Sketch of Our Algorithm

Determine charging volume for each ISP
- Compute $V_0$
- Find $p_k$ that minimizes $\sum_k c_k(p_k)$ subject to $\sum_k p_k = V_0$ using dynamic programming

Assign traffic given traffic volume
- Pseudo-capacity: the total amount of traffic assigned to each ISP
- Non-peak assignment: each ISP $k$ is assigned $\leq p_k$
- Peak assignment: pick ISP $k$ that bursted fewer than its allowance to burst and assign $p_k$ traffic to the other ISP $k'$
Dealing with Capacity Constraints

**Goal**
- Maximize # peak intervals subject to that multiple burst ISPs together can carry all the traffic

**Approach**
- \( f = 1 - V_0 \)

while (! IsPeakAssignable(f))
  reduce \( f \) by \( \Delta \)

Assign \( f \cdot I \) peak intervals s.t.
  each ISP \( k \) bursts \( \leq (1 - q_k) \cdot I \) intervals, and
  exists enough capacity for each peak interval
IsPeakAssignable

Determine if a given $f$ is assignable
- Cover all peak intervals
- No ISP bursts more than their peak interval quota
- Exists enough capacity in each peak interval

Our approach
- Let $g$ denote a set of ISPs when bursting together can carry any peak load traffic
- Let $t(g)$ denote the number of intervals $g$ burst
- $\sum t(g) \leq (1-q_k)*I$ for all $k$
- $\text{Max } \sum t(g) \geq f*I$
Online Cost Optimization

Traffic prediction
- Exponential weighted moving average (EWMA)
- Keep statistics for only large flows

Accommodate prediction errors
- Update V0 conservatively
- Adding some margin when computing charging volumes

Perform integral assignment
- Similar to bin packing
- Greedy heuristic
Optimizing Cost + Performance

Optimizing a metric that is a combination of cost and performance
- How to determine relative weights?

Our approach: optimize performance under a cost constraint
- Using the cost constraint to derive pseudo capacity for each ISP
- Using performance to arrive at actual assignment
Optimizing Performance Under Cost Constraints

**Offline algorithm**
- Mixed integer programming
- Using LP + rounding for large problems

**Online algorithm**
- Predict traffic and performance using EWMA
- Assign the flow to the best performing ISP among all ISPs with sufficient pseudo capacity
- Assignment order matters
  - In descending order of \(|\text{BestPerf}(f) - \text{worstPerf}(f)| \times v(f,i)|


Evaluation Methodology

- Abilene traces
  - Netflow data on Internet-2
  - RedHat, NASA/GSFC, NOAA Silver springs Lab, NSF, National library of medicine
  - Univ. of Wisconsin, Univ. of Oregon, UCLA, MIT
- Popular Web access logs

Delay traces
- NLANR traces: RTT measurement between pairs of 140 universities
- Map delay traces to hosts in traffic traces
Cost Functions

Simple cost functions (Feb. 2002 Blind RFP)
Cost Functions (Cont.)

Complex cost function for DS3

Cost

Usage (Mbps)
Cost Functions (Cont.)

Complex cost function for OC3
Baseline Algorithms

Round robin
- In each interval, assign traffic to a single ISP
- Rotate in a round robin fashion

Equal split
- In each interval, traffic is split equally among ISPs
- Similar to hash-based load balancing

Offline local fractional
- In each interval, minimize the total cost assuming cost is based on the traffic in the current interval

Dedicated links
- Flat rate and independent of usage
Cost under Complex Price Functions

Normalized cost

- Red Hat
- MIT
- UCLA
- Wisconsin
- Web Server

- GFA offline
- GIA online
- round robin
- equal
- LFA offline
Cost under Simple Price Functions

Normalized cost

- Red Hat
- MIT
- UCLA
- Wisconsin
- Web Server

GFA offline
GIA online
round robin
equal
LFA offline
Results for Varying # ISPs

Cost benefit increases with # ISPs.
Cost + Performance Evaluation

Optimizing performance alone often doubles the cost.
Our dual metric optimization achieves low cost and latency.
Global Effects of Smart Routing

• How well do the smart routing perform when traffic assignment affects link latency?

• How well do different smart routing users co-exist?

• How well do smart routing users co-exist with single-homed users?
Evaluation Methodology

Abilene traffic traces
Rocketfuel inter-domain topology
- 4 ASes, 170 nodes, 600 edges
- Assign propagation delay and OSPF weights according to Rocketfuel
- M/M/1 queue model (OC3 and OC12 for inter and intra domain links)

Routing
- User selects best performing ISP subject to cost constraints
- Inter-domain: shortest AS hop path
- Intra-domain: shortest OSPF path

Compute traffic equilibria using a variant of Frank Wolfe algorithm and relaxation [QYZS03]
Impact of self-interference is small.
Interaction Among Smart Routing Users

Smart routing users co-exist well with each other.
Smart Routing Users vs. Single-Homed Users

![Graph showing average latency (ms) over time interval]

- **reg 2 (reg 1)**
- **reg 2 (SR 1)**
- **reg 1 (reg 2)**
- **SR 1 (reg 2)**
Smart routing users co-exist well with single-homed users.
Summary

Design smart routing algorithms
- Offline/online cost optimization
- Offline/online cost + performance optimization

Show their effectiveness in optimizing cost and improving performance

Under traffic equilibria, smart routing improve performance without hurting other traffic
Future Work

• Conduct wide-area experiments
• Dynamics of interactions among different users
• Smart routing poses new challenges to ISPs
Thank you!

http://research.microsoft.com/
Results for Varying Time

GFA offline and GIA online consistently out-perform the alternatives.
Performance Benefits

Smart routing improves performance by 10% - 18%.
Computing Traffic Equilibrium of Selfish Routing

- **Computing traffic equilibrium of source routing traffic**
  - Use the linear approximation algorithm
    - A variant of the Frank-Wolfe algorithm, which is a gradient-based line search algorithm

- **Computing traffic equilibrium of overlay routing**
  - Construct a logical overlay network
  - Use Jacob's relaxation algorithm on top of Sheffi's diagonalization method for asymmetric logical networks
  - Use modified linear approximation algo. in symmetric case

- **Computing traffic equilibrium of multiple overlays**
  - Use a relaxation framework
    - Each overlay computes its best response by fixing the other overlays’ traffic
    - Merge the best response and the previous state using decreasing relaxation factors.