A BGP-based Mechanism for Lowest-Cost Routing

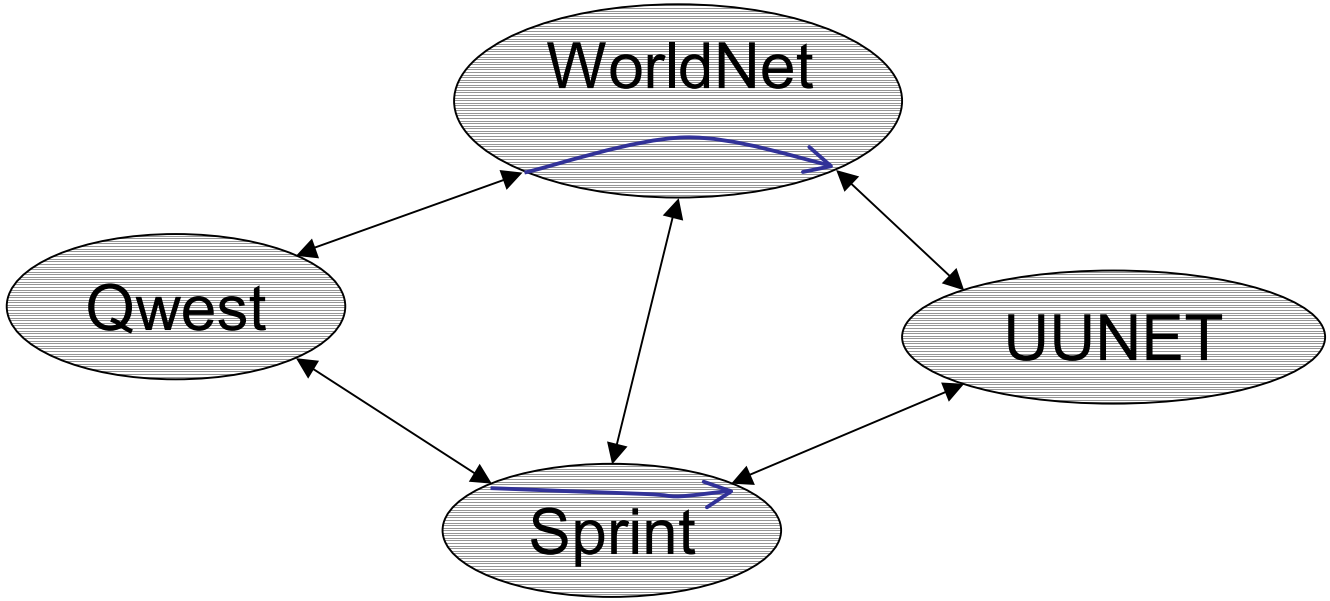
Rahul Sami
Yale University
http://www.cs.yale.edu/~sami

Joint work with:

Joan Feigenbaum	**Yale**
Christos Papadimitriou	**U. C. Berkeley**
Scott Shenker	**ICSI**

[To appear in Proceedings of 2002 ACM PODC]

Lowest-Cost Routing Mechanism-design Problem



Agents: Transit ASs

Inputs: Transit costs

Outputs: Routes, Payments

Outline

- VCG Mechanism for Lowest-Cost Routing
- “BGP-based” Computational Model
- DAM for Lowest-Cost Routing
- Open Questions

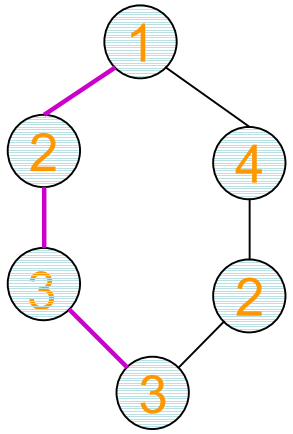
Problem Statement

Agents' types: Per-packet costs $\{c_k\}$

(Unknown) global parameter: Traffic matrix $[T_{ij}]$

Outputs: $\{route(i, j)\}$

Payments: $\{p^k\}$



Objectives:

- Lowest-cost paths (LCPs)
- Strategyproofness
- "BGP-based" distributed algorithm

Previous Work

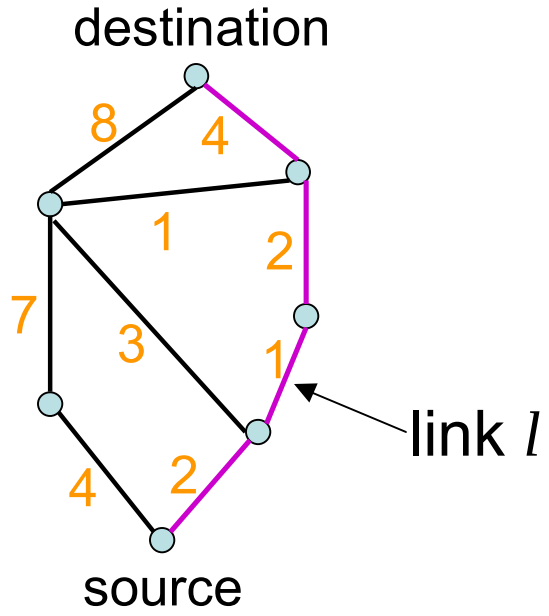
Nisan-Ronen, 1999

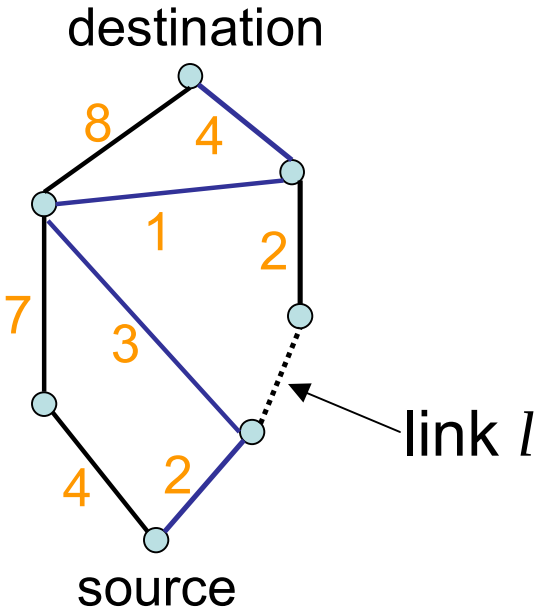
- Single (source, destination) pair
- Links are the strategic agents
- "Private type" of l is c_l
- (Centralized) strategyproof, polynomial-time mechanism
  - A Vickrey-Clarke-Groves (VCG) mechanism

Hershberger-Suri, 2001

- Compute m payments as quickly as 1

Lowest-Cost Path Mechanism [NR,HS]





Payment to link l on LCP =

Cost of l + Cost of lowest cost path without l
– Cost of LCP

(Similar payment function appears in *all* VCG mechanisms.)

Our Formulation vs. NR, HS

- *Nodes*, not links, are the strategic agents.
- *All (*source, destination) pairs
- *Distributed* "BGP-based" algorithm

Advantages:

- ✓ More realistic model
- ✓ Deployable via small changes to BGP

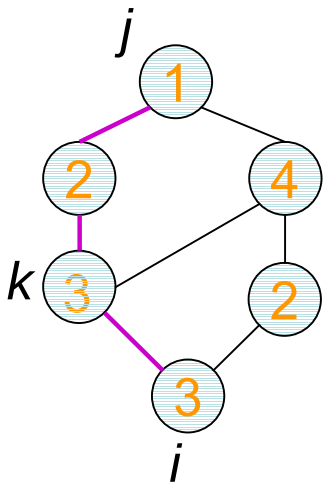
A Unique VCG Mechanism

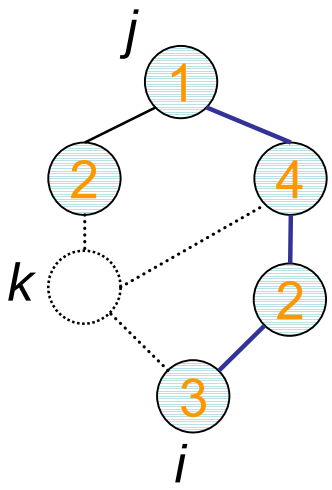
Theorem 1: Unique strategyproof mechanism that gives no payments to non-transit nodes.

Payment to transit node k is $p^k = \sum_{i,j} T_{ij}\, p^k_{ij}$, where

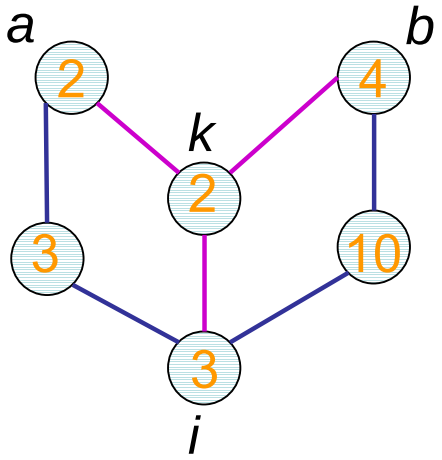
p^k_{ij} = c_k + Cost of LCP from i to j without using k

– Cost of LCP from i to j using k





Features of this Mechanism



- Cost c_k is independent of *i* and *j*, but price p_{ij}^k depends on *i* and *j*.

- Price p_{ij}^k is determined by cost of min-cost path from *i* to *j* not passing through *k* (min-cost "*k*-avoiding" path).

BGP-based Computational Model (1)

- Follow abstract BGP model of Griffin and Wilfong:
  Network is a graph with nodes corresponding to ASes and bidirectional links; intradomain-routing issues are ignored.

- Each AS has a routing table with LCPs to all other nodes:

Dest.	LCP				LCP cost
AS1	AS3	AS5	AS1		3
AS2	AS7	AS2			2

Entire paths are stored, not just next hop.

BGP-based Computational Model (2)

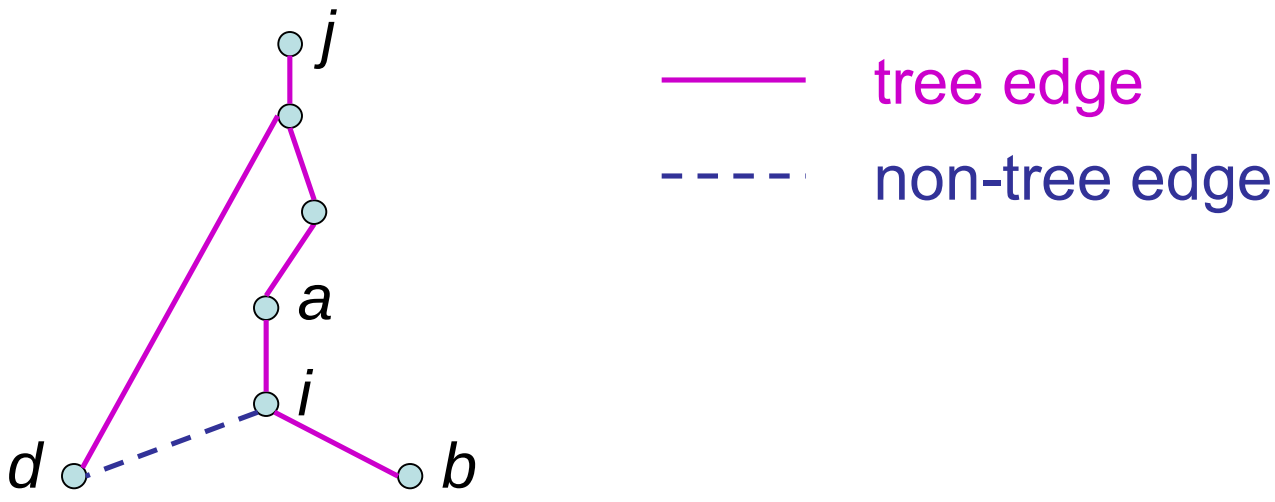
- An AS "advertises" its routes to its neighbors in the AS graph, whenever its routing table changes.
- The computation of a single node is an infinite sequence of stages:

- Complexity measures:
  - Number of stages required for convergence
  - Total communication

Towards Distributed Price Computation

p_{ij}^k = c_k + *Cost of min-cost k-avoiding path from i to j*
– *Cost of LCP from i to j*

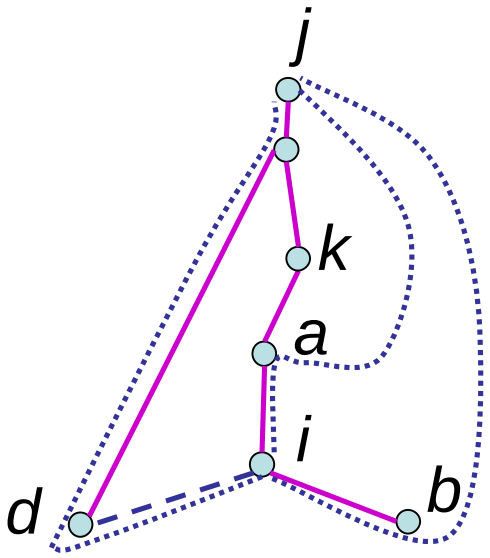
- LCPs to destination *j* form a tree



- Use data from *i*'s neighbors *a,b,d* to compute p_{ij}^k .

Constructing k-avoiding Paths

Three possible cases for min-cost k-avoiding path from i to j:



i's neighbor on the path is

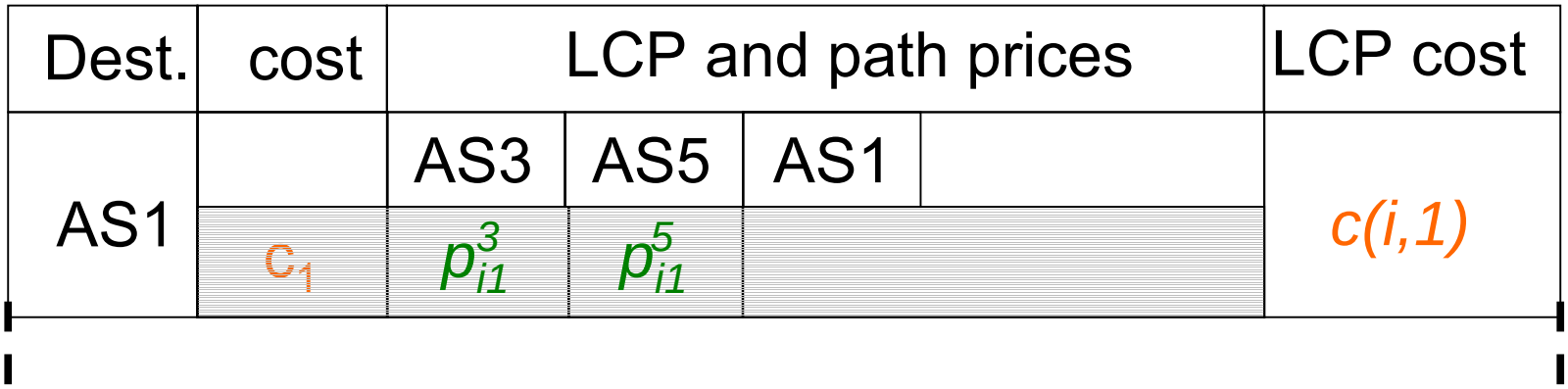
(*a*) parent
(*b*) child
(*d*) unrelated

In each case, a relation to neighbor's LCP or price, *e.g.*,

(b) $p^k_{ij} = p^k_{bj} + c_b + c_i$

✓ p^k_{ij} is the minimum of these values.

A "BGP-based" Algorithm



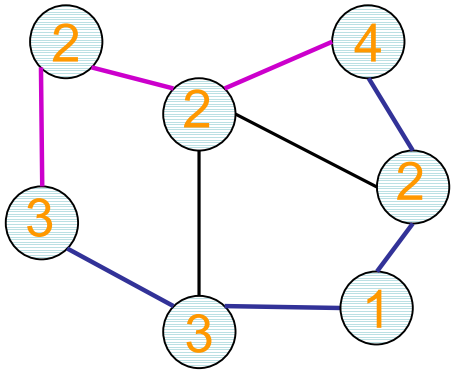
- LCPs are computed and advertised to neighbors.
- Initially, all prices are set to ∞.
- Each node repeats:
  - Receive LCP costs and path prices from neighbors.
  - Recompute path prices.
  - Advertise changed prices to neighbors.

Final state: Node *i* has accurate p^k_{ij} values.

Performance of Algorithm

d = maximum length of LCP

d' = maximum length of min-cost k-avoiding path



Theorem 2:

Our algorithm computes the VCG prices correctly, uses routing tables of size *O(nd)* (a constant factor increase over BGP), and converges in at most *(d + d')* stages (worst-case additive penalty of d' stages over the BGP convergence time).

Open Question: Strategy in Computation

- *Mechanism* is strategyproof : ASes have no incentive to lie about c_k's.

- However, payments are *computed* by the strategic agents themselves.
  How do we reconcile the strategic model with the computational model?

- First approach : Digital Signatures
  [Mitchell, Sami, Talwar, Teague]

Is there a way to do this without a PKI?

Open Question: Overcharging

- In the worst case, path price can be arbitrarily higher than path cost [Archer&Tardos, 2002].
- This is a general problem wiith VCG mechanisms.
- Statistics from a real AS graph, with unit costs:

  Mean node price : *1.44*
  Maximum node price: *9*
  90% of prices were *1* or *2*

  Overcharging is not a major problem!

How do VCG prices interact with AS-graph *formation*?