Distributed Algorithmic Mechanism Design for Network Problems

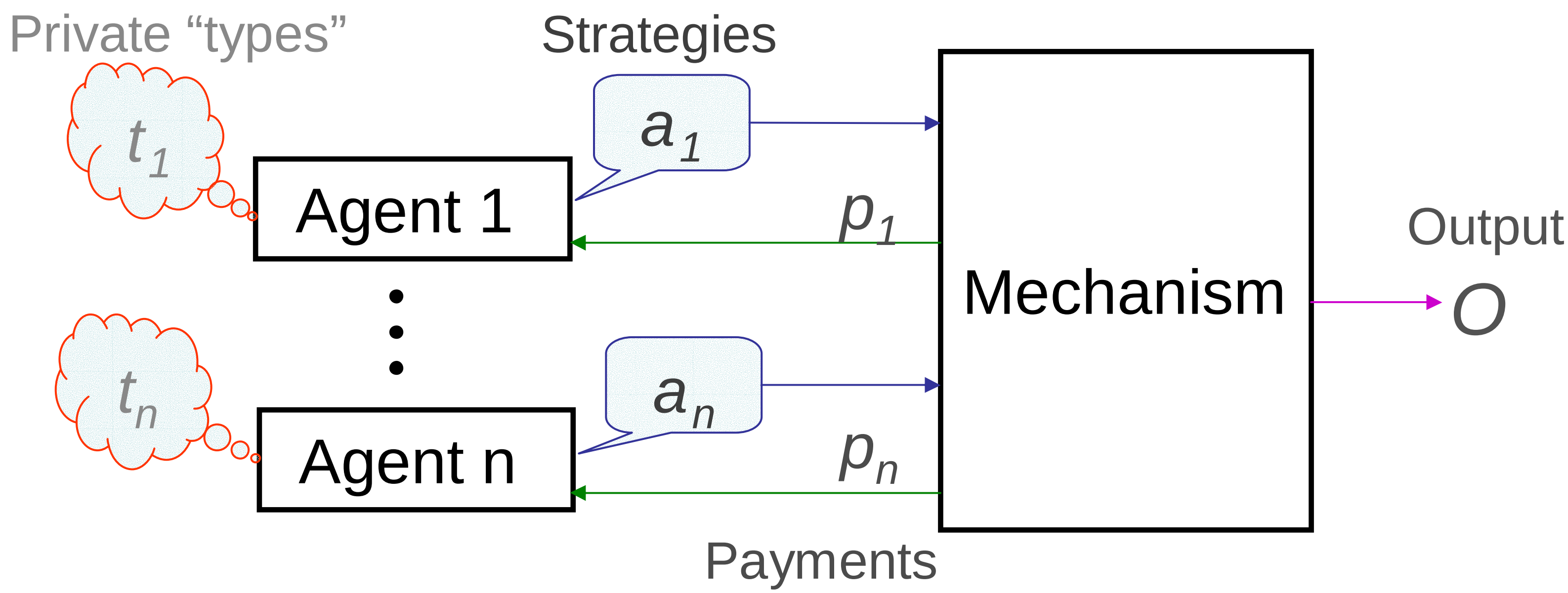
Rahul Sami

Ph.D. Dissertation Defense

Advisor: Joan Feigenbaum

Committee: Ravindran Kannan
Arvind Krishnamurthy
Scott Shenker (ICSI & Berkeley)

Mechanism Design Framework



Agent *i* chooses strategy a^i to maximize her welfare.

Strategyproof Mechanisms

Strategyproof mechanism:

Regardless of what other agents do, each agent *i* maximizes her welfare by revealing her true private type information t^i.

Group-strategyproof mechanism:

Even if agents can collude, no group of agents can benefit from not revealing their true types– regardless of the strategies of non-members, any strategy that makes one group member strictly better off would make at least one group member strictly worse off.

Algorithmic Mechanism Design

Algorithmic Mechanism Design (Nisan-Ronen ’01):

- Introduced computational efficiency into mechanism-design framework.
- Polynomial-time computable *O( )* and $p_i( )$
- Centralized model of computation

Distributed AMD (Feigenbaum-Papadimitriou-Shenker ’01)

- Computation is distributed across the network.
- “Good network complexity”:
  - Polynomial-time local computation
  - Modest in total number of messages, messages per link, and maximum message size

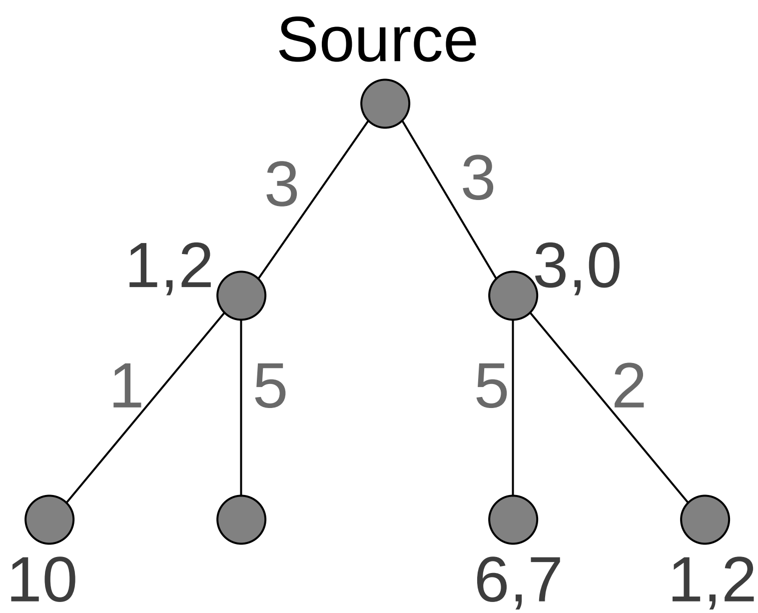
Thesis Statement

"The distributed-computing context can have a major impact on the feasibility of a mechanism."

Thesis supported with results on

- Multicast Cost-Sharing
- Interdomain Routing

Multicast Cost Sharing Mechanism-design Problem



Users' utilities

Link costs

<u>Receiver Set</u>

Which users receive the multicast?

<u>Cost Shares</u>

How much does each receiver pay?

Earlier Work on Multicast Cost Sharing

- Moulin and Shenker ‘01

  Two mechanisms:
  - Efficient: marginal cost (MC)
  - Budget-balanced: Shapley value (SH)

- Feigenbaum, Papadimitriou and Shenker ‘01
  - Distributed algorithm for MC with good network complexity
  - Restricted lower bound on network complexity of SH

Multicast Cost-Sharing Results (1)

[Feigenbaum-Krishnamurthy-Sami-Shenker.
To appear in *Theoretical Computer Science.*
Extended abstract appeared in *FSTTCS ’02*.]

- Any deterministic or randomized algorithm for any exactly budget-balanced group-strategyproof mechanism must send $\Omega(n)$ bits on a single link in the worst case, where n is the number of users.

- Any algorithm to compute a κ-approximately budget-balanced group-strategyproof mechanism, for $\kappa < \sqrt{2}$, must send $\Omega(\log n / \log \kappa)$ bits on a single link in the worst case.

- There is no mechanism that is strategyproof, approximately budget-balanced, and approximately efficient.

Multicast Cost-Sharing Results (2)

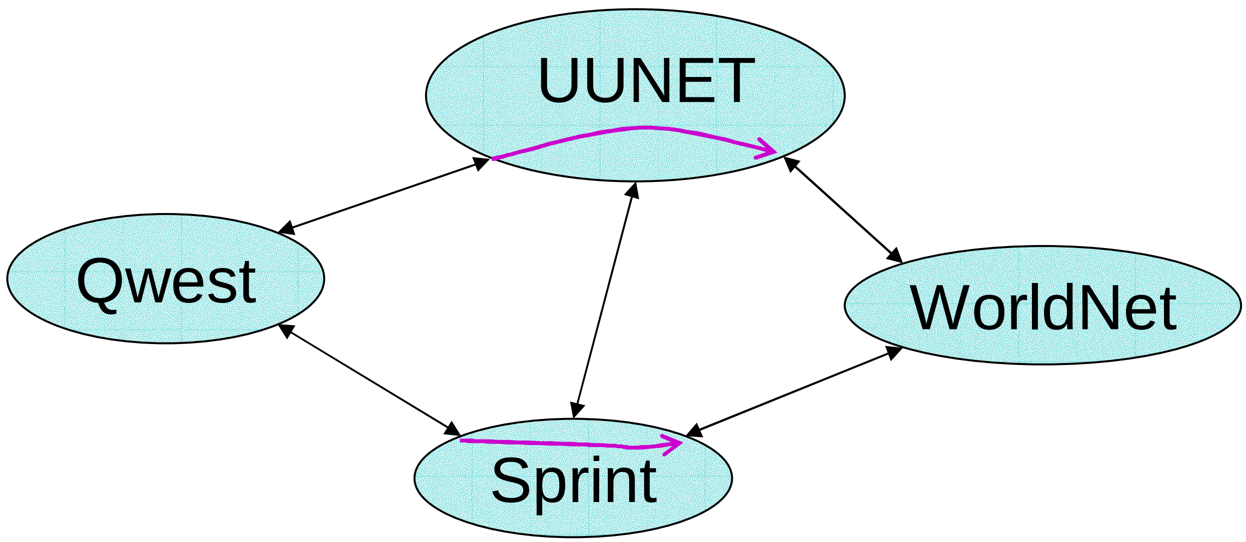
[Archer-Feigenbaum-Krishnamurthy-Sami-Shenker, to appear in *Games and Economic Behavior* .]

- Constructed a mechanism SSF (Scaled Step-Function) that is
  - Group-strategyproof
  - Network complexity:

    $O\left(\frac{\log n}{\log \kappa}\right)$ values communicated on a link
  - Budget-deficit bound: $1 \geq \frac{\text{Revenue}}{\text{Cost}} \geq \frac{1}{\kappa^h}$
  - Efficiency bound: Eff.(SSF) ≥ Eff.(SH) – $(\kappa^h - 1)U$

$\kappa > 1$ controls trade-off of computational goals *vs.* economic goals.

Mechanism Design for Interdomain Routing



Agents: Autonomous Systems (ASs)

Two formulations:

- Lowest-cost Routing
- General Policy Routing

Routes currently computed with the Border Gateway Protocol (BGP).

Lowest-Cost Routing

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

Outputs: {*route(i, j)*}

Payments: $\{p_k\}$

Objectives:

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

Prior work:

[Nisan-Ronen '01]:

- Links (not nodes) are strategic agents.
- Strategyproof mechanism for single source-destination pair.
- Polynomial-time centralized computation.

[Hershberger-Suri '01]:

- Faster payment computation for Nisan-Ronen mechanism.

Lowest-Cost Routing Results

[Feigenbaum-Papadimitriou-Sami-Shenker, PODC ‘02]:

- For a biconnected network, if LCP routes are always chosen, there is a unique strategyproof mechanism that gives no payment to nodes that carry no transit traffic.

- Prices required by this mechanism can be computed by a “BGP-based” algorithm with routing tables of size $O(nd)$ (a constant factor increase over BGP) and convergence time at most $max(d\ ,d')$ stages of computation, where

$$d = max_{i,j} \text{ (\# hops in LCP from } i \text{ to } j\text{)}$$

$$d' = max_{i,j,k} \text{ (\# hops in lowest-cost } i\text{-}j \text{ path not using } k \text{)}$$

General Policy-Routing Mechanism Design Problem

- Per-packet c_k is an unrealistic cost model.
- AS route preferences are influenced by reliability, customer-provider relationships, *etc*.

General Policy Routing:

- For all *i,j* , AS *i* assigns a value $u_i(P_{ij})$ to each potential route P_{ij}.
- Mechanism-Design Goals:
  - Maximize $W = \sum_{i,j} u_i(P_{ij})$.
  - For each destination *j*, $\{P_{ij}\}$ forms a tree.
  - Strategyproofness
  - BGP-based distributed algorithm

Policy-Routing Results

[Feigenbaum-Griffin-Ramachandran-Sami-Shenker]:

- Arbitrary valuation functions $u_i(P_{ij})$
  - NP-hard to find routing tree that maximizes *W*.
  - NP-hard even to approximate maximum *W* within $O(n^{1/4 - \varepsilon})$.

- Next-hop preferences: $u_i(P_{ij})$ depends only on next-hop AS *a.*
  - Captures preferences due to customer/provider/peer agreements.
  - There is a unique strategyproof mechanism that maximizes *W* and does not pay non-transit nodes.
  - This mechanism is polynomial-time computable (in a centralized model).
  - BGP-based distributed algorithm is infeasible:
    - May take $\Omega(n)$ stages to converge.
    - May require $\Omega(n)$ messages for every preference change.

Other Research done at Yale

Theoretical Computer Science

[Feigenbaum-Fortnow-Pennock-Sami, EC '03]:
Computation in a Distributed Information Market

[Batu-Ergun-Kilian-Magen-Rubinfeld-Raskhodnikova-Sami, STOC '03]:
A Sublinear Algorithm for Weakly Approximating Edit Distance

Computer Architecture

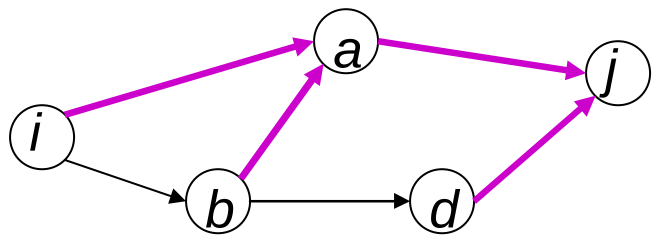
[Henry-Kuszmaul-Loh-Sami, ISCA '00]:
Circuits for Wide-Window Superscalar Processors

[Henry-Loh-Sami, ISHPC '02]:
Speculative Clustered Caches for Clustered Processors

[Loh-Sami-Friendly, WDDD '02]:
Memory Bypassing: Not Worth the Effort

General Policy-Routing Problem Statement

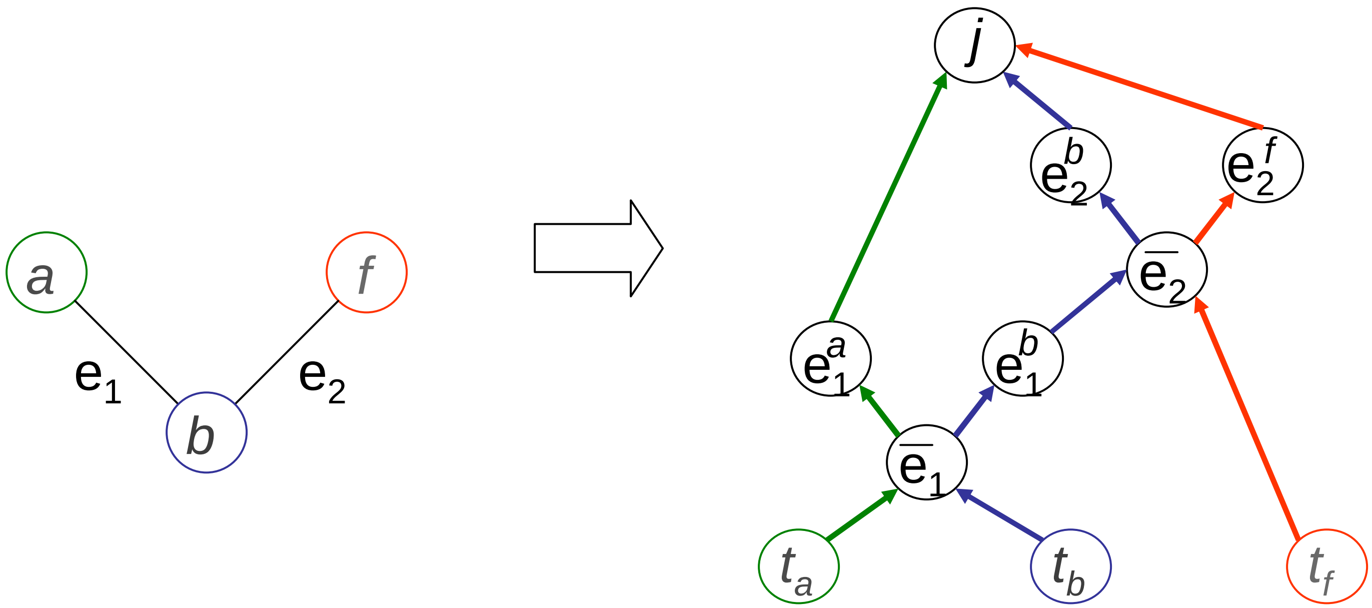
- Consider each destination j separately.
- Each AS i assigns a value $u_i(P_{ij})$ to each potential route P_{ij}.



- Mechanism-design goals:
  - Maximize $W = \sum_i u_i(P_{ij})$.
  - For each destination j, $\{P_{ij}\}$ forms a tree.
  - Strategyproofness
  - BGP-based distributed algorithm

NP-Hardness with Arbitrary Valuations

- Approximability-preserving reduction from Independent-set problem:



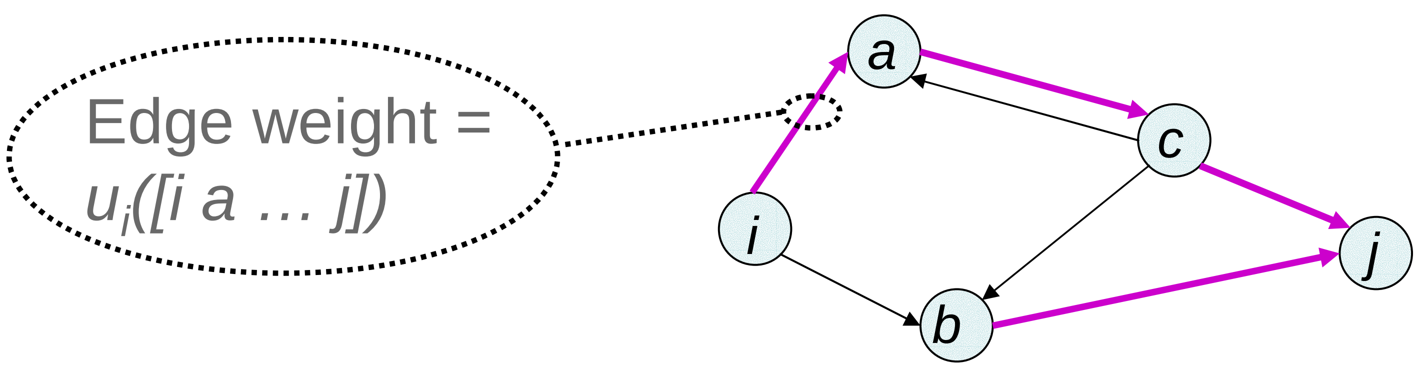
Paths from terminals t_a, t_b, t_f have valuation 1, all other paths 0.

- NP-hard to compute maximum W exactly.
- NP-hard to compute $O(n^{1/4 - \varepsilon})$ approximation to maximum W.

Next-Hop Preferences

- $u_i(P_{ij})$ depends only on next-hop AS *a.*
- Captures preferences due to customer/provider/peer agreements.

For each destination *j* , optimal routing tree is a Maximum-weight Directed Spanning Tree (MDST):



Strategyproof Mechanism

Let

T^* = Maximum weight directed spanning tree (MDST) in G

T^{-i} = MDST in $G - \{i\}$

- For 2-connected networks, there is a unique strategyproof mechanism that always picks a welfare-maximizing routing tree and never pays non-transit nodes. The payments required for this mechanism are

$$p^i = W(T^*) - u_i(T^*) - W(T^{-i})$$

- Routes and payments can be computed in polynomial time (in a centralized computational model.)

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 and route aggregation are ignored.

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

Dest.	AS Path				Cost/Pref.
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

Proving Hardness for “BGP-based” Routing Algorithms

Requirements for routing to any destination j :

[P1] Each AS uses O(*l*) space for a route of length *l*.

[P2] The routes converge in O(*log n*) stages.

[P3] Most changes should not have to be broadcast to most nodes: *i.e.*, there are o(*n*) nodes that can trigger Ω(*n*) updates when they fail or change valuation.

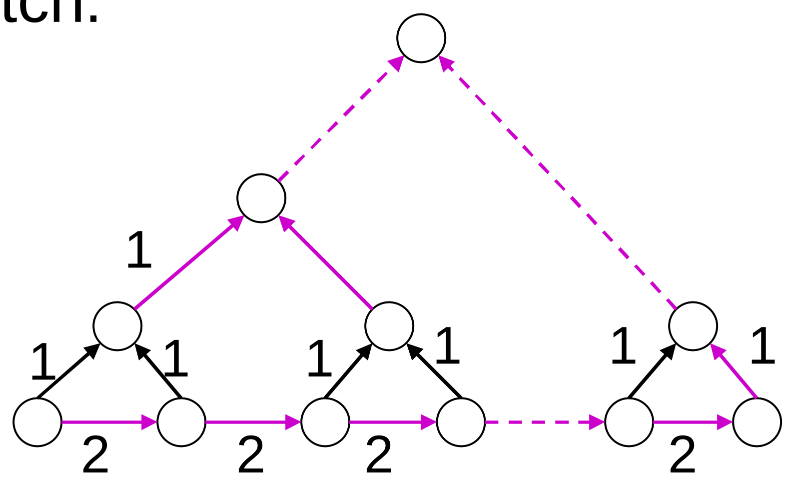
Hardness results must hold:

- For “Internet-like” graphs:
  - *O(1)* average degree
  - *O(log n)* diameter
- For an open set of preference values in a small range.

Long Convergence Time of MDST

Theorem 1: An algorithm for MDST mechanism cannot satisfy property [P2].

Proof sketch:



- Network has diameter O(*log n*), but MDST has a path of length Ω(*n*).
- BGP routes in a hop-by-hop manner
  ⇒ Ω(n) stages required for convergence.

Extensive Dynamic Communication

Theorem 2: A distributed algorithm for the MDST mechanism cannot satisfy property [P3].

Proof outline:

(i) Construct a network and valuations such that for $\Omega(n)$ nodes i, T^{-i} is disjoint from the MDST T^*.

(ii) A change in the valuation of any node a may change

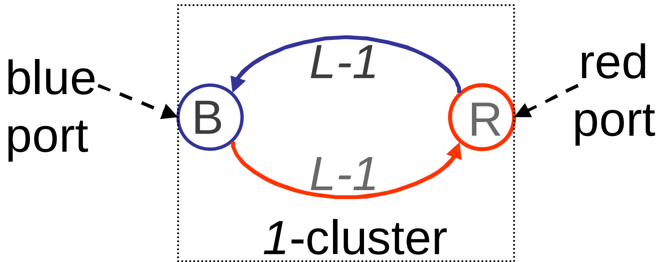
$$p_i = W(T^*) - u_i(T^*) - W(T^{-i}).$$

(iii) Node i (or whichever node stores p_i) must receive an update when this change happens.

$\Rightarrow$ $\Omega(n)$ nodes can each trigger $\Omega(n)$ update messages.

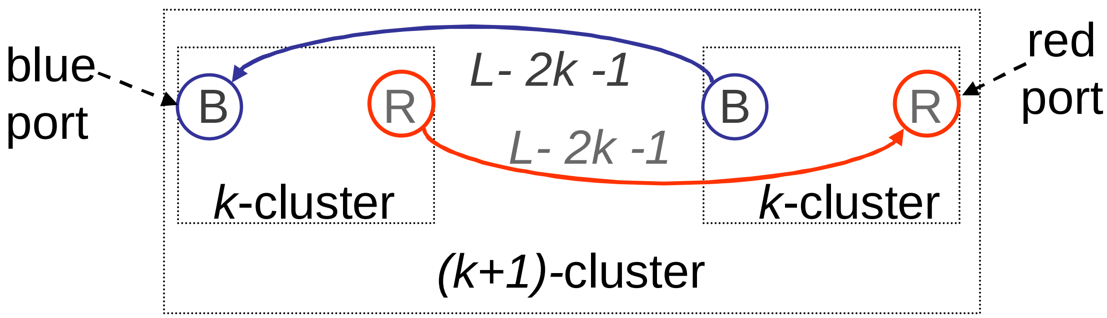
Network Construction (1)

(a) Construct *1-cluster* with two nodes:



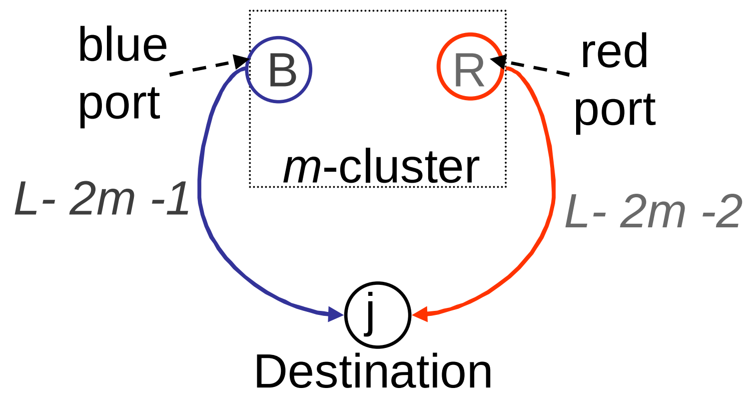
$L \approx 2 \log n + 4$

(b) Recursively construct *(k+1)*-clusters:

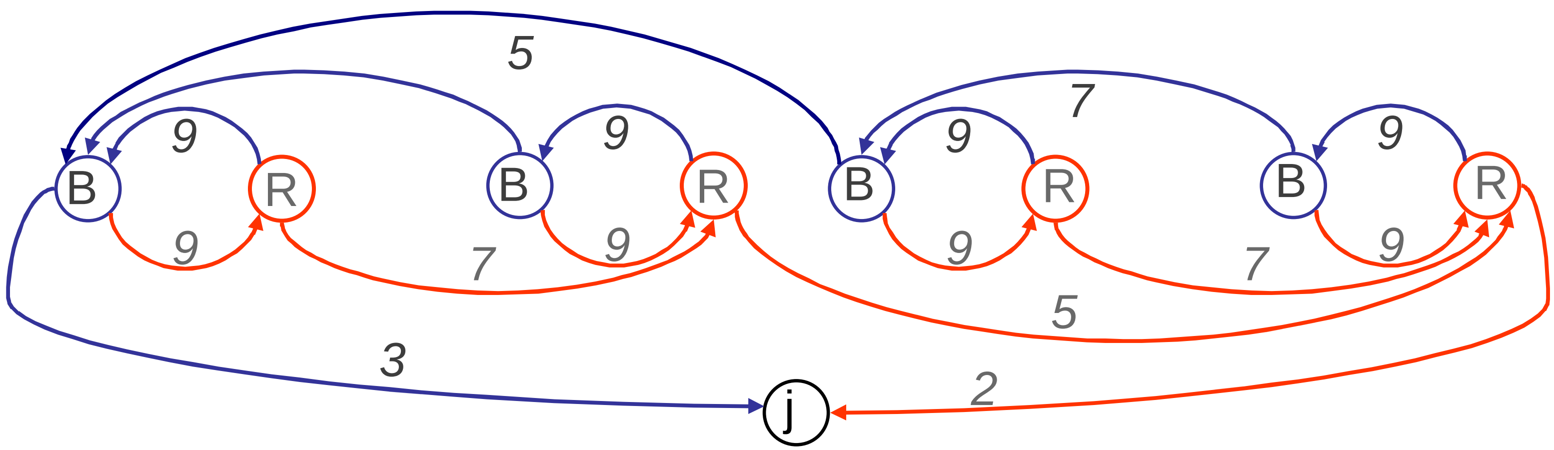


Network Construction (2)

(c) Top level: *m*-cluster with $n = 2^m + 1$ nodes.



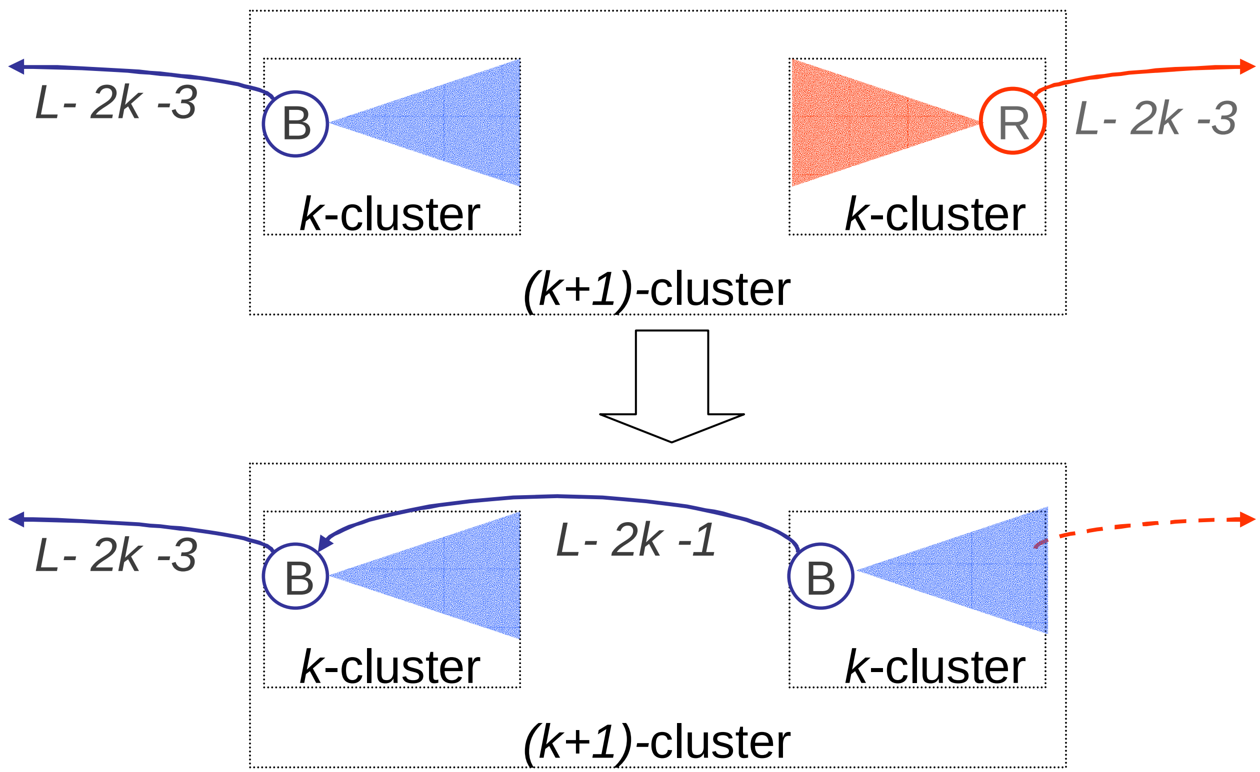
Final network ($m = 3$):



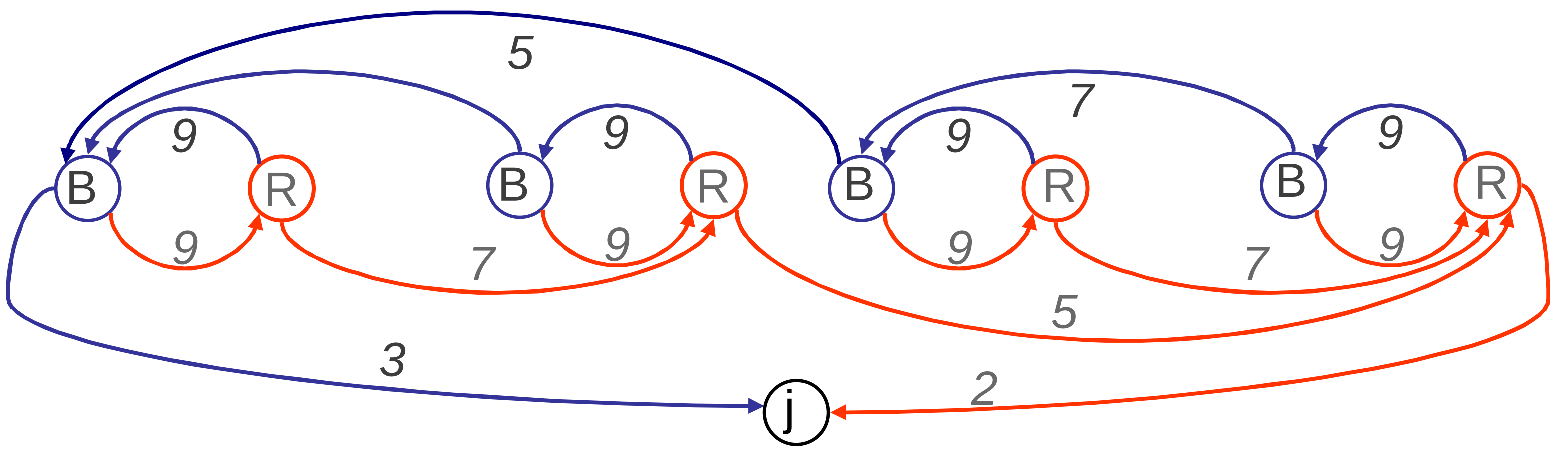
Optimal Spanning Trees

Lemma: *W*(*blue tree*) = *W*(*red tree*) + 1 ≥ *W*(*any other sp.tree*) + 2

Proof: If a directed spanning tree has red and blue edges, we can increase its weight by at least 2:



Proof of Theorem 2



- MDST T^* is the blue spanning tree.
- For any blue node B, T^{-B} is the red spanning tree on $N - \{B\}$.
- A small change in any edge, red or blue, changes

$$p^B = W(T^*) - u_B(T^*) - W(T^{-B})$$

$\Rightarrow$ Any change triggers update messages to all blue nodes!

Summary

Dissertation shows that the distributed-computing context can have a major impact on mechanism design.

- Multicast cost-sharing
  - Budget-balanced, group-strategyproof mechanisms are hard.
  - Tractable mechanism with bounded budget deficit.

- Interdomain routing
  - Lowest-cost routing mechanism is easy in BGP-based computational model.
  - Next-hop policy routing mechanism is hard in BGP-based computational model.

Some Open Questions..

- Multicast cost-sharing: Can we achieve constant-factor approximate budget-balance with O(*log n*) messages per link?
- Routing: Can we prevent strategy in computation without digital signatures?
- General:
  - New solution concepts
  - Other problem domains (caching, wireless, ...)