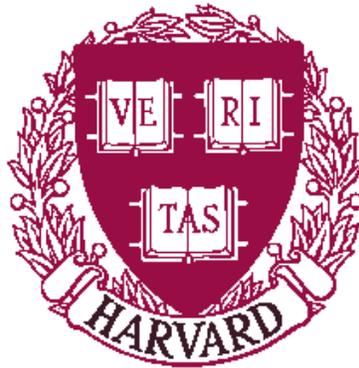


**Collusion-resistant mechanisms for
lowest-cost interdomain routing**

Geoffrey Goodell
and
David Parkes

TR-22-04



Computer Science Group
Harvard University
Cambridge, Massachusetts

Collusion-resistant mechanisms for lowest-cost interdomain routing

Geoffrey Goodell
goodell@eecs.harvard.edu

David Parkes
parkes@eecs.harvard.edu

May 14, 2003

1 Abstract

The problem of determining the price that an Internet Service Provider should charge for connectivity has received recent attention in several recent articles [2, 5, 10]. The central idea is that it is possible to design a distributed mechanism to solve the lowest-cost routing problem for the network. We propose to extend the solution put forth by Nisan and Ronen to accommodate a more realistic model of collusion among ISPs. We will begin by defining the problem: what are the ways in which an ISP might benefit from collusion? We will propose a mechanism that aims to protect against some of these forms of collusion, focussing on the problem of collusion for the purpose of increasing payoff once a particular lowest-cost route is selected. In order to determine under what circumstances the mechanism is incentive-compatible, we will analyze the rational collusive strategy for this mechanism. Finally, we will investigate the effects that our mechanism has on choosing the lowest-cost routes to determine the extent to which it is possible to create a system that both guards against some collusive behavior and still promotes the selection of lowest-cost routes.

2 Motivation

The Internet consists of an amorphous collection of internet-worked Autonomous Systems (ASes). ISPs charge each other for connectivity services, which essentially consist of a set of agreements that results in the establishment of a path to carry packets from their source to their destinations. In practice, we observe that these agreements tend to take two forms: customer-provider arrangements, in which one party pays another for connectivity, and peering relationships, in which parties generally agree to provide transit services for each other without a monetary exchange. This observation suggests an implicit hierarchy in the underlying topology of the Internet, and while such a hierarchy may be convenient for organization, it seems that intuitively there is no inherent need for such strict rigidity. In fact, perhaps it is in the best interests of some ISPs to organize differently, and a mechanism for determining the winner of a route and the necessary payments to the various nodes involved may help promote incentive-compatibility in this industry.

Choosing a VCG mechanism that solves the combinatorial auction that considers routes as bundles of edges is a good

start, but it does not capture the inevitable collusion between ISPs that has the potential to undermine the system. We cannot provide a different mechanism without waiving some of the advantages of the VCG mechanism, but what if we *expect* ISPs to collude instead? Here, we have the problem of collusion. Yokoo[14] has provided some negative results about the extent to which mechanisms can capture group-strategyproofness, and clearly our options are quite limited. However, it may be possible to provide a mechanism that protects against the ill-effects of some forms of collusion by assuming that some kinds of collusion are unavoidable and configure the game to provide optimal payoffs given this collusion. For example, suppose that a particular path from the source to the destination is chosen because it is the least-cost route. We may want to dispense only the marginal contribution of the path equally to the various ISPs involved, rather than dispensing the marginal contribution of each link to the ISP responsible for that link, since it is possible that the ISPs have colluded to get a better payoff from the VCG mechanism as a whole (and then used transfer payments to share the wealth). Other possibilities for protecting against collusion exist as well, such as considering the case in which two or more ISPs must be chosen together or not at all in order to create a valid path.

Consideration of only the possible paths from a source to a destination imposes a certain ordering on the set of edges, and thus we don't have the possibility for seeing truly arbitrary bundles that we do in the case of the generalized combinatorial auction. By taking into account some *partial* knowledge about the potentially colluding nodes, it may be possible to exploit this structure to yield some intermediate results that are not subject to all of the negative theorems proven about combinatorial auctions in general. By taking into account some collusive behavior, we may be able to provide a distributed mechanism that is of greater practical utility than one that yields the outcome dictated by VCG.

It is not difficult to show that using the VCG mechanism suggested by Nisan and Ronen, it is possible for an ISP to establish two ASes and provide links and corresponding advertised link costs such that the ISP can receive a higher payment while providing the same service.

3 Problem Statement

We begin by considering the routing information conveyed by the Border Gateway Protocol (BGP), the de facto interdomain routing system used on the internet today[12]. We presume the abstract model of autonomous systems and BGP provided by Griffin and Wilfong[4]. This model essentially abstracts away the details of intradomain routing, considering autonomous systems as self-contained, centrally-administered entities that provide connectivity between nodes in the network. Thus, for which the cost of routing packets internally is recouped by charging users of the network for the service of transiting their packets. One can envision the nodes as “peering points” to which a number of ISPs connect, and the service for which an ISP charges users of the network is represented by edge between such peering points.

One simple use of this model is to determine how much to charge a user of the network for connectivity between a source and a destination. Thus, we have a lowest-cost routing problem. The various ISPs advertise their costs for transiting data between nodes in the network. The user chooses the shortest path, and the mechanism determines how much the user should pay the various ISPs (edges) along this path. So, we desire a mechanism such that it is a dominant strategy for edges to advertise their true costs.

We refer to a few definitions involving theoretical properties of computational mechanisms, as compiled by Jackson[7].

Definition 3.1 *A mechanism is **efficient** if it maximizes the sum of the utilities of all of the agents in the system.*

Definition 3.2 *A mechanism is **individual-rational** if its payoff function satisfies the constraint that participation in the mechanism is always a (weakly) dominant strategy for all agents in the system.*

Definition 3.3 *A mechanism is **strategyproof** (or **incentive-compatible**) if truth-telling is a (weakly) dominant strategy for all agents in the system, given that agents act independently.*

Nisan and Ronen observe[10] that we can achieve edgewise incentive-compatibility by choosing a VCG mechanism in which each edge is an agent. Suppose we have a graph G with some source node S and destination node T . For every edge $e \in G$, we use c_e to represent the actual cost to edge e and \hat{c}_e to represent the cost advertised by edge e . We use d_G to represent the length of the shortest path from S to T . So, using the edgewise VCG mechanism, we define the payment p_e to each edge as follows.

$$p_e = \hat{c}_e + (d_{G \setminus e} - d_G) \quad (1)$$

So, the payment to an edge is not a function of its bid, and truth-telling is a dominant strategy. However, this mechanism lacks a certain degree of feasibility for deployment in the real world, since the model upon which it is based has some notable flaws. In particular, the model fails to take into account how edges may manipulate the system by violating the assumption

that each edge functions independently and without an interest in the payment afforded to other edges. We observe that not only is there reason for edges to collude, but that even individual ISPs often provide a diversity of connections that involve a multitude of peering points, ensuring that collusion between edges is in some ways inevitable.

Feigenbaum et al.[2] cite another factor that limits the mechanism’s feasibility, namely the fact that from the perspective of an observer of BGP *UPDATE* messages, the Internet Service providers are nodes and not edges. Feigenbaum et al. then formulate a model for the lowest-cost routing problem in which nodes are the agents that charge customers for path utilization. Thus, the cost of a path is not the sum of the weights of the edges along the path, but rather the sum of the values of the nodes that the path traverses. This formulation exhibits both advantages and disadvantages when compared to the formulation of Nisan and Ronen; we will discuss these matters at greater length in the section describing our simulations.

3.1 Collusion

Perhaps the most immediately evident concern with the edgewise VCG mechanism presented in the paper by Nisan and Ronen is that a group of agents can work together to yield a higher payoff for the group as a whole than they would by bidding truthfully as individuals. And since they are working together, if we assume a transferable utility market, then all parties involved will be able to exchange the additional profit such that each agent receives as least as much or more than what it would have received had it not participated in the collusive effort.

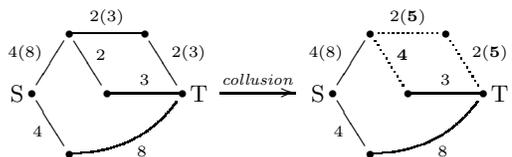


Figure 1: How collusion benefits a set of edges.

Figure 1 provides an example of how a set of edges can work together to increase the overall payment to that set. The graph on the left illustrates a system in which edges bid their true values. The edge weights represent advertised costs, and the numbers in parentheses represent payment (i.e., the value of p_e) for that edge. The graph on the right shows the same network, but in this case the three dotted edges agree to collude. Observe that an edge not included in the shortest path increases its advertised cost from 2 to 4, thus allowing edges on the shortest path to procure a greater Vickrey discount, thus yielding an aggregate payment of 10 to the collusive group rather than the 6 that its edges would otherwise have received. Since all three nodes are able to transfer their respective utility, the edges on the shortest path can use a portion of their collective profit to subsidize the effort of the edge that advertises an invalid cost.

This attack is only possible when multiple competitors (i.e. edges that cannot both lie on the same shortest path) decide to arrange for *one* party to win and then divide the proceeds. This is the classic trust arrangement in which parties manage to circumvent a price war. It is often advantageous and never deleterious for a set of competitors to do this.

3.2 The Sybil attack

Another threat to incentive-compatibility is the *Sybil attack*[1], in which an edge takes on multiple identities in order to procure a greater payment from the system. Observe that the edgewise VCG mechanism provides to each edge e a Vickrey discount equal to the difference between the length d_G of the shortest path of the graph and the length $d_{G \setminus e}$ of the shortest path of the graph with that edge removed. Thus, if an edge on the shortest path were to divide itself into n edges connected in series, it could receive a payment of $\hat{c}_e + n(d_{G \setminus e} - d_G)$.

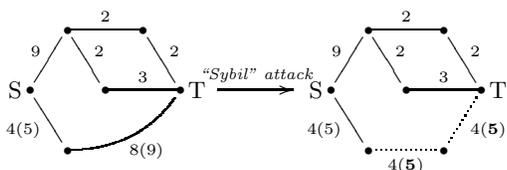


Figure 2: Exploitation of multiple identities.

Figure 2 provides an example of a Sybil attack in action. The graph on the left shows the advertised cost and payments when each of the edges actually “occurs naturally” in the network. Observe that the lower path is chosen because it has length 12. Let e be the edge with length 8. Observe that $d_{G \setminus e} - d_G = 1$. If e divides itself into two edges of length 4, then even though each of those two edges may advertise its true cost, the entire Vickrey payment is offered to both edges, so we see an added utility of 1.

Unlike the collusion strategy above, any edge can perform the attack without collaboration with other edges. In fact, if an agent splits into n “virtual” agents in series with the same total advertised cost, then it will automatically increase its received Vickrey discount by a factor of n . Thus, it is *always* advantageous for an edge on the shortest path to exploit multiple identities in this fashion whenever it is on the shortest path, and it is *never* deleterious to do so, even if it is not on the shortest path. Note that this attack bears a certain similarity to “false-name bids” [13, 6]. We say that an auction is susceptible to false-name bids if a single bidder can submit two or more bids, claim that each bid corresponds to a distinct bidder, and collect a greater utility than she could by acting as a single bidder. In both the case of the false-name bids and the case of our payments to edges, participants have an incentive to establish multiple identities; arguably it is not difficult for ISPs to subdivide or facilitate the emergence of new ISPs willing to collude. Yokoo has shown[14] that VCG with marginal-decreasing values yields an auction that is resistant to false-name bids, but in

general VCG mechanisms are not false-name bid proof. Note that we do not have marginal-decreasing values in our auction.

4 Approach

In order to address the difficulties associated with collusion among participants in the edgewise VCG mechanism, we must acknowledge two fundamental incentive characteristics:

- *Parallel manipulation.* For any set of edges such that not all members are on the shortest path, members have an incentive to collude by raising the cost of all edges not on the shortest path.
- *Serial manipulation.* All edges along the shortest path have an incentive to represent themselves as a set of several distinct edges in series in order to receive multiple instances of the Vickrey discount, and no edges have a disincentive to represent themselves as a set of several distinct edges in series.

We propose two mechanisms that attempt to protect against some forms of collusion. Fundamentally, we must consider sets of ISPs as consortia that should be paid as if they were each a single entity. The difficulties lie in determining what edges form the various consortia and then determining how to provide payments in such a manner to ensure incentive-compatibility, the primary motivation behind the choice of the VCG mechanism for the edgewise treatment in the first place. We begin by formally defining our notion of collusion.

Definition 4.1 *We say that a set C consists of edges that are in collusion if its edges are able to coordinate a strategy by which each edge $e \in C$ advertises a (possibly nontruthful) cost \hat{c}_e such that in equilibrium, the aggregate utility to the set C is maximized.*

We presume (1) transferable utility, (2) the ability for edges to communicate with each other via side channels, and (3) the ability for edges to offer each other payment via a variety of procedures that could potentially be difficult to audit. In addition, we presume that the “coalitional” problem is solved: there is no private information within a coalition, and agents within a coalition can distribute the surplus efficiently among themselves. The definition prompts a second definition, which is essentially a reformulation of our notion of strategyproofness to accommodate collusion.

Definition 4.2 *We say that a mechanism is incentive-compatible under collusion (C-IC) if truthtelling is a (weakly) dominant strategy for all agents in the system, given that for all agents x and y not in collusion, agent x acts independently of agent y .*

Thus, the edges will choose to advertise values that maximize the payment to the group, because this is necessarily at least as beneficial to the group as having each edge advertise a value

that maximizes its individual payment, because the group is assumed able to redistribute the payment in such a manner that all edges are (weakly) better off.

Our first mechanism, which we call *Omniscient*, assumes that the problem of determining which groups are in collusion has already been solved *ex ante*. Given sets of collaborators, we treat each such set as a single entity, and we design the payment scheme to provide incentive-compatibility at this level rather than strictly for individual edges.

Next, we extend this basic principle by describing a framework that allows us to describe a more generalized class of mechanisms resistant to Sybil attacks. This framework is designed to allow for the possibility that we have less *a priori* knowledge of the structure of the collusive groups. We then describe a few mechanisms implementable with this algorithm, including a way in which we can impose a more “natural” collusive structure upon the network in order to protect against a few well-defined attacks.

5 The *Omniscient* Mechanism

As the name suggests, the *Omniscient* mechanism knows who the colluders are and provides payment with this in mind. We treat each collusive group as a single unit, presuming that the group will internally provide the proper incentives for each constituent member to participate, and we divide payment among the individual edges within a collusive group.

5.1 Mechanism

The algorithm for running the *Omniscient* mechanism is as follows.

1. Given graph G , presume that we know which sets of edges are in collusion, and partition the edges into subsets $G = (C_1, \dots, C_k)$ such that the following statements hold:
 - $\forall i, j < k : i \neq j \Leftrightarrow C_i \cap C_j = \emptyset$
 - $\forall i < k$: the edges of C_i are in collusion.
 - For all $j < k$, we have that $i \neq j$ implies that for all $e \in C_j$, the edges of $e \cap C_i$ are not in collusion.

Refer to all members of $\{C_1, \dots, C_k\}$ as collusive subsets of G .

2. Run the ordinary VCG mechanism, but instead of considering the set of agents to the set of edges of G , consider the set of agents to be the set of collusive subsets of G . For each collusive subset C that contains an edge e that is on the shortest path P , pay C the advertised cost of the edges in $C \cap P$ plus the marginal contribution of C . Distribute the payment to C among its constituent members.

The problem of how to distribute payment to participating members of a coalition is worth consideration. We assume that members of a coalition will redistribute this payment anyway,

since they are in collusion, but nonetheless we must provide them with an initial payment. So, for the purpose of discussion, let us say that the payment is divided evenly among coalition members. (We address concerns about this in the next subsection.)

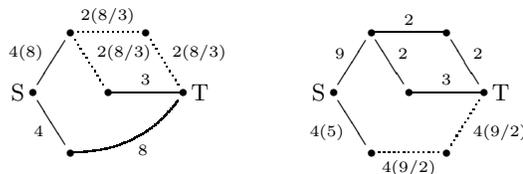


Figure 3: The *Omniscient* mechanism.

Figure 3 shows how the *Omniscient* mechanism can be used to protect against the deleterious effects of collusion. The graph on the left demonstrates that it is possible to preserve incentive-compatibility in the event that a particular set of agents is known to collude. Suppose that we know that the dotted edges form a collusive subgroup C . Treating C as a single agent results in paying C as if its constituent members had colluded to optimize their aggregate utility; we divide the total payment 8 evenly among the edges $e \in C$.

The graph on the right demonstrates how it is possible to eliminate the incentive for an edge to employ a Sybil attack. Suppose that we know that the dotted edges form a collusive subgroup C' . Treating C' as a single agent ensures that we only administer the Vickrey discount exactly once for C' . Thus, we provide C' with 9, which is the same payment that C' would have received had it consisted of a single link.

5.2 Properties

We observe that for any collusive subset C , the payment to a coalition C is as follows.

$$p_C = \begin{cases} [\sum_{e' \in C \cap P} c_{e'}] + (d_{G \setminus C} - d_G) & \text{if } C \cap P \neq \emptyset \\ 0 & \text{otherwise.} \end{cases} \quad (2)$$

The VCG class of mechanisms are the only mechanisms that are efficient, individual-rational, and strategyproof.[7, 8] Thus, whenever we choose to implement a mechanism that is not VCG, we can no longer guarantee all of these properties. As such, with this mechanism we observe that an individual edge may receive less than its reported cost: for example, consider Figure 4. The path from S to T by way of node B is shorter, and both dotted edges receive payment 4. Note that this is less than the cost advertised by the edge from B to T . However, this is not a problem in our collusion context, since we know that the dotted edges can redistribute utility since they are in collusion and we know that the total utility to the dotted edges will be at least as much as it would be if they were not in collusion.

Incentive-compatibility (under collusion). By treating each collusive subset C as a single agent with respect to the VCG

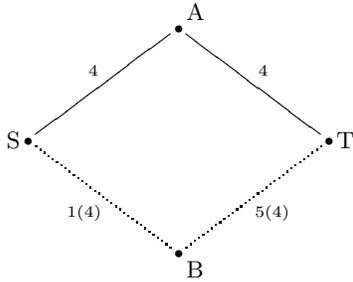


Figure 4: An individual edge may receive less than its cost.

mechanism, we ensure that for each edge in C , increasing the cost of the edge will either have no effect on the utility to C (in the case that it is not on a path shorter than any path not including any nodes in C), or it will only cause C to no longer contain any edges along the shortest path (in all other cases). Similarly, decreasing the cost of an edge will either have no effect on the price paid to C (in the case that the paths including that edge are already too expensive), or it will decrease the utility to C (in the case that it wins the auction and gets paid less than its true valuation). Thus, as in the direct VCG mechanism, the payment to C is not a function of the advertised costs of the edges in C and truth-telling is a (weakly) dominant strategy for each of the edges $e \in C$. Further, underbidding can only result in a situation in which the payment to C is less than the advertised cost of the edges in $C \cap P$ plus the marginal contribution of C , and overbidding can only result in not being selected.

Efficiency. The mechanism is strategyproof. Since the mechanism always chooses the shortest advertised path, we know that the mechanism will always choose the shortest path; thus, the mechanism is efficient.

Revenue. We observe two different phenomena at work here. In order to provide protection against collusion whereby competing agents raise prices to receive higher payment, the mechanism must incur a cost. We can think of this cost as “financing strategyproofness,” in a way: we essentially provide the edges in the collusive subset with the maximum utility that they can get by colluding. On the other hand, the mechanism actually receives increased revenue from protection against Sybil attacks. For a collusive subset C' that can be replaced by a single edge, rather than paying each edge $e \in C'$ for its marginal contribution, we spread the Vickrey discount across C' , thus saving a value of $(n - 1)(d_{G \setminus C} - d_G)$.

5.3 Limitations

The most important shortcoming of this mechanism is that it requires *a priori* knowledge of the members of the collusive subsets. Indeed, in reality it may be difficult to know in advance which edges are capable of colluding with each other; there is no way to understand why one group is more likely to collude than

another group. Also, the trouble with encouraging collusion in general is that collusion eliminates competition: if the whole graph colludes, then the user ends up paying the reservation price.

However, the *Omniscient* mechanism may become more useful when we consider that there is not a one-to-one correspondence between ISPs and edges. As noted earlier, large ISPs effectively connect many different pairs of peering points; it is thus reasonable to model an ISP as a collusive subset of edges in the graph. This model provides a good example of a circumstance in which collusive information is readily available, and the Omniscient mechanism provides a concise way of allowing for collusion within the ISP without fundamentally breaking the properties of the edgewise VCG mechanism.

5.4 The *Restricted Omniscient* mechanism

Following the same reasoning, it is possible to create a slightly different mechanism to achieve a similar goal. Partition all of the edges into collusive subgroups as in the *Omniscient* mechanism and determine the shortest path. Next, instead of determining the marginal contribution of the entirety of an individual collusive subgroup in order to determine the Vickrey payment of that subgroup, consider only the intersection of that subgroup with the shortest path. In particular, only the edges that are both along the shortest path and within the collusive subgroup are considered to be in collusion. Thus, this mechanism determines the marginal contribution of the intersection of each of the individual collusive subgroups with the shortest path, and distribute the Vickrey payment to the edges included in that intersection. Refer to this process as the *Restricted Omniscient* mechanism.

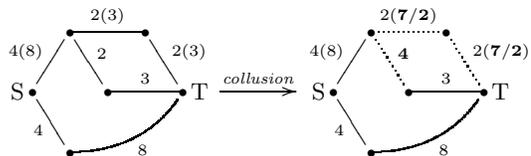


Figure 5: The *Restricted Omniscient* mechanism.

Like the *Omniscient* mechanism, the *Restricted Omniscient* mechanism addresses Sybil attacks, but the *Restricted Omniscient* mechanism does nothing to address routine collusion among edges on parallel paths. The result is a mechanism that considers only edges along the chosen path as possible colluders, meaning that the payment to an individual edge will never be greater than what it would have been under the VCG mechanism. Figure 5 demonstrates how the *Restricted Omniscient* mechanism would handle the collusion portrayed in Figure 1. Observe in particular that the coalition represented by the dotted edges can do better; for example, the dotted edge that bid 4 could bid something between 4 and 5 instead and increase the payment to the other two edges. However, it is important to note that the *Omniscient* mechanism solves this problem by

paying more to the coalition. From the standpoint of revenue to the user of the network, the *Restricted Omniscient* mechanism always yields a total payment that is at most that given by the straightforward, edgewise VCG mechanism. Conversely, the *Omniscient* mechanism provides no such guarantee: because edges appearing in parallel may both be considered part of the same collusive subgroup, the cost of ensuring strategyproofness given that they collude may actually result in a higher total cost than the edgewise VCG mechanism.

6 Partial Knowledge Mechanisms

The *Omniscient* and *Restricted Omniscient* mechanisms are useful when we know the members of the various collusive groups in advance, but in reality this may often be impossible. Since arguably it is the nature of collusion to exist as a secret process transacted via confidential channels, we may expect that often it is not possible for the mechanism to have complete knowledge of all of the collusive subgroups in advance. For example, the mechanism may know what sets of edges could potentially collude, but it may not know in advance which of these sets actually will. In such cases, it is desirable to have a mechanism that can accommodate *partial* knowledge as well. In this manner, we may be able to get better performance vis-a-vis efficiency and revenue without violating any of the well-known impossibility results.

We construct the following algorithm to describe a class of mechanisms that may accommodate such partial knowledge.

1. Determine the set $C = \{C_1, \dots, C_k\}$ of *possible* collusive subgroups. Some of these collusive subgroups will be “activated” later.
2. Collect reported edge costs and determine the shortest path, breaking ties randomly.
3. Select n values $\{i_1, \dots, i_n\} \subseteq \{1, \dots, k\}$ such that the set of edges contained in the members of the set of collusive subgroups $C' = \{C_{i_1}, \dots, C_{i_n}\}$ form a partition of the set of all edges. Refer to C' as the set of “activated” collusive subgroups, and run the *Omniscient* mechanism, as described in Section 5, considering only the set of activated collusive subgroups.

To describe a specific mechanism, it is sufficient to describe a method of obtaining C for Step 1 and a method of obtaining C' for Step 3. Observe that if the edges contained in the sets $\{C_1, \dots, C_k\}$ form a partition of the set of all edges, then the generalized algorithm instantiates the *Omniscient* mechanism.

Example 6.1 *We can express the Restricted Omniscient mechanism in terms of the generalized algorithm.*

Presume that we start with a partition $S = \{S_1, \dots, S_m\}$ of the set of all edges such that each set $S_i \in S$ corresponds to a set of edges that could potentially collude. For Step 1 of the algorithm, define C as follows.

$$\forall S_i \in S : \forall c \in C : c \subseteq S_i \Leftrightarrow c \in C \quad (3)$$

This ensures that the intersection of S_i with the shortest path is a possible collusive subgroup. Step 2 of the algorithm finds the shortest path P as usual. For Step 3 of the algorithm, define C' as follows. For every edge not on the shortest path, include every singleton set consisting of that edge in C' , since every edge may trivially collude with itself. Also, for each set S_i of edges that intersects P , include the set of edges that consists of $S_i \cap P$. We know that this set is in C , since C contains all subsets of S_i . More formally, we have the following expression.

$$\begin{aligned} \forall c \in C & : (|c| = 1 \wedge c \cap P = \emptyset) \\ & \vee (c \subseteq P \wedge (\forall c' \in C' : c \not\subseteq c')) \\ \Leftrightarrow & c \in C' \end{aligned} \quad (4)$$

Having produced a well-defined method for determining C and C' , we have completed the reduction. \square

6.1 Properties

Next, we outline some of the salient theoretical properties of the framework; we believe that the generalized algorithm is powerful enough to provide useful properties yet flexible enough to have application in a variety of circumstances. Suppose that M is a mechanism produced according to the generalized algorithm described above. Then, we have the following propositions.

Proposition 6.1 *If the activated collusive groups along the shortest path are all singleton sets, then M produces payments equivalent to the VCG payments.*

Proof. This follows directly from the fact that running the *Omniscient* mechanism on the set of collusive groups consisting of singleton edges yields VCG payments to all edges.

Proposition 6.2 *If one of the activated collusive groups includes all of the edges on the shortest path, then the total payment provided by M to the nodes on the shortest path is equal to the reported cost of the shortest path that does not contain any of the edges in that collusive group (or the reservation price, if removing the collusive group from the network disconnects the source from the destination).*

Proof. If the shortest path is subsumed by a single collusive group G , then the *Omniscient* mechanism divides the aggregate cost of G plus the marginal contribution of G among its members. The marginal contribution of G is equal to the difference between its aggregate cost and the aggregate cost of the shortest path that does not contain any edges in G . If there is no such shortest path, then it will assign the reservation price.

Proposition 6.3 *If an activated collusive group contains edges on the shortest path but does not contain edges not on the shortest path, then the aggregate payment to the members of the collusive group under M does not exceed the aggregate payment to the collusive group under the edgewise VCG mechanism.*

Proof. Consider the Vickrey discount offered to individual edges in the edgewise VCG mechanism. Suppose that E is a set of edges along the shortest path. Each edge $e \in E$ gets its advertised cost plus the difference between the cost $c(P)$ of the shortest path and the cost $c(P \setminus e)$ of the shortest path not containing that edge. Then $c(P) - c(P \setminus E)$ is just the cost of the shortest path not containing any edges in E , and we know the following:

$$\sum_{e \in E} (c(P) - c(P \setminus e)) \geq c(P) - c(P \setminus E) \quad (5)$$

It follows that paying the set of edges along the shortest path its marginal contribution will cost at most as much as paying each edge its individual marginal contribution.

Corollary 6.1 *If all activated collusive groups contain edges on the shortest path but do not contain edges not on the shortest path, then the utility to the user can never be worse than it would be under the edgewise VCG mechanism.*

Proof. Follows directly from Proposition 6.3.

6.2 The *Trusted Node* mechanism

To demonstrate the flexibility of our framework, we show how we can use the algorithm to describe a mechanism in response to the following scenario. Suppose that we want to prevent Sybil attacks, and we do not know which groups of edges are capable of collusion. Suppose also that we know a set of *nodes* that will not be used as part of an attack in which some set of edges claim connectivity to that node in order to derive the benefit of a Sybil attack. In this case, we can presume that connections between trusted nodes are legitimate, even if they traverse untrusted nodes in the process, but that the particular path taken from one trusted node to another through untrusted nodes is unimportant. This is to say that the mechanism treats the trusted nodes as an abstraction, choosing the shortest path by examining the network consisting exclusively of trusted nodes and then providing VCG payments to edges along this path. If the shortest path between two consecutive trusted nodes along this path requires traversal of untrusted nodes, the VCG payment is divided among the edges between the two consecutive trusted nodes.

Figure 6 illustrates the behavior of the *Trusted Node* mechanism. Formally, the process of determining the payment to the edges along the shortest path consists of the following four-stage process, with each stage corresponding to one of the graphs in Figure 6.

- (a) Determine the reported costs of each edge in the network, and partition the set of nodes into a *trusted* subset and an *untrusted* subset. The source S and destination T are presumed to always be trusted. In Figure 6, we include one untrusted node, marked as U . Refer to this graph as G_a .

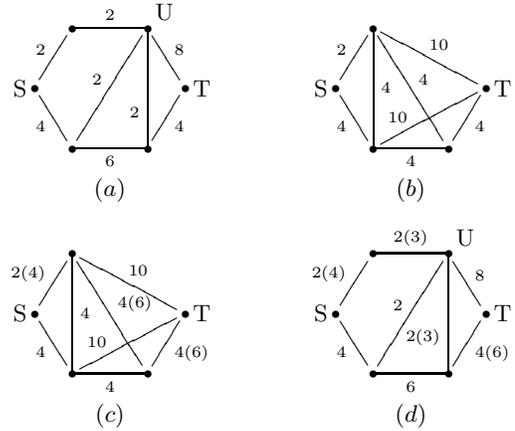


Figure 6: The *Trusted Node* mechanism.

- (b) Construct a new graph G_b as follows. For every trusted node $x \in G_a$, create a corresponding node $x' \in G_b$. Next, suppose that $\{x, y\}$ are trusted nodes in G_a and $\{x', y'\}$ are their corresponding nodes in G_b . If x and y are connected via one or more paths that does not traverse any other trusted nodes, then create an edge between x' and y' and define its weight to be the shortest such path; otherwise, do not create an edge between x' and y' .
- (c) Determine the shortest path P_b in G_b between the source and destination, and run the VCG mechanism to determine the payment to edges along this path.
- (d) Since G_a corresponds to the “real” network, we now have the task of issuing payments to edges in G_a . So, assign payments as follows. Suppose that nodes $\{x, y\} \subseteq G_a$ correspond to nodes $\{x', y'\} \subseteq G_b$, respectively. If the link between x' and y' received payment p in stage (c), then distribute that payment evenly among the edges along the shortest path connecting x and y that does not traverse any other trusted nodes.

We can see how the *Trusted Node* mechanism may be useful in circumstances in which we want to hold a VCG auction to determine the edge payments but we do not have enough information about each of the nodes to be sure that the user of the network will not have to issue the VCG discount multiple times to a single set of collusive edges. With the *Trusted Node* mechanism, as long as the diameter of the graph consisting of trusted nodes (G_b) is bounded, then so is the number of times that the mechanism will issue a Vickrey discount. This property may be useful for users who believe that the risk of lost revenue inherent to the VCG mechanism is prohibitive.

Finally, as promised, we show how to express the *Trusted Node* mechanism in terms of the generalized algorithm described at the beginning of the section.

Example 6.2 *We can express the Trusted Node mechanism in terms of the generalized algorithm.*

Presume that we start with a network $G_a = \langle V, E \rangle$, and $V = T \cup U$, where $V_t = \{t_1, \dots, t_m\}$ constitutes the set of trusted nodes and $V_u = \{u_1, \dots, u_n\}$ constitutes the set of untrusted nodes. If v_1 and v_2 are nodes, then let $SP(v_1, v_2)$ denote the shortest path between nodes v_1 and v_2 . For Step 1 of the generalized algorithm, define C as the set of all acyclic paths connecting two nodes in V_t that do not traverse other nodes in V_t . More formally, if $t_i, t_j \in V_t$, then $SP(t_i, t_j) \in C$. For Step 2, compute the shortest path P between the source and destination as usual. Suppose without loss of generality that the nodes adjacent to the edges in P form an ordered sequence $P_v = (v_1, \dots, v_{|P|+1})$, where v_1 is the source and $v_{|P|+1}$ is the destination. Next, suppose that $P_t = (t_1, \dots, t_k) \subseteq P_v$ is the ordered set of trusted nodes along the path P . Since the source and destination are necessarily trusted, we know that $t_1 = v_1$ and $t_k = v_{|P|+1}$. Since the shortest path from t_1 to t_k consists of the shortest path between consecutive pairs of members of P_t , we know that $\forall i : 1 \leq i < k : SP(t_i, t_{i+1}) \subseteq P$. Thus, we formally define C' for Step 3 of the generalized algorithm as follows.

$$\forall c \in C : (|c| = 1 \wedge c \cap P = \emptyset) \vee \exists i : c = SP(t_i, t_j) \Leftrightarrow c \in C' \quad (6)$$

Finally, we demonstrate the added utility of the generalized algorithm by illustrating how it can take advantage of partial knowledge about collusive subgroups. To accomplish this, we show that the *Trusted Node* mechanism is distinct from *Omniscient* and *Restricted Omniscient* mechanisms described in Section 5.

Proposition 6.4 *The Trusted Node mechanism cannot be reduced to an instance of the aforementioned Omniscient or Restricted Omniscient mechanisms.*

Proof. To demonstrate this proposition, we must provide an example of a network (a set of trusted nodes, untrusted nodes, and edges) such that the *Trusted Node* mechanism behaves differently from the other mechanisms. We observe that the salient property of the *Trusted Node* mechanism that it exploits the flexibility of the generalized algorithm in taking into account partial knowledge. For example, while the *Omniscient* and *Restricted Omniscient* algorithms rely upon an *a priori* partitioning of the set of edges into collusive groups. So, it is sufficient to provide an example of a network for which two different reported edge costs cause the *Trusted Node* algorithm to yield two different sets of payments such that it is not possible for a single mechanism with *a priori* knowledge of collusive subgroups to provide the same payments given the same respective choices of reported edge costs. Since the VCG is used to determine payments to the subgroups in both cases, this means that it is necessary to find a network for which different edge weights result in different choices for the set of activated collusive subgroups C' .

Stage (a) of Figure 7 provides the an example of such a network. Observe that if $a + c < b + d$, then we execute Stages (c₁) and (d₁); otherwise we execute Stages (c₂) and (d₂). Note in

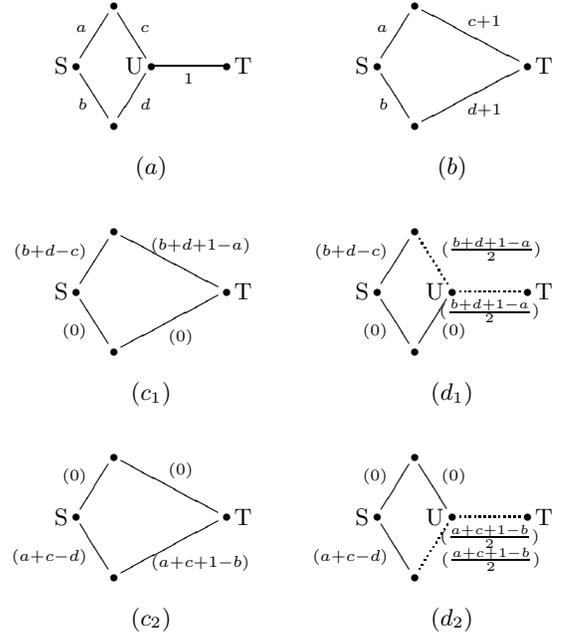


Figure 7: *Trusted Node* mechanism.

particular that the coalition to which the edge connecting the untrusted node U to the destination T is assigned is a function of the reported edge costs. Since the payment to that edge depends upon the coalition to which it is assigned as well as the shortest path of the graph, the benefit of the generalized algorithm is realized. It is significant that the *Trusted Node* mechanism is only one example of a “realistic” use of the generalized algorithm; there are other methods of using the algorithm to exploit partial knowledge.

7 Remarks

The *Omniscient* mechanism and the algorithm described in Section 6 are modifications to VCG that provide properties that provide some protection against collusion. In particular, the *Omniscient* mechanism gives us a means of characterizing collusion and getting a strategyproof auction at the layer of the collusive subgroups, which may represent, for example, an ISP or a consortium of ISPs.

Note that we made the decision to represent internet service providers as edges rather than as nodes; our model thus differs from the model presented by Feigenbaum et al. There are a few reasons for this decision:

- Links are actually owned by ISPs, and the peering points at which links meet are often collaborative efforts.
- Treating links as agents with some particular cost provides a more intuitive representation of peering arrangements and customer-provider relationships between ISPs.

- An individual ISP may want to charge different rates to traffic that traverses different neighbors.

However, there are some disadvantages to this model. Most importantly, while BGP UPDATE messages provide some information about the nodes that the messages traverse, they essentially do not provide information about the actual edges. Thus, it is not really possible to determine what the peering points are, given BGP data alone. This makes it difficult to use our model to run experiments with genuine data.

7.1 Efficiency

The *Omniscient* mechanism manages to preserve efficiency, but it essentially accomplishes this by sacrificing individual strategyproofness and replacing it with strategyproofness for a well-defined group structure. We can achieve efficiency in the presumptuous case as well, provided that we do not have any false negatives in our understanding of which edges are potentially in collusion with each other. We do not believe it to be possible to preserve efficiency in the general case, using an algorithm without knowledge of who is in collusion. However, we do not prove this result. We propose an experiment to evaluate the performance of this mechanism:

- **EXPERIMENT 1: Efficiency of Restricted Omniscient.** Generate random graphs of particular sizes and plot the social efficiency (ratio of chosen path length to optimal path length).

7.2 Revenue

Protecting against trust arrangements involving side payments between competitors is a task that costs the mechanism, but protecting against “Sybil” attacks actually saves the mechanism some money. Since mechanisms derived from the generalized algorithm are primarily designed to counter Sybil attacks, we submit that there is value in determining whether mechanisms such as *Trusted Node* positively affect revenue to the system. We use the model suggested by Joan Feigenbaum[2], in which we generate randomly connected networks such that each edge has weight 1, and we apply the VCG mechanism and compare the results to the output of the *Trusted Node* mechanism. We propose two experiments to evaluate this mechanism:

- **EXPERIMENT 2: Revenue of Restricted Omniscient.** Generate random graphs of particular sizes and plot the revenue to the dealer (ratio between RO revenue and VCG revenue).
- **EXPERIMENT 3: Revenue of Trusted Node.** Generate random graphs of particular sizes and plot the revenue to the dealer (ratio between TN revenue and VCG revenue).

8 Discussion

We note that ISPs already seem to have an established system for determining payments. While much of the details are secret, there is some existing literature that describes how this works. I intend to find some references and present an analysis of this with respect to our incentive models in order to demonstrate how a mechanism for lowest-cost routing may be useful. Admittedly, BGP is fundamentally about policy and not minimizing cost per se, but perhaps some illustrative examples will help shed light on how ISPs should act if their interests are easily described by a simple model in which each edge has a particular cost that is universally agreed upon by everyone. We will also explore the effect that instituting our mechanism has on efficiency.

We question whether Nisan[10] actually thought that we could model an ISP as a single edge in a network from a source to a destination that could either choose to transit my traffic or not. The reality is that large ISPs have many connections throughout the network, with numerous peering points, and if this is the case, then we must be prepared to extend the model to allow for the possibility that some edges in the network are in collusion *simply because the same ISP controls them*.

Our goal has been to present a concise exploration of several of the issues related to attaining a better pricing model for interdomain routing on the Internet. First, we provided an exploration of the usefulness of the model and mechanism put forth in the paper by Feigenbaum et al. given the realities of side channels for developing collusive strategies. We also designed something more practical that manages to resist certain forms of collusion.

We promised to offer some insight into why the existing pricing model has emerged and how we might want to provide incentives in order to deploy a mechanism-based system to price the routes chosen at each AS. We believe that despite our inability to handle many of the collusive attacks on the system, a mechanism for these tasks will help to streamline a process that today is mediated by humans and policies that are not at all optimized for efficiency, let alone strategyproofness. We believe that systems such as the one proposed by Feigenbaum et al. have promise in being used at least as a guideline for internet pricing.

Although it is well outside the scope of this project, we believe that it would be interesting to consider potential modification to routing in order to abrogate the need to choose just one route rather than many. This has the potential to mitigate some the problems associated with having to receive one bundle but not another; perhaps it would eliminate the need to do the relaxation in solving the linear program as well.

References

- [1] J. Douceur. The sybil attack. In *Proceedings of the IPTPS02 Workshop*, 2002.

- [2] J. Feigenbaum, C. Papadimitriou, R. Sami, and S. Shenker. A bgp-based mechanism for lowest-cost routing. In *Proceedings of the 21st Symposium on Principles of Distributed Computing*, pages 173–182, 2002.
- [3] J. Feigenbaum, C. Papadimitriou, and S. Shenker. Sharing the cost of multicast transmissions. In *Proceedings of the 32nd Annual ACM Symposium on the Theory of Computing*, pages 218–227, 2000.
- [4] T. Griffin and G. Wilfong. An analysis of bgp convergence properties. In *Proceedings of SIGCOMM 99*, pages 277–288, 1999.
- [5] J. Hershberger and S. Suri. Vickrey prices and shortest paths: What is an edge worth? In *IEEE Symposium on Foundations of Computer Science*, pages 252–259, 2001.
- [6] A. Iwasaki, M. Yokoo, and K. Terada. A robust open ascending-price multi-unit auction protocol against false-name bids. In *Proceedings of the 4th ACM Conference on Electronic Commerce*, 2003.
- [7] M. O. Jackson. Mechanism theory. In *The Encyclopedia of Life Support Systems*. EOLSS Publishers, 2000.
- [8] A. Mas-Colell, M. D. Whinston, and J. R. Green. *Microeconomic Theory*. Oxford University Press, 1995.
- [9] H. Moulin and S. Shenker. Strategyproof sharing of submodular costs: Budget balance versus efficiency. In *Economic Theory*, To appear, full text available at <http://www.aciri.org/shenker/cost.ps>.
- [10] N. Nisan and A. Ronen. Algorithmic mechanism design. Number 35 in *Games and Economic Behavior*, pages 166–196, 2001.
- [11] D. Parkes, J. Kalagnanam, and M. Eso. Achieving budget-balance with vickrey-based payment schemes in combinatorial exchanges. In *IBM Research Technical Report 22218*, 2001.
- [12] J. Stewart. *BGP4: Interdomain Routing in the Internet*. Addison-Wesley, 1998.
- [13] M. Yokoo, Y. Sakurai, and S. Matsubara. Robust combinatorial auction protocol against false-name bids. In *Artificial Intelligence*, volume 130, pages 167–181, 2001.
- [14] M. Yokoo, Y. Sakurai, and S. Matsubara. The effect of false-name bids in combinatorial auctions: New fraud in internet auctions, 2003.