The Strategic Justification for BGP

Levin, Hagay; Schapira, Michael and Zohar, Aviv
The Hebrew University of Jerusalem

20. November 2006

The Strategic Justification for BGP (Working Paper)

Hagay Levin* Michael Schapira† Aviv Zohar‡

Abstract

The Internet consists of many administrative domains, or *Autonomous Systems* (ASes), each owned by an economic entity (Microsoft, AT&T, The Hebrew University, etc.). The task of ensuring connectivity between ASes, known as *interdomain routing*, is currently handled by the *Border Gateway Protocol* (BGP).

ASes are self-interested and might be willing to manipulate BGP for their benefit. In this paper we present the strategic justification for using BGP for interdomain routing in today's Internet: We show that, in the realistic Gao-Rexford setting, BGP is immune to almost all forms of rational manipulation by single ASes, and can easily be made immune to all such manipulations. The Gao-Rexford setting is said to accurately depict the current commercial relations between ASes in the Internet. Formally, we model interdomain routing as a game and prove that a slight modification of BGP is incentive-compatible in *ex-post Nash equilibrium*. Moreover, we show that, if a certain reasonable condition holds, then this slightly modified BGP is also *collusion-proof* in ex-post Nash – i.e., immune to rational manipulations even by *coalitions* of *any* size. Unlike most previous works on achieving incentive-compatibility in interdomain routing, our results *do not require any monetary transfer between ASes* (as is the case in practice). Our results help explain why BGP is, in practice, resilient to rational manipulation (even without changes).

We also weaken the Gao-Rexford constraints by proving that one of the three constraints can actually be enforced by the rationality of ASes if the other two constraints hold.

1 Introduction

The Internet is composed of smaller networks called *Autonomous Systems* (ASes). The term *interdomain routing* refers to the computational transfer of information (packets) between these ASes. As connectivity is the raison d'etre of the Internet, interdomain routing is vital to its functioning. The only standard protocol currently used for interdomain routing is the *Border Gateway Protocol* (BGP). BGP has many desirable qualities from an engineering, or networking, perspective (low "network complexity", quick adaptation to changes in the topology of the network, etc.). One important such quality is that BGP allows ASes to express rich *routing policies*[1] in a distributed manner. I.e., ASes can make local decisions, based on private preferences over routes, without global coordination. However, as first observed in [23], this property of BGP entails a disadvantage: The lack of coordination between local routing policies may result in anomalies that range from local route oscillations to overall protocol divergence. A great deal of networking research has addressed the problem of identifying sufficient conditions for *BGP convergence*(see, e.g., [14, 15, 23, 12, 11, 13, 17, 4, 21]. In a seminal paper, Gao and Rexford [12] proposed realistic constraints on routing policies that lead to route stability without global coordination. The Gao-Rexford setting is said to accurately describe today's commercial Internet [16].

However, the lack of coordination between routing policies is not the only risk to BGP performance. ASes are owned by competing economic agents that are interested in maximizing their personal gain, even

*The Department of Economics, The Hebrew University of Jerusalem, Israel. hagayl@mscc.huji.ac.il.

†The School of Computer Science and Engineering, The Hebrew University of Jerusalem, Israel. mikesch@cs.huji.ac.il. Supported by grants from the Israel Science Foundation and the USA-Israel Bi-national Science Foundation.

‡The School of Computer Science and Engineering, The Hebrew University of Jerusalem, Israel. avivz@cs.huji.ac.il. Supported by a grant from the Israel Science Foundation.

[1] A formal definition of interdomain routing, from both networking and game-theoretic perspectives is presented in Section 2.

if this comes at the expense of the global routing outcome (for instance, causing remote route oscillations that do not affect them). ASes might therefore be willing to try to manipulate the protocol if this serves their purposes. Naturally, it is interesting to seek cases in which BGP is *incentive-compatible*, meaning that no AS has an incentive not to execute BGP. This is considered a central problem in *distributed algorithmic mechanism design* (DAMD) – an area of research that combines Computer Science and Game Theoretic concepts and tools. It is very natural to approach interdomain routing from the DAMD perspective since computation in the Internet is inherently distributed among independent computational nodes, and since its functioning, and its very existence, are due to the interaction of independent economic entities.

The results presented in this paper can be viewed as the strategic justification for using BGP for interdomain routing in practice. We consider the realistic Gao-Rexford setting[2]. Informally, we demonstrate that, in this setting, BGP (as is) is immune to all forms of rational manipulation but one (lying about the availability of routes). This implies that BGP can easily be made incentive-compatible by denying ASes that form of rational manipulation. Thus, we show that a slightly modified BGP is incentive-compatible. Lying about the availability of routes to an AS's advantage is a complicated matter – it requires knowledge on the routing policies of other ASes and on the topology of the network, and it requires the lying AS to deny other ASes from tracing routes (manipulating TCP/IP and not only BGP). Therefore, our results help explain why BGP is, in practice, resilient to rational manipulation, even without changes.

Our results weaken the Gao-Rexford constraints, as they imply that one of the three Gao-Rexford constraints can actually be enforced by the rationality of the nodes if the other two constraints hold (and therefore need not be assumed). Thus, we show that two natural economic assumptions, that are said to hold in today's Internet, imply both BGP convergence and incentive-compatibility. Unlike most previous works on incentive-compatible interdomain routing (e.g., [6, 8, 5, 7]), our results *do not involve any monetary transfer between ASes* (as is the case in practice).

Formally, our main result is the following theorem:

Theorem: *If the Gao-Rexford constraints hold then BGP with route verification is incentive-compatible in ex-post Nash equilibrium.*

Route verification means that it is possible for an AS that announces an available route to another AS to prove that this route is indeed available to it. If an AS is unable to verify the availability of a route to its neighbouring AS it will simply treat that announcement as one indicating that the announcing AS has no available routes. As mentioned before, rationally lying about the availability of a route is a very complicated matter in practice. Hence, it is reasonable to assume that ASes do not lie about the availability of routes (thus ensuring route verification). However, if one wishes to enforce route verification via computational means, this can easily be done, for instance, via cryptographic signing of ASes on the announced route. We stress that even though route verification prevents ASes from lying about availability of routes it leaves them with many other possible forms of rational manipulation: ASes can lie about their preferences, they can deny routes from other ASes (by not announcing them), they can send contradictory messages to different ASes, they can act inconsistently (e.g., declare that they prefer one route over another at one point in time and the opposite at another), they can announce that they are sending traffic along one route (as long as it is available to them) while sending traffic along another, etc. However, we prove that, if route verification is possible, these forms of rational manipulation cannot aid an AS in improving its routing outcomes by unilaterally deviating from BGP.

Moreover, we show that if ASes can verify the availability of routes even if many dishonest ASes exist, a property we call *strong route verification*, then BGP is *collusion-proof ex-post Nash equilibrium*. This means that even a group of ASes, of *any* size, collaborating to better their routing outcomes, cannot do so without harming at least one AS in the group (compared to its outcome if they all executed BGP). We demonstrate the necessity of route verification by providing an example that shows that BGP is *not* incentive-compatible

[2]Unlike previous works, we do not attempt to optimize a global quantifiable goal (social welfare maximization, cost minimization, etc.). Feigenbaum, Ramachandran, and Schapira [7] proved that if one only assumes the Gao-Rexford constraints, and makes no other assumptions (e.g., policy consistency), then BGP might always converge to a solution that is arbitrarily far from optimum, with respect to social-welfare maximization.

without route verification, even if the Gao-Rexford constraints hold.

In Section 2 we present a game-theoretic modeling of interdomain routing. To the best of our knowledge, this is the first such model that captures the intricacies of real-life interdomain routing (its asynchronous nature, distributed computation, lack of payments per packets, partial information, sequential interaction). In particular, we explain the game-theoretic solution concepts we adopt including *ex-post Nash equilibrium*[3]. This solution concept was suggested by Parkes and Shneidman [20] as a suitable game-theoretic solution concept for distributed settings such as ours. Ex-post Nash is a robust solution concept. No assumptions whatsoever are made on the knowledge ASes have on the routing policies of other ASes or on the topology of the network. This should be contrasted with the stronger knowledge assumption required when aiming for the more standard *Nash equilibrium*. In the interdomain routing setting, aiming for a Nash equilibrium means assuming that ASes are familiar with the routing policies of all other ASes – clearly, an unrealistic assumption.

1.1 Related Work

A great deal of networking research has addressed the problem of identifying sufficient conditions for BGP convergence (see, e.g., [14, 15, 23, 12, 11, 13, 17, 4, 21]. In a landmark paper, Griffin, Shepherd, and Wilfong [14] presented such a condition, namely *no dispute wheel*. This condition is, to date, the most general condition known to guarantee BGP convergence. Subsequently, in a seminal paper, Gao and Rexford [12] proposed realistic constraints on routing policies that lead to route stability without global coordination. The *Gao-Rexford constraints* were later shown by Griffin, Gao, and Rexford [11] to be a special case of the no dispute wheel condition. In the Gao-Rexford setting there are two types of business relationships between ASes: A *customer-provider* relationship in which an AS purchases connectivity from another AS, and *peering*, in which ASes find it mutually advantageous to exchange traffic for free among their respective customers. An AS can participate in many such relationships – it can be a customer to some ASes, a provider to other ASes, and a peer to yet other ASes. These business relationships are based on agreements that are assumed to be long-term contracts formed because of various factors (e.g., the traffic patterns between ASes). The Gao-Rexford setting is said to accurately describe today's commercial Internet [16]. These business relationships naturally induce constraints on the routing policies of the ASes, formally stated by Gao and Rexford (see Subsection 2.2).

Interdomain routing was also approached from the distributed algorithmic mechanism design (DAMD) perspective. DAMD, a term coined by Feigenbaum and Shenker [10], is a sub-field of *algorithmic mechanism design*, an area of research presented in the seminal paper by Nisan and Ronen [19]. DAMD deals with the design of feasible *distributed* algorithms (protocols) for economic settings, using both computational and game-theoretic tools (see [10] for further explanations on DAMD). The DAMD approach to interdomain routing was initiated by Feigenbaum, Papadimitriou, Sami, and Shenker [6] and has been the subject of several works since [5, 8, 7, 9, 18]. Feigenbaum et al. [6] presented a *BGP-based* protocol for the case of *lowest-cost routing*. The incentive-compatibility of this protocol was guaranteed by VCG payments to ASes in order to compensate them for the costs they incurred for transmitting packets. However, as noted by the authors themselves, lowest-cost routing is not used in practice. Subsequent works on the design of BGP-based incentive-compatible protocols for more realistic settings were mostly discouraging – impossibility results were obtained for several cases (general policy routing [8], subjective-cost policy routing [5], forbidden-set policy routing [5], and next-hop policy routing [8]). Feigenbaum, Ramachandran, and Schapira [7] presented a BGP-based protocol and proved that it is incentive-compatible if the Gao-Rexford constraints, and an additional constraint called *policy consistency*, hold. As in [6], the incentive-compatibility of the protocol in [7] is obtained via VCG payments. As observed in [9], the results of Feigenbaum, Ramachandran, and Schapira [7] imply that BGP is incentive-compatible if the Gao-Rexford constraints and policy consistency hold, even without monetary transfer. That is, one can achieve incentive-compatibility simply by running BGP (without VCG-payments). However, policy consistency is a very strong assumption on routing policies[4].

[3] Actually, all the results presented in this paper are stronger, in the sense that they are subgame perfect. See explanation in Section 2.

[4] The result in [7] is closely related to the work of Sobrinho on routing algebras [21]

Informally, it dictates that it cannot be that an AS prefer one route to another, but a neighbouring AS that can forward traffic to it disagrees. In this paper, we contribute to this ongoing line of research.

1.2 Organization of the Paper

In Section 2 we provide the reader with the necessary explanations about interdomain routing and game theory. In this section, we formally define the interdomain routing problem and give an explanation of the way BGP works (by describing an abstract family of routing protocols to which BGP belongs, namely *path vector protocols*). We also explain the Gao-Rexford constraints on routing policies. Finally, we present a game-theoretic modeling of interdomain routing. In particular, we discuss the concept of ex-post Nash equilibrium and why this equilibrium notion is highly suitable for interdomain routing.

In Section 3 we present two negative results: We show that any "reasonable" protocol that deterministically chooses a routing tree for *all* possible routing policies of ASes cannot be incentive-compatible. This provides a strong justification for protocols like BGP that only obtain routing trees in certain cases, but can be incentive-compatible in these cases. We also provide an example in which an AS can better its outcome by not executing BGP even though the Gao-Rexford constraints hold (without any further assumption, or changes to the protocol). This shows that despite the natural economic structure of the Gao-Rexford setting, and the fact that BGP convergence is always guaranteed in this setting, BGP, as is, can still be rationally manipulated by self interested ASes.

In Section 4 we prove our main result: We show that BGP with route-verification is incentive-compatible in ex-post Nash if the Gao-Rexford constraints hold. We also show that, in this setting, BGP with strong route-verification is collusion-proof in ex-post Nash.

2 On Interdomain Routing and Game Theory

2.1 Interdomain Routing and Path-Vector Protocols

The *interdomain routing problem* is defined by an *AS graph* $G = (N, L)$ that represents the network topology. Each node in N represents an AS. N consists of n *source nodes* $\{1, \ldots, n\}$ and a unique *destination node* d[5]. Each edge in L represents a physical communication link between a pair of neighbouring ASes. Let L^i be the set of all simple (noncyclic) routes from i to d. Let S^i be the set of all simple (noncyclic) routes from i to d in the complete graph we get by adding links to G. Each source node i has a private *valuation function* $v_i : S^i \to R_{\geq 0}$ that is its private information[6]. Every valuation function v_i specifies the "monetary value" of each route $R \in S^i$ to node i. We *do not* assume that nodes are familiar with the topology of the network. The valuation functions should be interpreted as representing the ability of nodes to express their preferences over routes, when informed of their existence. We make two assumptions on each valuation function v_i: $v_i(\emptyset) = 0$, that is, no route is worth nothing, and that, for all pairs of routes R_1 and R_2 such that i forwards traffic to two different neighbouring nodes in R_1 and R_2, $v_i(R_1) \neq v_i(R_2)$[7]. The *routing policy* of each node i consists of v_i and of i's *filtering policy*: Preference among the routes made available to i by neighbouring nodes is given by the valuation function v_i, and i's filtering policy determines which routes it makes available to its neighbouring nodes.

A major concern in interdomain routing is to reach a *stable routing tree*. A *routing tree* is an assignment of routes to source-nodes (each node $i \in [n]$ is allocated a route $R_i \in L^i$) such that the allocated routes form a confluent tree to the destination d. A routing tree $T_d = \{R_1, ..., R_n\}$ is said to be *stable* if $v_i(R_i) \geq$

[5]This formulation makes sense because today's interdomain routing computes routes for each destination AS independently

[6]The valuation functions are defined over the complete graph, so that we do not need to assume that ASes are familiar with the topology of the network.

[7]This standard assumption is consistent with current interdomain routing: Because at most one route can be installed in a router's forwarding table to each destination, nodes have some deterministic way to break ties, *e.g.*, based on the next hop's IP address. So, valuations can be adjusted accordingly to match this. However, because only one route per neighbour is considered at a time, ties in valuation are permitted for routes through the same neighbouring node.

$v_i((i,j)R_j)$[8] for every node j that is i's neighbour in G.

The Border Gateway Protocol (BGP) is the only standard protocol for interdomain routing in today's Internet. BGP belongs to an abstract family of routing protocols named path-vector protocols defined in [14]. Path-vector protocols compute routes to every destination AS independently. Basically, the routing tree to a given destination is built, hop-by-hop, as knowledge of how to reach that destination propagates through the network. The process is initialized when d announces itself to its neighbours by sending update messages. From this moment forth, every node establishes a route to d by repeatedly choosing a best route from the routes announced by its neighbours and announcing this route to all its neighbours. This process continues until a stable routing tree is reached. We assume that the network is asynchronous. So, it is possible that the network delays the arrival of update messages along selective links.

Path-vector routing has many desirable qualities from a networking perspective: It is distributed and adapts quickly to changes in the topology of the network (e.g., node or link failures). It allows nodes to express semantically rich routing policies without global coordination. It is, in general, communication- ,and space-efficient. It enforces the tree structure of the route allocation reached by the protocol (a node only forwards traffic to one neighbouring node, and nodes check for loops and remove them from consideration). Finally, it assumes no a-priori knowledge of the nodes on the topology of the network.

However, path-vector routing also comes with a great disadvantage: The lack of coordination between local routing policies means that the interdomain routing process could go on indefinitely. There is an inherent trade-off between achieving the desired autonomy and policy expressiveness at a local level and *protocol convergence* at the global level. Networking researchers seek conditions that guarantee protocol convergence for *every* possible timing of update messages along the network (it is possible for protocols to converge for some timings and diverge for others). This strong requirement is called *protocol safety*. Furthermore, they wish protocols to be safe even in the presence of topology changes such as link or node failures. For a more detailed explanation the reader is referred to the work of Griffin, Shepherd, and Wilfong [14].

2.2 The Gao-Rexford Constraints

Studies of the commercial Internet [16] suggest two types of business relationships that characterize AS interconnections: Pairs of neighbouring ASes have either a *customer-provider* or a *peering* relationship. Customer ASes pay their provider ASes for connectivity – access to Internet destinations through the provider's links and advertisement of customer destinations to the rest of the Internet. Peers are ASes that find it mutually advantageous to exchange traffic for free among their respective customers, *e.g.*, to shortcut routes through providers. An AS can be in many different relationships simultaneously: It can be a customer of one or more ASes, a provider to others, and a peer to yet other ASes. These agreements are assumed to be longer-term contracts that are formed because of various factors, e.g., the traffic pattern between two nodes.

In a seminal paper Gao and Rexford [12] suggest constraints on routing policies that ensure BGP safety, even in the presence of link or node failures. These constraints are naturally induced by the business relationships between ASes.

No customer-provider cycles: Let G_{CP} be the directed graph with the same set of nodes as G and with a directed edge from every customer to its direct provider. We require that there be no directed cycles in this graph. This requirement has a natural economic justification as it means that no AS is indirectly its own provider.

Prefer customers to peers and peers to providers: *customer route* is a route in which the *next-hop* AS (the first AS to which packets are forwarded on that route) is a customer. *Provider* and *peer routes* are defined similarly. We require that nodes always prefer customer routes over peer routes, which are in turn preferred to provider routes. This constraint is on the valuation functions of the nodes – it demands every node assign customer routes higher values than peer routes, which should be valued higher than provider routes.

[8] $v_i((i,j)R_j)$ is the route that has the link (i,j) as a first link and then follows R_j to the destination node d.

Provide transit services only to customers: Nodes do not always carry *transit traffic*—traffic that originates and terminates at hosts outside the node. ASes are obligated (by financial agreements) to carry transit traffic to and from their customers. However, ASes do not carry transit traffic between their providers and peers. Therefore, ASes should share only customer routes with their providers and peers but should share *all* of their routes with their customers. This constraint is on the filtering policy of the nodes – it requires that nodes only export peer and provider routes to their customers (customer routes are exported to all neighbouring nodes).

2.3 Interdomain Routing as a Game

Interdomain routing can be regarded as a game. The players in this game are the source-nodes, located on the AS graph. The *type* of every player is defined by his valuation function. The *interdomain routing game* is a *multi-round game*, with an infinite number of rounds. In each round of the game players make decisions based on their types and the information they learnt in the previous rounds about other players. The interdomain routing game is *asynchronous* – the asynchronous nature of the game is intruduced via an entity we shall call the *scheduler*. The scheduler decides which players are to participate in each round of the game (not necessarily all players play in all rounds). The *schedule* chosen must be such that every player plays in an infinite number of rounds.

In each round of the game, a player chosen to play in that round goes over messages from neighbouring players, each describing a simple route from each such player to the destination node[9] That player can then choose which of his neighbouring players he wishes to forward traffic to. It can also send messages to each of its neighbours describing simple routes from himself to the destination[10] (it may announce different routes to different neighbours). The scheduler delays messages along selective links. That is, it can choose in which round a sent message will reach its destination. However, it cannot prevent messages from eventually reaching their destinations. The choices made by a player in the game are dictated by his *strategy*. In networking terminology, this strategy is the local routing protocol this player decides to execute (and the *strategy space* of a player contains all possible routing protocols). We define a *strategy profile* to be a set of player's strategies.

The interdomain routing game is a *partial-information game*. Players do not have full knowledge of the game. In particular, we *do not* assume that players have a-priori knowledge about the topology of the network, the valuation functions of other nodes, or the scheduling decisions made by the scheduler. In fact, beside their private types, the only knowledge players have is the identities of their neighbouring players.

We say a route is *stable* if, from some round in the game onwards, every player on that route always chooses to forward his traffic to the next player on that route. The gain of a player in the interdomain routing game is defined as follows: If player's route is unstable then the player's gain is 0[11]. Otherwise, the player's gain is his value for the route along which his traffic is forwarded (even if the player is under the impression that its traffic is forwarded along a different route).

We wish to prove that a slightly modified BGP (such that route verification holds) is incentive-compatible. That is, we wish to show that the strategy profile in which the strategy of every player is executing a slightly modified BGP is such that no node has the motivation to deviate from the strategy profile. To prove this we must clarify the game-theoretic solution concept (equilibrium notion) we adopt. It has been argued by Parkes and Shneidman [20] that *ex-post Nash* is a suitable solution concept for partial-information distributed settings such as ours. Incentive-compatibility in ex-post Nash means that a node cannot gain from unilateral deviation from the suggested strategy profile (in our case, executing the slightly modified BGP). Ex-post Nash can be regarded as the middle ground between strategyproofness (incentive-compatibility in a *dominant strategy equilibrium*) and the *standard Nash equilibrium*. As explained by Shneidman and Parkes [20], since

[9] Of course, in practice players can choose to send any kind of message they like. However, we assume that players ignore messages that do not describe routes. Hence, to simplify the game we only allow messages that do describe routes.

[10] As before, a player can actually send any kind of message but since we assume any message not describing a simple route will be ignored, we choose not to allow this.

[11] Another possible formulation is to define the gain as the expected value of a player from his oscillating routes. For simplicity, we have not chosen this formulation. However, our results will still hold given such a formulation.

the computation is performed by the strategic agents, aiming for strategyproofness might be futile. If, for instance, all players but one do not execute the suggested protocol, and run a totally different one, it is very likely that the "faithful" node will suffer for its loyalty to the suggested protocol. The standard Nash equilibrium makes significantly stronger knowledge assumptions than ex-post Nash. In our context, aiming for a Nash equilibrium necessitates the assumption that nodes are familiar with the routing policies of all other nodes. This assumption is clearly unrealistic in interdomain routing. Ex-post Nash does not necessitate knowledge on the network structure and players' types (preferences), or on the chosen schedule of players and messages.

All solution concepts discussed above are susceptible to collusion. That is, while it is true that a unilateral deviation of an AS from BGP is not worthwhile to that AS, a coordinated deviation of several ASes might prove to be beneficial to some, while not harmful to the others. Feigenbaum, Schapira, and Shenker [7] presented the notion of *collusion-proof ex-post Nash equilibrium*. Incentive compatibility in this equilibrium means that deviation from BGP of more than one node cannot strictly improve the routing outcome of even a single node without strictly harming at least one other node in that group (see definition in [9]).

Remark 2.1 *Griffin, Shepherd, and Wilfong [14] defined safety to be BGP convergence for* every *initial allocation of routes and for every possible timing of messages between nodes. The game-theoretic interpretation of this property is that the strategy profile in which every player runs the slightly modified BGP is in equilibrium for every sub-game of the interdomain routing game. Such an equilibrium is called* ***subgame perfect****. We note that since the Gao-Rexford constraints imply BGP safety, this means that we actually prove incentive-compatibility and collusion proofness in subgame perfect equilibriums.*

3 Negative Results

In Subsection 3.1 we give a strong justification for considering protocols like BGP that only produce routing trees in certain cases. Informally, we show that any "reasonable" protocol that deterministically chooses a routing tree for *all* possible routing policies can be manipulated. This impossibility result makes no computational assumptions on routing protocols and therefore holds even for exponential-time protocols.

Gao and Rexford [12] have shown that if the Gao-Rexford constraints hold then BGP safety is guaranteed (assuming all ASes execute BGP). Furthermore, BGP is computationally efficient in this case. However, can BGP be rationally manipulated even in the Gao-Rexford setting? In Subsection 3.2 we demonstrate that it can. We show that BGP, with no additional assumptions or changes to the protocol, is *not* incentive compatible even in this realistic economic setting.

3.1 A General Impossibility Result

We prove the following impossibility result:

Theorem 3.1 *Fix any AS graph. Every protocol that deterministically chooses a routing tree for all possible routing policies of source-nodes on that graph, is incentive-compatible in ex-post Nash, and has at least three possible outcomes, must be dictatorial (i.e, there is some specific AS that is always assigned its most valued route in the routing tree chosen by the protocol).*

Proof:

The proof is a rather straightforward extension of the *Gibbard-Satterthwaite (GS) impossibility theorem* for non-manipulable social choice functions. The GS theorem deals with *voting systems*, in which a single winner is chosen from the preferences of individuals (voters) over candidates. Informally, the GS theorem states that if there are at least three candidates, and the *voting rule* (the way the winner is chosen) is *non-manipulable* then the voting rule must be *dictatorial*. A voting rule is said to be dictatorial if there is some specific voter such that the candidate elected is always his most preferred candidate. A voting rule is manipulable if it is possible for a voter to improve the choice of a candidate (from his perspective) by voting in a manner that does not truly reflect his preferences. It has been shown [22] that the GS theorem holds

even if ties are allowed in the ballots. In this case the winner is chosen from the (possibly more than one) tied candidates at the top of the dictator's ballot.

In our setting, the voters are the ASes and the candidates are all the possible routing trees in the AS graph. The value an AS assigns to a routing tree will simply be the value it assigns to its route in that tree. Thus, the valuation functions of ASes define weak preferences (that allow ties) of the voters (ASes) over the candidates (routing trees). A voting rule, in our case, is a protocol that deterministically chooses a routing tree for every possible set of ASes' preferences. The GS theorem implies that if such a protocol can output more than three routing trees, and is non-manipulable, then it must be dictatorial. I.e., if such a protocol has more than three possible outcomes and is non-manipulable then there is a certain AS such that the protocol always chooses a routing tree that is ranked at the top of that AS's preferences (or one of the tied routing trees, if more than one such tree exists).

Since, this impossibility result relies on the GS theorem, it applies even in a centralized setting in which the only possible form of manipulation available to ASes is reporting false preferences, and all computations are executed by a trusted computational centre. Hence, it also holds in our distributed setting in which many possible forms of rational manipulation exist. Since incentive-compatibility in ex-post Nash of a protocol implies non-manipulability, the theorem follows.

□

3.2 BGP (as is) is Not Incentive-Compatible in the Gao-Rexford Setting

We present an example of a routing instance in which BGP, as is, can be rationally manipulated. This example shows that an AS can improve its routing outcome by unilaterally deviating from BGP. Specifically, a node can benefit from lying about the availability of a route to it.

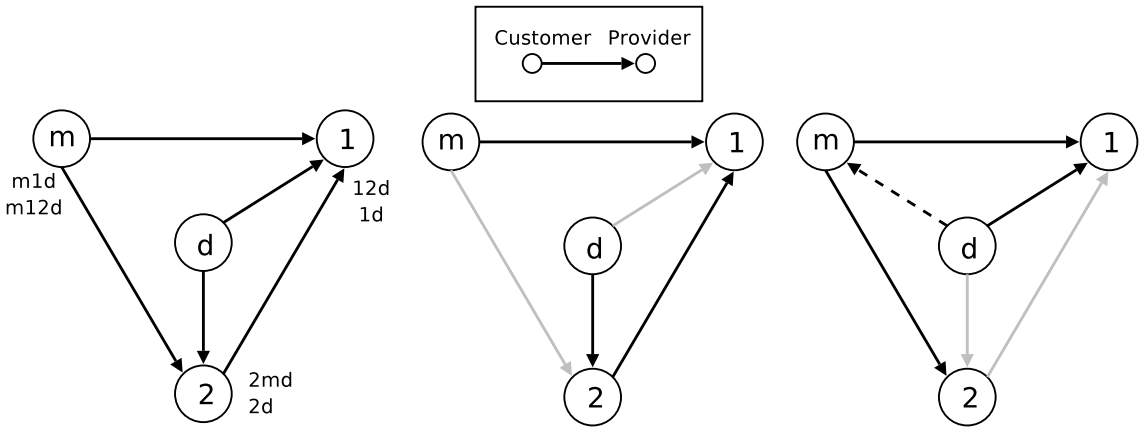


Figure 1: The example. The leftmost figure shows the instance of the interdomain routing problem for which the Gao-Rexford constraints hold. The central figure shows the unique stable solution (the gray lines represent unused links). The rightmost figure depicts the routing tree reached after the manipulation of M (claiming that d is its direct customer). The dotted line represents a nonexistent link.

In our example (see Figure 1) we present an interdomain routing instance with 3 source-nodes. Next to each node in Figure 1 we specify its two most preferred routes, where the upper one is strictly preferred over the lower one (observe, that node 2 would choose to send traffic along a nonexistent route if it were convinced that it is available). All routes not mentioned in Figure 1 should be regarded as very undesirable. Observe, that both the "no customer-provider cycles", and "prefer customers to peers and peers to providers" Gao-Rexford constraints hold. We assume that all nodes but M execute BGP, and uphold the "provide transit services only to customers" constraint. Figure 1 depicts the unique stable routing tree that would be reached if M did not manipulate BGP. In addition, Figure 1 also depicts the routing tree that would be reached

if M informed 2 that the link Md exists, and is available to M, and that d is M's direct customer. M's manipulation affects 2's choice of a route, which in turn affects 1's route in a way that allows M to be assigned its most preferred route. Since M is not assigned this route in the unique stable solution, the manipulation is strictly beneficial for M.

4 BGP with Route Verification is Incentive-Compatible

We have presented an example (see Subsection 3.2) that proves that BGP, as is, is not incentive-compatible even in the Gao-Rexford setting. As we shall show in this section, if the Gao-Rexford constraints hold for all ASes, then BGP is, in fact, immune to all forms of rational manipulation but one (lying about the availability of routes).

In Subsection 4.1 we prove our main result: We show that if the Gao-Rexford constraints hold then BGP with *route verification* is incentive-compatible in ex-post Nash. Route verification means that a node that announces a route to a neighbouring node must be able to prove that this route is indeed available to it. Rationally lying about the availability of routes is a very complicated matter. Hence, it is reasonable to simply assume that route verification holds. However, route verification is also easily achievable via computational means, for instance, via cryptographic signing of the ASes on the announced route. BGP with route verification means that every node executes BGP, with this minor modification: It verifies the availability of any route announced by its neighbours. If a route is not proven to be available, that node will simply consider the announcement as an announcement indicating that the announcing node has no available routes to d. Hence, BGP can easily be made immune to rational manipulation in the Gao-Rexford setting. In fact, as will be discussed in Subsection 4.1, BGP can be modified such that not all the Gao-Rexford constraints need even be assumed.

In Subsection 4.2 we consider a stronger version of route verification – *strong route verification*. Strong route verification means that an AS can verify the availability of a route announced by a neighbouring AS even if *many* other ASes might be attempting to deceive it. We strengthen the result of Subsection 4.1 by proving that BGP with strong route verification is collusion-proof in ex-post Nash (for any number of colluders). I.e., that no group of ASes can join forces to strictly improve the routing outcome of even one AS in the group without strictly harming at least one other AS in the group.

4.1 BGP with Route Verification is Incentive-Compatible in Ex-Post Nash

We prove that BGP with route verification is incentive-compatible if the Gao-Rexford constraints hold. Recall, that the Gao-Rexford constraints are "no customer-provider cycles", "prefer customers to peers and peers to providers", and "provide transit services only to customers" (see Subsection 2.2). The first two constraints are natural economic assumptions on the structure of business relationships between ASes, and on ASes's preferences over routes, respectively. In contrast, the third of these constraints ("provide transit services only to customers") is on the filtering policy of ASes. It is assumed that nodes will filter routes as instructed. Later on, we will show that the "provide transit services only to customers" constraint need not be assumed and can actually be incorporated into the protocol.

The proof of Theorem 4.1 involves a mathematical object named *dispute wheel* defined in [14]. Griffin et al. [14] prove that whenever BGP diverges, the routing policies of the nodes, and the structure of the AS graph, induce a dispute wheel. The absence of a dispute wheel is therefore a sufficient condition for BGP safety. In fact, "no dispute wheel" is the broadest condition known for BGP safety and guarantees safety even in the presence of link or node failures. Moreover, the absence of a dispute wheel is known to imply not only the existence but also the uniqueness of a stable routing tree. It is known that if the Gao-Rexford constraints hold then the routing policies and the AS graph do not induce a dispute wheel [11]. Hence, the Gao-Rexford setting is a special case of "no dispute wheel".

The proof of Theorem 4.1 actually proves a stronger statement: If the routing policies of the nodes do not induce a dispute wheel, then BGP with route verification is incentive-compatible in ex-post Nash. We do so by showing that if unilateral deviation from BGP is strictly beneficial for a node then it must be that the

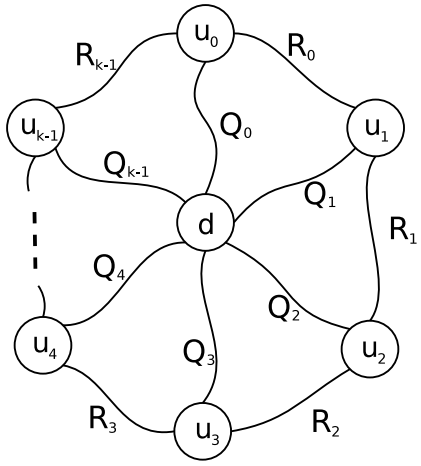
routing policies induce a dispute wheel – a contradiction to the premises of the Gao-Rexford setting. The key idea in the proof is showing that iterative leaps between the unique stable routing tree and the routing tree obtained if a node deviates from BGP and benefits from it, form a special kind of dispute wheel with unique properties. Thus, the proof does not deal with BGP dynamics but only with final routing outcomes.

Theorem 4.1 *If the Gao-Rexford constraints hold, then BGP with route verification is incentive-compatible in ex-post Nash.*

Proof: In order to prove the theorem we shall first define the notion of a *dispute wheel*, as defined in [14]. A dispute wheel is defined as the 3-tuple $(\mathcal{U}, \mathcal{R}, \mathcal{Q})$ where $\mathcal{U} = (u_0, u_1, \ldots, u_{k-1})$ is a sequence of k nodes in the AS graph and $\mathcal{R} = (R_0, R_1, \ldots, R_{k-1})$, $\mathcal{Q} = (Q_0, Q_1, \ldots, Q_{k-1})$ are sequences of routes[12].

- Each route Q_i starts at u_i and ends at the destination node d.
- Each route R_i starts from node u_i and ends in node u_{i+1}.
- $v_i(Q_i) \leq v_i(R_i Q_{i+1})$.[13]

Observe, that a dispute wheel is formed by the combination of the structure of the AS graph and of the valuation functions of the ASes. The term *dispute wheel* is due to the resemblance of its form to that of a wheel. Figure 2 depicts a dispute wheel structure.



Each node u_i would rather route clockwise through node u_{i+1} than through the path Q_i

Figure 2: A Dispute Wheel

We now turn to the proof of the theorem. Observe, that we do not restrict the behaviour of the manipulator in any way. It may deliberately fail to report routes to the destination, fabricate false messages, send different messages to different neighbours, etc.

Consider an instance of the interdomain routing problem such that the AS graph, and the valuation functions of the nodes, do not induce a dispute wheel. From the work of Griffin et al [14] we know that the absence of a dispute wheel implies BGP safety. Moreover, BGP will always converge to a single, unique routing tree. We denote this routing tree by T, and denote the route of every source-node v in T by T_v.

We assume, by contradiction, that the theorem does not hold and that some manipulator node v_m manages to reach a different outcome M by unilaterally diverging from the protocol, and gains by doing so.

[12] In the definitions of the properties the node indices should be considered modulo k, so that node u_k is actually the node u_0. The routes must be such that they were not removed from consideration due to the filtering policies of nodes.

[13] $R_i Q_{i+1}$ is the concatenation of the routes R_i and Q_{i+1}.

We shall show that this gives rise to a contradiction, by proving that if that is the case then there must be a dispute wheel. The proof shall proceed in steps, pointing out a sequence of routes in the graph that will eventually form a dispute wheel. We define the route M_v to be the route node v *believes* it is assigned in M. That is, it could be that the manipulator tricked nodes that send traffic through it in M to believe that their traffic is forwarded along a route not used in practice (see example in Subsection 3.2). Observe, that due to route verification, all routes M_v must exist in the AS graph (the manipulator cannot announce non-existing routes without being detected). We note, that it could be the case that node v_m intentionally causes route oscillations that do not affect it, in order to improve its routing outcome. If this is the case then the route M_v of a node v affected by the route oscillations will simply be one of the oscillating routes.

Since we assumed that v_m gained from its manipulation, we deduce:

$$v_{v_m}(T_{v_m}) < v_{v_m}(M_{v_m}) \tag{1}$$

Because v_m strictly prefers M_{v_m} to T_{v_m}, but did not choose it in the routing tree T, we must conclude that the route M_{v_m} is not available to v_m in T. This means that there must exist some node v (other than v_m) that is on the route M_{v_m} that does not have the same routing in M as it has in T. Let v_1 be the node on the path M_{v_m} that is closest to d on M_{v_m}, such that $M_{v_1} \neq T_{v_1}$.

By definition, all nodes that follow v_1 on the route M_{v_m} have exactly the same routes in T and in M. This means that the node v_1 could choose route M_{v_1} in T. Since it did not choose that route we must conclude that:

$$v_{v_1}(M_{v_1}) < v_{v_1}(T_{v_1})^{14} \tag{2}$$

We can now proceed to the next step in the proof. Since T_{v_1} is preferred by v_1, and was not chosen by v_1 in the routing tree M, it must be that it was not an available option. Therefore, there is some node v on the route T_{v_1}, that is not v_1, such that $T_v \neq M_v$. We select v_2 to be the node v closest to d on the path T_{v_1} for which $T_v \neq M_v$. As before, all nodes that precede v_2 on the route T_{v_1} send traffic along identical routes in both T and M. Hence, the route T_{v_2} must therefore be available to v_2 even in the routing tree M. The fact that it was not chosen in M implies that prefers M_{v_2} over it. Thus, we have that:

$$v_{v_2}(T_{v_2}) < v_{v_2}(M_{v_2}) \tag{3}$$

We can continue these steps, alternating between the routing trees T and M and create a sequence of nodes as follows:

- $v_0 = v_m$

for $n = 0, 1, 2, \ldots$ we perform the following steps:

- **M step**: Let v_{2n+1} be the node v on the route $M_{v_{2n}}$ such that $M_v \neq T_v$, and v is closest among all such nodes to d on $M_{v_{2n}}$.
- **T step**: Let v_{2n+2} be the node v on the route $T_{v_{2n+1}}$ such that $M_v \neq T_v$, and v is closest among all such nodes to d on $T_{v_{2n+1}}$.

Note, that the destination node d cannot appear in this sequence because the route $L_d = T_d$ is the empty path. Due to our construction, and to arguments similar to the ones presented before, the preferences over routes are as follows:
for $i = 0, 2, 4, \ldots$

$$v_{v_i}(T_{v_i}) < v_{v_i}(M_{v_i}) \tag{4}$$

for $i = 1, 3, 5, \ldots$

$$v_{v_i}(M_{v_i}) < v_{v_i}(T_{v_i}) \tag{5}$$

[14]The reason that the inequality is strict is that, as defined in the problem definition, equality exists only if the two routes go through the same neighbouring node. This cannot be the case as $M_{v_1} \neq T_{v_1}$.

Since there is only a finite number of nodes, at some point a node will appear in this sequence for the second time. We denote the first node that appears two times in the sequence by u_0. Let $u_0, ..., u_{k-1}, u_0$ be the subsequence of $v_0, v_1, ...$ that begins in the first appearance of u_0 and ends in its second appearance. We shall examine two distinct cased.

CASE I: The manipulator v_m does not appear in the subsequence $u_0, ..., u_{k-1}, u_0$.

Proposition 4.2 *If for all $i \in \{0, ..., k-1\}$ $v_m \neq u_i$ (the manipulator is not one of the nodes in the subsequence) then k must be even.*

Proof: If k is odd, then it must be that u_{k-1} and u_0 (in its second appearance in the subsequence) were both selected in M steps, or were both selected in T steps. However, if this is the case we reach a contradiction as both nodes were supposed to be the node v closest to d on a certain route, such that $T_v \neq M_v$. Since $u_{k-1} \neq u_0$ this cannot be. □

If k is even then the subsequence of nodes $u_0, ..., u_{k-1}, u_0$, along with the T_{u_i} and M_{u_i} route, and the preferences over these routes (expressed before) form a dispute wheel (for an example see Figure 3).

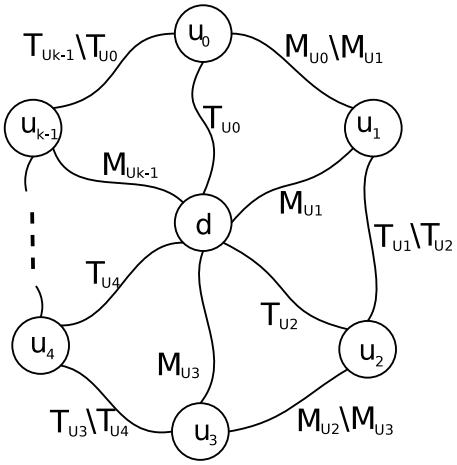


Figure 3: The Dispute wheel constructed during the proof of Theorem 4.1.

CASE II: The manipulator v_m appears in the subsequence (that is, $u_0 = v_m$).

We now need to handle two subcases: The subcase in which k is even and the subcase in which k is odd. If k is even then the second appearance of the manipulator (u_0) in the subsequence is due to a T step. If so, a dispute wheel is formed, as in the example in Figure 3[15].

We are left with the subcase in which k is odd. In this case the second appearance of v_m was chosen in an M step. If so, it must be that $M_{u_{k-1}}$ (that goes through v_m) is not used in practice (otherwise, both u_{k-1} and the second appearance of $v_m = v_0$ were chosen in M steps, and arguments similar to those of Proposition 4.2 would result in a contradiction). This must be the result of a manipulation by v_m. Let L_{v_m} be the false route reported by the manipulator to the node that comes before it on $M_{u_{k-1}}$. Due to route verification L_{v_m} must exist and be available to the manipulator in M. Recall, that the second appearance of the manipulator was chosen due to an M step. Therefore, all nodes that follow it on $M_{u_{k-1}}$ (which are the same nodes as in L_{v_m} are assigned the same routes in T and M. Hence, L_{v_m} was available to v_m in T. It must be that $v_{v_m}(L_{v_m}) \leq v_{v_m}(T_{v_m})$, for otherwise v_m would have chosen L_{v_m} as its route in T (a

[15]The route R
S (where S is a sub-route of R) is route R truncated before the beginning of S

contradiction to the stability of T). We know that $v_{v_m}(T_{v_m}) \leq v_{v_m}(M_{v_m})$ because we assumed that the manipulation performed by v_m were beneficial to it. We get:

$$v_{v_m}(L_{v_m}) \leq v_{v_m}(T_{v_m}) \leq v_{v_m}(M_{v_m}) \tag{6}$$

Thus, we form a dispute wheel with L_{v_m} as shown in Figure 4.

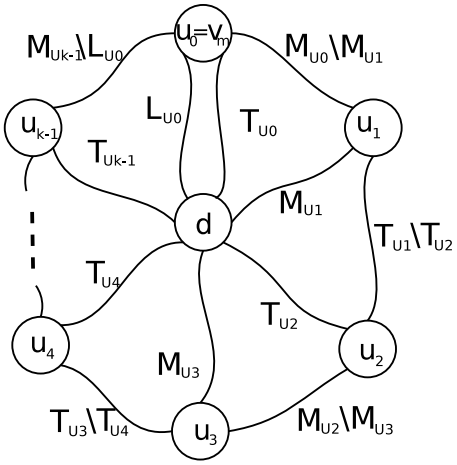


Figure 4: A Dispute wheel formed by an odd number of nodes.

□

As previously mentioned, BGP can be modified so that the "provide transit services only to customers" Gao-Rexford constraint would be rationally enforced by ASes, if the other two natural economic constraints hold. The only extra modification to BGP with route verification is the following: Enabling nodes to verify whether routes announced by their neighbours are customer routes of these neighbours. Once again, this is a reasonable assumption, and is also easily attainable via cryptographic methods (a customer need merely cryptographically sign a confirmation of this fact prior to sending messages). The modified BGP will not only verify routes as before, but also require nodes to perform the following: If a node is announced a route by its customer or peer it must verify that the route is a customer route of the announcing node. If the verification fails the node will consider the announcement as indicating no available routes. We note, that despite this stronger kind of route verification nodes are still left with many possible forms of rational manipulation (lying about preferences, reporting inconsistent information, etc.). Following the proof Theorem 4.1 it can be shown that in this case the modified protocol is incentive-compatible in ex-post Nash if the first two Gao-Rexford constraints hold[16]. This implies that the third Gao-Rexford constraint is enforced by the rationality of the nodes.

4.2 BGP with Strong Route Verification is Collusion-Proof Ex-Post Nash

In this subsection we strengthen the result of Subsection 4.1. We show that if the Gao-Rexford constraints hold, then BGP with strong route verification cannot be rationally manipulated even by coalitions of nodes. Strong route verification is not necessarily easy to obtain via computational means. The problem arises when there is a cluster of dishonest nodes that are not separated by honest ones. Such nodes could be able to simulate each other's actions (e.g., use each other's cryptographic signatures) without it being visible to any node outside the cluster. However, rational lies about the availability of routes are, once again, a complicated matter: They require tight coordination between ASes, knowledge on the preferences of other

[16]If the two first Gao-Rexford constraints hold then it can be shown that, after removing routes that cannot be verified from the AS graph, the routing policies and the AS graph do not induce a dispute wheel.

ASes, preventing ASes from tracing their routes (by manipulating TCP/IP), sharing private information with other ASes, etc.

The proof of Theorem 4.3 is similar to the proof of Theorem 4.1. Again, we actually prove a stronger statement: If the routing policies do not induce a dispute wheel, then BGP with strong route verification is collusion-proof ex-post Nash. The idea at the heart of the proof is showing that even if many dishonest nodes exist, a contradiction in the form of the special dispute wheel presented in the proof of Theorem 4.1 is still reached.

Theorem 4.3 *If the Gao-Rexford constraints hold, then BGP with strong route verification is collusion-proof ex-post Nash.*

Proof: [**Sketch**] We shall assume, by contradiction, that a group of manipulators colludes in an interdomain routing instance with no dispute wheel in order to improve their routing outcomes. We define of T_v and M_v as in the proof of Theorem 4.1. We assume, by contradiction, that all manipulators are not harmed by this manipulation:

$$\forall v \in Manipulators \quad v_v(T_v) \leq v_v(M_v) \tag{7}$$

We shall arrive at a contradiction by showing the existence of a dispute-wheel in a similar manner to that demonstrated in the proof of Theorem 4.1.

We begin the construction by selecting one of the manipulators that strictly gained from the collusion. We shall denote this manipulator by v_m (it must be that $T_{v_m} \neq M_{v_m}$). We then construct a sequence of nodes in a way similar to that explained in the proof of Theorem 4.1:

- $v_0 = v_m$
- **M step**: For a node v_n which is a manipulator we define v_{n+1} to be the node v on the route M_{v_n}, such that $M_v \neq T_v$, and v is the closest to d on M_{v_n} among all such nodes.
- **T step**: For a node v_n that is not a manipulator, and was chosen in an M step, we define v_{n+1} to be the node on the route T_{v_n}, such that $M_v \neq T_v$, and v is the closest to d on T_{v_n} among all such nodes.
- **M step**: For a node v_n that is not a manipulator, and was chosen in a T step, we define v_{n+1} to be the node on M_{v_n}, such that $M_v \neq T_v$, and v is the closest to d on M_{v_n} among all such nodes.

We define a subsequence of nodes $u_0, ...u_{k-1}, u_0$ as in the proof of Theorem 4.1. We now handle two cases. The first case is that no manipulator appears in $u_0, ..., u_{k-1}, u_0$. The handling of this case is precisely the same as in the proof of Theorem 4.1 (Case I).

The other case, is that at least one of the nodes in $u_0, ..., u_{k-1}, u_0$ is a manipulator. First, we prove the following proposition:

Proposition 4.4 *There is no $i \in \{0, ..., k-1\}$ such that both u_i and u_{i+1} (modulo k) are manipulators (no two manipulators come one after the other in the subsequence $u_0, ...u_{k-1}, u_0$).*

Proof: By contradiction, let u_i and u_{i+1} be two consecutive manipulators. u_{i+1} was chosen in an M step. u_{i+1} is therefore the node v closest to d on M_{u_i} such that $M_v \neq T_v$. Hence, $M_{u_{i+1}}$ must be available to u_{i+1} in both M and T. We know that $v_{u_{i+1}}(T_{u_{i+1}}) \leq v_{u_{i+1}}(M_{u_{i+1}})$, as u_{i+1} is a manipulator. Since u_{i+1} chose $T_{u_{i+1}}$ over $M_{u_{i+1}}$ in T it must also be that $v_{u_{i+1}}(T_{u_{i+1}}) \geq v_{u_{i+1}}(M_{u_{i+1}})$. We conclude that $v_{u_{i+1}}(T_{u_{i+1}}) = v_{u_{i+1}}(M_{u_{i+1}})$. However, equality of the values of routes assigned by u_{i+1} is only possible if u_{i+1} forwards traffic to the same node in both routes. Since both routes are available in T, this means that $T_{u_{i+1}} = M_{u_{i+1}}$. This contradicts the reason for which u_{i+1} was selected ($M_{u_{i+1}} \neq T_{u_{i+1}}$). □

The handling of the case in which at least one of the nodes in $u_0, ..., u_{k-1}, u_0$ is a manipulator, is very similar to CASE II in the proof of Theorem 4.1. The tricky part of the proof arises when a manipulator is selected in an M step. Due to Proposition 4.4, it must be that the node that precedes this appearance of the manipulator in the subsequence is not a manipulator. Such an event can be handled as the subcase in which k is odd in CASE II (in the proof of Theorem 4.1). □

5 Open Questions

There are many open question regarding interdomain routing in general, and the game-theoretic approach to interdomain routing in particular. We mention a few:

- The absence of a dispute wheel is, to date, the broadest condition known to guarantee BGP safety [14]. The question of characterizing the settings in which BGP is safe is an interesting, and well studied, open question.
- The former question, is closely related to the question of characterizing the settings in which BGP (with and without route verification) is incentive-compatible. Since per-packet costs might be undesirable in real-life routing, we are especially interested in settings that do not require monetary transfer between ASes.
- In this paper, we did not deal with the way the structure of the AS graph, as well as the business relations between ASes, were formed. Studies of the network and commercial structures of the Internet suggest unique properties of these structures [3, 1, 16]. A game-theoretic approach to this question, is the modeling and analysis of realistic network-formation games (for an example of a network formation game see [2]). In particular, it is interesting to explore the way business relations are formed between ASes.

Acknowledgements

We thank Joan Feigenbaum, Aaron Jaggard, Noam Nisan, Ahuva Mu'alem, Motty Perry, Vijay Ramachandran, Jeff Rosenschein, Rahul Sami, and Scott Shenker for helpful discussions.

References

[1] A.L. Barabasi and R. Albert. Emergence of Scaling in Random Networks. Science, October 1999.

[2] Alex Fabrikant, Ankur Luthra, Elitza Maneva, Christos H. Papadimitriou, and Scott Shenker. On a network creation game. In *Proceedings of PODC 2003*, 2003.

[3] Michalis Faloutsos, Petros Faloutsos, and Christos Faloutsos. On power-law relationships of the internet topology. In *SIGCOMM 1999: Proceedings of the 1999 conference on Applications, technologies, architectures, and protocols for computer communications*. ACM Press, 1999.

[4] Nick Feamster, Ramesh Johari, and Hari Balakrishnan. Implications of autonomy for the expressiveness of policy routing. In *SIGCOMM '05: Proceedings of the 2005 conference on Applications, technologies, architectures, and protocols for computer communications*, New York, NY, USA, 2005. ACM Press.

[5] Joan Feigenbaum, David R. Karger, Vahab S. Mirrokni, and Rahul Sami. Subjective-cost policy routing. In Xiaotie Deng and Yinyu Ye, editors, *First Workshop on Internet and Network Economics*, volume 3828 of *Lecture Notes in Computer Science*, pages 174–183, Berlin, 2005. Springer.

[6] Joan Feigenbaum, Christos H. Papadimitriou, Rahul Sami, and Scott Shenker. A BGP-based mechanism for lowest-cost routing. *Distributed Computing*, 18(1):61–72, 2005.

[7] Joan Feigenbaum, Vijay Ramachandran, and Michael Schapira. Incentive-compatible interdomain routing. In *7th Conference on Electronic Commerce*, pages 130–139, New York, 2006. ACM.

[8] Joan Feigenbaum, Rahul Sami, and Scott Shenker. Mechanism design for policy routing. *Distributed Computing*, 18(4):293–305, 2006.

[9] Joan Feigenbaum, Michael Schapira, and Scott Shenker. *Distributed Algorithmic Mechanism Design: A chapter in the upcoming book "Algorithmic Game Theory"*. Cambridge University Press.

[10] Joan Feigenbaum and Scott Shenker. Distributed algorithmic mechanism design: recent results and future directions. In *6th International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications*, pages 1–13, New York, 2002. ACM.

[11] Lixin Gao, Timothy G. Griffin, and Jennifer Rexford. Inherently safe backup routing with BGP. In *20th INFOCOM*, pages 547–556, Pistacaway, 2001. IEEE.

[12] Lixin Gao and Jennifer Rexford. Stable Internet routing without global coordination. *IEEE/ACM Transactions on Networking*, 9(6):681–692, 2001.

[13] Timothy G. Griffin, Aaron D. Jaggard, and Vijay Ramachandran. Design principles of policy languages for path vector protocols. In *SIGCOMM '03: Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 61–72, New York, 2003. ACM.

[14] Timothy G. Griffin, F. Bruce Shepherd, and Gordon Wilfong. Policy disputes in path-vector protocols. In *7th International Conference on Network Protocols*, pages 21–30, Los Alamitos, 1999. IEEE Computer Society.

[15] Timothy G. Griffin and Gordon Wilfong. A safe path vector protocol. In *Proceedings of IEEE INFOCOM 2000*. IEEE Communications Society, IEEE Press, March 2000.

[16] Geoff Huston. Interconnection, peering, and settlements. In *Internet Global Summit (INET)*. The Internet Society, 1999.

[17] Aaron D. Jaggard and Vijay Ramachandran. Robustness of class-based path-vector systems. In *Proceedings of ICNP'04*, pages 84–93. IEEE Computer Society, IEEE Press, October 2004.

[18] Ahuva Mu'alem and Michael Schapira. Setting lower bounds on truthfulness. In *To appear in SODA 2007*.

[19] Noam Nisan and Amir Ronen. Algorithmic mechanism design. *Games and Economic Behavior*, 35(1):166–196, 2001.

[20] David C. Parkes and Jeffrey Shneidman. Specification faithfulness in networks with rational nodes. In *23rd Symposium on Principles of Distributed Computing*, pages 88–97, New York.

[21] João L. Sobrinho. An algebraic theory of dynamic network routing. *IEEE/ACM Transactions on Networking*, 13(5):1160–1173, 2005.

[22] Alan D. Taylor. The manipulability of voting systems. *The American Mathematical Monthly*, April 2002.

[23] Kannan Varadhan, Ramesh Govindan, and Deborah Estrin. Persistent route oscillations in inter-domain routing. *Computer Networks*, 32(1):1–16, March 2000.