University of New Mexico
UNM Digital Repository

Electrical & Computer Engineering Faculty Publications

Engineering Publications

6-1-2006

Identifier-Based Discovery in Large-Scale Networks

Chaouki T. Abdallah

Joud Khoury

Chapter 15
Identifier-Based Discovery in Large-Scale Networks
An Economic Perspective

Joud Khoury and Chaouki T. Abdallah

Abstract The design of any network mechanism that requires collaboration among selfish agents could only benefit from accounting for the complex social and economic interactions and incentives of the agents using the design. This chapter presents a broad treatment of the main economic issues that arise in the context of identifier-based discovery on large scale networks, particularly on the Internet. An "identified" object (such as a node or service), referred to as a player, demands to be discoverable by the rest of the network on its "identifier". A discovery scheme provides such a service to the players and incurs a cost for doing so. Providing such a service while accounting for the cost and making sure that the incentives of the players are aligned is the general economic problem that we address in this work. After introducing the identifier-based discovery problem, we present a taxonomy of discovery schemes and proposals based on their business model and we pose several questions that are becoming increasingly important as we proceed to design the inter-network of the future. An incentive model for distributed discovery in the context of the Border Gateway Protocol (BGP) and path-vector protocols in general is then presented. We model BGP route distribution and computation using a game in which a BGP speaker advertises its prefix to its direct neighbors promising them a reward for further distributing the route deeper into the network. The neighbors do the same thing with their direct neighbors, and so on. The result of the this cascaded route distribution is a globally advertised prefix and hence discoverability. We present initial results on the existence of equilibria in the game and we motivate our ongoing work.

Joud Khoury · Chaouki T. Abdallah
1 University of New Mexico, Albuquerque NM 87131, USA;
e-mail: {jkhoury, chaouki}@ece.unm.edu

15.1 Introduction

Traditionally, the design process in the context of the Internet has focused on sources of value as they relate to performance, robustness, resilience, reliability, etc. with less emphasis on the socio-economical dynamics that underly the latter. The value of any new design in the new era does not solely depend on performance and must take into account the complex social and economic interactions and incentives of the agents using the design if success is to be reached [25, 37]. Check [37] for an interesting overview of several tools that are important in bridging computer science and economics to better understand the complex socio-economic interactions in the context of the Internet, and [25] for an interesting overview of several of the problems and applications arising at the interface between information and networks.

Almost every networking application relies on discovery and naming (alternatively referred to as identification throughout the discussion since we ignore name semantics) services. An identifier (or alternatively a name) in this context refers to an address that is independent of the network topology but that could nevertheless be routable. Identifier-based discovery (simply referred to as discovery hereafter) is a core network service aimed at discovering a network path to an identified object. Discovery is usually the first step in communication, before a path to the destination object is established. Given an identifier of some object on the network, discovering a path to the object could either utilize mapping/resolution where the identifier is mapped to some locator (see for example [17, 32, 34], and the Domain Name System (DNS)), or it could utilize routing-on-identifiers (see [4, 6, 11, 27] etc.). In either case however, an underlying routing scheme that routes on locators typically exists and is utilized after a path has been discovered for efficient communication. Note that the terms identifier and locator are both addresses at different layers of abstraction. We differentiate the two terms only after we fix an upper layer: an identifier at the upper layer maps into a locator which is an address relative to the upper layer. The locator itself is a path identifier at a lower layer.

In this discussion we assume that a naming or identification system for a large scale network, the Internet mainly, is required given the network's mobile and ubiquitous usage models. For example, on the Internet, this translates into either designing a new system or enhancing the current ones (for example DNS). While there is a rich literature on applying game theory and economics models to Internet games, we find in the networking literature a number of proposals for Internet discovery schemes (and id routing) requiring significant coordination among selfish users while ignoring the economic aspects that may possibly render them infeasible or inefficient (and we shall give several examples of such system or proposals later in Section 15.3). In a future Internet in which domains or Autonomous Systems (ASes) are selfish agents trying to maximize their local utilities, the design of any identifier based discovery scheme could benefit from establishing the right economic models. The problem on the Internet specifically is exacerbated as there are multiple layers of identification managed by different systems, mainly DNS at the application and the Border Gateway Protocol (BGP) [38] at the network layer. Note that in the case of the latter, the Internet Protocol (IP) address space has been aggressively de-aggregated for reasons that we shall discuss shortly.[1] In this sense, the IP address space has been transformed into an identifier space in which the address is almost independent of the topological location especially in the case of Provider Independent (PI) addressing. In this work, we take a first step at designing discovery schemes for a future Internet (check NSF's FIND [2], and the GENI [1] initiatives for current efforts to redesign and implement the future Internet). We introduce two interrelated design goals that we have identified as missing in the current design process, *service differentiation* and *incentives*, and we elaborate on the latter.

Simply stated, a named object (such as a node or service), referred to as a player, demands to be discoverable by the rest of the network. A discovery scheme provides such service to the players. We define the *discovery level* to be a measure of "how discoverable" a player is by the rest of the network (this is "how easy" it is for the network to discover the player not the opposite). The performance of discovery, or the discovery level, could significantly affect the player's business model especially in time-sensitive application contexts. If discovering an object takes a significant time relative to the object's download time, the requesting user's experience suffers. As an example, when no caching is involved, the DNS resolution latency comprises a significant part of the total latency to download a webpage (10–30%) [8, 22]. This overhead becomes more obvious in Content Distribution Networks (CDNs), where content objects are extensively replicated throughout the network closer to the user and the discovery (or resolution) could potentially become the bottleneck. Traditionally, the design of discovery schemes has assumed that all players have the same discovery performance requirements, thus resulting in homogeneous demand. In such a setting, the discovery schemes deliver a discovery service that is oblivious to the actual, possibly heterogeneous, discovery requirements – and valuations – of the different players. In reality however, the CNN site will likely value a higher discovery level more than a generic residential site. An interesting question to ask is therefore the following: should the design of discovery mechanisms account for discovery service differentiation? To further motivate the need for differentiation, we note that on the current Internet, Akamai provides such an expedited resolution service [3]. However, the service which is based on DNS suffers from the same pitfalls of the latter (expensive first lookup and critical dependence on caching) and tightly couples the content distribution provider with the resolution service provider. In a recent work [24], we have presented a first attempt to answer this question by introducing the *multi-level discovery* framework which is concerned with the design of discovery schemes that can provide different service levels to different sets of players.

Obviously, there is a cost associated with being discoverable. This could be the cost of distributing and maintaining information (state) about the identifiers. Accounting for and sharing the cost of discovery is an interesting problem whose absence in current path discovery schemes has led to critical economic and scalability concerns. As an example, the Internet's BGP [38] control plane functionality is oblivious to cost. More clearly, a node (BGP speaker) that advertises a

[1] IP is the network layer protocol in the current Internet that allows interconnecting disparate Internet domains or ASes.

provider-independent prefix (identifier) does not pay for the cost of being discoverable. Such a cost may be large given that the prefix is maintained at every node in the Default Free Zone (DFZ)[2] (the rest of the network pays!). Such incentive mismatch in the current BGP workings is problematic and is further exacerbated by provider-independent addressing, multi-homing, and traffic engineering practices [33]. Notice here that BGP with its control and forwarding planes represents a discovery scheme on prefixes which are technically flat identifiers in a largely de-aggregated namespace. Hereafter, we refer to this form of BGP as BGP-DA for De-Aggregation. Hence, *we conjecture that a discovery scheme should be aware of incentives and cost necessitating that players/nodes pay for the cost of getting the service*.

The rest of the chapter is organized as follows: first we motivate the notion of strategic interactions on networks by presenting three games in Section 15.2 that we shall refer to throughout the discussion. Section 15.3 presents a taxonomy of discovery schemes based on their business models as well as our initial thoughts on suitable economic models for the different discovery models. An incentive model for distributed discovery in the context of BGP and path-vector protocols in general is then presented in Section 15.4 before concluding.

15.2 Networks and Strategic Behavior

Game theory is a fundamental mathematical tool for understanding the strategic interactions among selfish network agents, particularly on the Internet over which autonomous agents (e.g. ASes) interact. The theory provides several solution concepts to help study games that arise in different situations and that have specific requirements and varying underlying assumptions [36]. We overview some basic ones here and we provide examples to illustrate each. The most central and widely applicable solution concept is the *pure strategy* Nash equilibrium (PSNE or NE) which could be simply thought of as a set of strategies that forms a stable solution of the game. A set of strategies for the players is termed a *strategy profile*. Under NE strategy profile, no player can move profitably (i.e. increase her payoff) by deviating from her strategy given every other player's strategy. Despite its wide applicability, the NE solution has several shortcomings in that it may not exist (and hence might require mixing), there could be multiple equilibria, and there is no clear way of how to get to it. In this sense, the *mixed strategy* solution concept was developed by Nash to guarantee that an equilibrium will always exist in the game by mixing the player's strategies (introducing probability distributions over the pure strategies and hence rendering the strategy space a convex set). A more stringent solution concept is the *dominant strategy* solution. Unlike the pure strategy solution, a dominant strategy yields a player the highest payoff independent of the strategies of the rest of the players. Dominant strategies are very attractive solutions when they exist, and when

[2] The DFZ refers to the set of BGP routers in the Internet that do not have any default route as part of their routing table i.e. any such router keeps state about every advertised prefix/destination.

they don't exist game designers might try to design for them. For example, when a player's strategy is to declare some private information that is necessary to the social welfare of the game, an attractive solution would be to make the truthful revelation a dominant strategy hence making sure that the player will never have an incentive to lie. The mechanism design framework [31] provides exactly this solution allowing the mechanism "designer" to achieve a dominant strategy solution (in addition to other design goals). A extension to mechanism design, Algorithmic Mechanism Design (AMD) [35], deals with the computation complexity of the solution and Distributed AMD [15] further considers the "network complexity" in distributed settings. Several other solution concepts exist; however, we will only overview one more which is the *subgame perfect* Nash equilibrium (SPNE) which extends the one-shot NE concept to settings in which players take turns playing (e.g. player 1 plays first, then player 2 plays). In such setting, the subgame perfect NE becomes more "natural" as it captures the order of decision taking. Briefly, a subgame perfect NE is a NE in every subgame of the original game where a subgame could be informally defined as a portion of the game that can be independently analyzed. Note that by the formal definition of a subgame, every game is a subgame of itself and hence every SPNE is necessarily a NE. For formal definitions of the solution concepts and a comprehensive treatment of the topic, we refer the reader to [16].

How does strategy factor into networking problems? To motivate the importance of strategic behavior, we hereby present three networking applications that employ different solution concepts and that we shall refer to throughout the discussion. Our hope is that the games highlight some of the basic economic issues that are of interest to network settings and the tools that are useful in studying these settings. Note that the games we present here might not be straightforward for the unexperienced reader who we refer to [16, 36] for introductory material on the subject. The first application we present is that of "query incentive networks" and is due to Kleinberg and Raghavan [26]. The second application is that of "trading networks with price setting agents" due to Blume et al. [9]. The common aspect of the first two games is that price setting is a strategic behavior of the players which is not the case with the third application we present, "Incentive-compatible interdomain routing" due to Feigenbaum et al. [14]. Additionally, while the first two games are solely interested in studying the equilibria, the third presents a distributed mechanism that achieves the solution.

15.2.1 Nash Equilibria and Query Incentive Networks Game [26]

Query incentives are motivated in peer-to-peer and in social networks where some root node issues a query seeking a piece of information or a service on the network. The seeker does not know which nodes on the network have the answer (neither does any other node) and hence the only way to find the answer is to propagate the query deeper into the network until a node with an answer is reached. In order to do so, every node needs to incentivize its direct children to propagate the query deeper

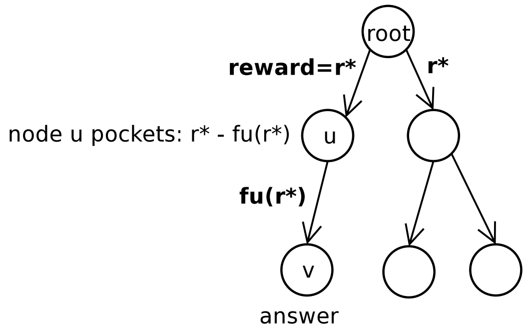

Fig. 15.1 Query Incentive Game: node v has an answer to the query.

where hopefully a destination node with an answer will be reached. Propagation is assumed to occur on a tree and incentives are provided by each parent on the tree to its children in the form of rewards. A node that gets offered a reward will itself offer a smaller reward to its children if its does not posses the answer hence pocketing some reward if an answer to the query is found under the node's subtree. We shall refer to this game hereafter as the QUERY-GAME and we note that this game is based on a similar game initially introduced by Li et al. [30].

Formally, each node (player) u receives a reward r from its parent and offers the same reward $f_u(r) < r$ to its children if it does not have the answer. Otherwise, if u has the answer to the query it responds to its parent with the answer. Each node holds the answer with probability $1 - p$ and on average one in every n nodes holds the answer (n is referred to as the rarity of the answer). The node's strategy is hence $f_u(r)$ which is assumed to be integer-valued and the payoff is simply $(r - f_u(r))\alpha_u(\mathbf{f})$ where $\alpha_u(\mathbf{f})$ is the probability that an answer is found in the subtree rooted at u given that node u has played f_u and every other node's strategy is given by $\mathbf{f} = \{f_v, \forall v\}$ ($\mathbf{f}$ is a *strategy profile*). Figure 15.1 depicts a sample game on a tree.

There are several questions that arise in such a game: how will a node act strategically to tradeoff its payoff and the probability that an answer is found in its subtree knowing that a higher promised reward potentially means higher probability of finding an answer but less payoff? how much initial investment $r*$ is required (as a function of the tree structure and the rarity of the answer n) in order to find an answer with high probability? The authors answer these questions in [26] by modeling a general class of branching processes parametrized on the branching factor b, where the latter is the mean number of active offsprings (or children) per node in the tree constructed using a random branching process [26] (when $b < 1$, the tree is almost surely finite while it is infinite when $b > 1$ with positive probability). When looking for the equilibria, one important point to notice in this game is the interdependency of the players' strategies as given by the tree structure - the strategy of a player will depend on the strategies of its children and so on. The authors show that the Nash equilibrium exists (and is unique with some caveats) by constructing a set of functions **g** (a strategy profile) inductively and showing that the resulting strategy profile is indeed an equilibrium. This result simply says that there exists a stable solution to the game such that if the nodes play the strategies **g** then no node will be able to move profitably given the strategy profile of the rest of the nodes. However, the model does not provide a recipe to get to the solution. Knowing that a solution exists, the next step is to study the breakpoint structure of rewards to be able to say something about the initial investment required (check [26] for results there). In summary, the goal of this game (and the one in [30]) is to provide incentives for query propagation in decentralized networks with complete uncertainty about the destination of the answer knowing that such a process could incur cost that must be paid for by someone to keep the incentives aligned. In the next game, we shall discuss a game the uses the SPNE solution.

15.2.2 Subgame Perfect Nash Equilibria and Trading Networks Game [9]

The next game we present is that of trading networks which despite being more motivated from a markets angle will provide several insights into networking games that involve competition. A set of sellers S wish to sell their goods to a set of buyers B indirectly through a set of traders T. While [9] studies both cases where the goods are distinguishable or not, in this brief overview we shall only focus on indistinguishable goods i.e. a single type of good where all copies are identical. Each seller holds exactly one copy of the good initially and each seller is only interested in buying one copy of the good as well. Trade between the buyers and the sellers can only happen through a set of traders T as specified by a graph G. G specifies how sellers and buyers are connected to the traders where each edge in G connects a node in $B \cup S$ to a node in T. Sellers are assumed to have zero value for the good while each buyer j has a value θ_j for the good. Figure 15.2 depicts such a setting where the indices i, j, t are used to refer to the sellers S, the buyers B, and the traders T, respectively.

We shall refer to this game as the TRADE-GAME. The game aims at studying the process of strategic price setting in markets with intermediaries, and proceeds as follows: first each trader offers a bid price β_{ti} to each seller i to which it is connected, and an ask price α_{tj} to each buyer j to which it is connected. The vector of bid/ask prices is the strategy profile of the traders. Then buyers and sellers choose among the offers they got, the traders pay the sellers the bid price and get the ask price from the buyers. If a trader gets more buyer offers than the seller offers it has, the trader will have to pay a large penalty. This is so that such a scenario will never happen at equilibrium. The payoffs of the different players are as follows: a player that does not participate in a trade gets no payoff. A buyer that participates in a trade through some trader t gets a payoff of $\theta_j - \alpha_{tj}$, while a seller i that participates in a trade with trader t gets a payoff of β_{ti} (again here assuming the seller has no value for keeping the good). Finally, a trader that participates in trade with a set of buyers

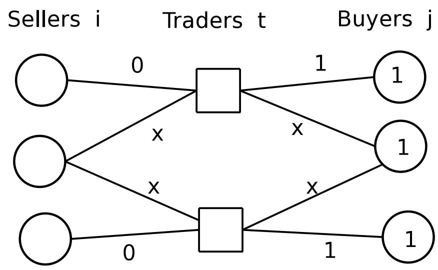

Fig. 15.2 Trading Network Game: sellers S to the left (circles) connect to traders T (squares) who in turn connect to buyers B to the right (circles). The buyers' values are indicated inside the circles (1 in this case). Equilibrium bid and ask prices are shown above the links.

and sellers gets a payoff of $\sum_r(\alpha_{tj_r} - \beta_{ti_r})$ minus a penalty if more buyers than sellers accept its offer (where the index r runs for each distinct buyer, seller combination that have accepted t's offer). It is important to notice that price setting in this game is strategic. Hence, as in the previous game, the first question to ask is how will the traders act strategically to set the market prices knowing that multiple traders could be competing for the same business? and what solution concept is most suitable to studying this game? The solution concept used in this game is the subgame perfect NE which is suitable in such a two stage game where traders play first and then buyers and sellers react. With this in mind, the next step to understanding the strategic behavior of the players (or equivalently the price setting dynamics) is to ask whether a solution (equilibrium) exists and to understand the structure of any such solution. In Figure 15.2, the equilibrium strategies are shown above the links. Two interesting equilibrium phenomena in this game are the effects of monopoly and perfect competition. Both traders in this example make a maximum profit (of 1) from the single monopolized buyer/seller pairs that have access to one trader, while the traders make zero profit when competing for the business of the middle seller and buyer. This must be the case at equilibrium. It turns out as shown by the authors that the equilibrium always exists and that every equilibrium is welfare maximizing (where the welfare of an outcome is simply the difference between the values of the buyers and those of the sellers). These results are shown by resorting to the primal/dual solutions of a welfare maximization linear program. In any solution, no trader will be able to make any profit unless the latter is essential for the social welfare of the game (this result captures the case where traders could have different costs and hence only the cheaper ones will be part of the equilibrium). The game (with distinguishable goods) could be directly extended to account for trading costs i.e. where traders incur costs to perform the trade and the same results hold i.e. a trader will be able to make profit only when the trader is crucial to the social welfare.

15.2.3 Mechanism Design and Interdomain Routing Game [14]

The third game we present in this section is that of interdomain routing incentives, particularly for BGP. First, we briefly overview how BGP operates after which we proceed to describe the incentive mechanism. The Internet is mainly composed of independent Autonomous Systems (ASes), or administrative domains, that must coordinate to implement a distributed routing algorithm that allows packets to be routed between the domains to reach their intended destinations. BGP is a policy-based path vector protocol and is the de-facto protocol for Internet interdomain routing. The protocol's specification [38] was initially intended to empower domains with control over *route selection* (which path or route to pick among multiple advertised routes to a destination), and *route propagation* (who to export the route to among an AS's direct neighbors). The commercialization of the Internet quickly transformed ASes into economic entities that act selfishly when implementing their internal policies and particularly the decisions that relate to route selection and propagation [12]. Intuitively, selfishness and the lack of coordination could potentially lead to instabilities in the outcome of the protocol, as is actually the case with BGP. Griffin et al. have studied this problem and the authors provided the most widely accepted formulation, the stable paths problem, with sufficient conditions under which the protocol converges to a stable solution, the *no dispute wheel* condition [20]. In addition to the algorithmic side of BGP which deals with convergence and stability, recent work has focused on the economic side particularly studying the equilibria of a BGP game and trying to align the incentives of the players (check [14, 29] and references therein).

The interdomain routing incentive game of [14], hereby referred to as ROUTING-GAME, aims to study the policies (strategies) under which BGP is welfare maximizing (i.e. it maximizes the social welfare), and incentive-compatible (i.e. no player has an incentive to deviate from telling the truth where the player's action is to declare private information), and to design a distributed mechanism to provide these attractive properties. Formally, in this game we are given a graph $G = (N, L)$ that represents the AS level topology (nodes N are the ASes and L the set of links between them). The route computation problem is studied for a single destination d and may be directly extended to all destinations assuming route computation is performed independently per destination. Hence, there exists a set of n players indexed by i, and the destination d. Each player has a valuation function $v_i : P^i \to \mathbb{R}$ which assigns a real number to every permitted route to d, P^i being the set of all permitted routes from i to d. Note that a route is permitted if it is not dropped by i and its neighbors. No two paths are assumed to have the same valuation. Social welfare of a particular outcome, an allocation of routes R_i, $\forall i$ that forms a tree T_d, is defined to be $W_{T_d} = \sum_{i=1}^{n} v_i(R_i)$. Clearly, the concept of internal policy is captured with the strict valuation or preference function v_i over the different routes to d which is private information given that the nodes are autonomous. In this sense, and as mentioned earlier, the goal of this problem is to design a mechanism that can maximize the social welfare despite the fact that its components, the v_i functions, are unknown or private. The mechanism design framework and particularly the Vickery–Clark–Groves (VCG) mechanism provides the solution [36]. To do so, a central bank is assumed to exist whose sole task is to allocate a payment $p_i(T_d)$ to each node i based on the outcome. More clearly, a player may either truthfully reveal her valuation to the mechanism (by always picking the best valued routes to d) or not, hoping to manipulate the outcome to her advantage. Based on the players' actions and hence on the outcome tree T_d, a payment $p_i(T_d)$ will be made by the central bank to each player. The utility of each player from an outcome will then be $u_i(T_d) = v_i(R_i) + p_i(T_d)$. The VCG payment scheme is intentionally designed to make the truthful action a *dominant strategy* for all players, hence no player has an incentive to lie about her valuation. To achieve this, AS i will be compensated an amount p_i proportional to the decrease in the value of all upstream ASes that have picked their best route to d through i when the latter does not participate. This is exactly the impact on the social welfare when i is not playing [36]. From a game standpoint, the solution concept that was targeted is the dominant strategy solution - playing truthfully is a dominant strategy and achieving such an attractive solution comes at the expense of assuming a central bank that regulates payments. The authors show that BGP augmented with a VCG payment scheme is incentive-compatible and welfare maximizing in several well studied settings (assumptions on policies or valuation functions).

In the above problem, and generally in problems involving mechanism design, the common scenario is an allocation mechanism that distributes some resource to a set of participating players. In order for a mechanism to implement the Social Choice Function (SCF), for example maximizing the social welfare of all players, the mechanism needs to know the real private information (such as true valuations for example) of the players. This is the case because players might be able to strategically manipulate the output of the mechanism by lying about their private information or strategies. Hence, *dominant strategies* is a desired solution concept that the mechanism would value and would aim towards in such a way so as to implement the SCF.

15.3 Identifier-Based Discovery

We now proceed to formalize the general discovery problem. We start by modeling the network as a graph $G = (V, E)$ with a set of nodes V, $|V| = n$, where each node $u \in V$ can host at most a single default object. The objects residing on the nodes are the players that demand to be discoverable by the rest of the network. The default object is meant to capture the case when the node itself is the object. An object has a unique identifier. Hereafter, we shall refer to the objects by index $i = 1, \ldots, n$ and it should be clear that anytime we refer to node i or player i, we are actually referring to the default object hosted on the node.

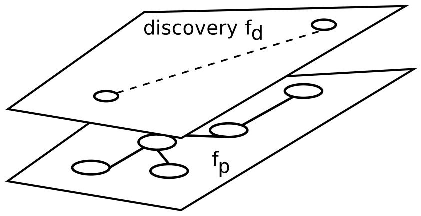

Fig. 15.3 Two layer model.

15.3.1 What Is Discovery?

Recall that an identifier represents an object's identity and remains unchanged when location (e.g. topology) information changes. A locator identifies the location of an object and must change when the object's location information changes.

The notion of discovery throughout this chapter refers to *path discovery based on identifiers*. A discovery scheme f_d generally operates on top of a routing scheme f_p that routes based on locators. We refer to this model as the two-layer discovery model as depicted in Figure 15.3. Whenever f_p exists, all that remains to be discovered by f_d is the identifier-to-locator mapping (e.g. DNS name to IP address mapping). When f_p is not available, then path discovery needs to be performed by f_d as well (e.g. BGP-DA on provider-independent prefixes). The two-layer model may be applied recursively i.e. a new discovery function f_d^* may operate on top of f_d where the latter is virtualized as the locator routing function. This chapter is particularly concerned with the design of mechanisms that implement the discovery function f_d.

Generally, the process of identifier-based path discovery involves a search or discovery query that is forwarded based on a series of calls "forward to *next node* that should have more ($\geq$) information about the named destination" starting at a source node. Discovery schemes in large-scale networks require maintaining distributed state about the identifier space in the upper plane of the two-layer model. Note here that by considering path discovery that involves distributed in-network state, we are clearly restricting the discussion to stateful routing (proactive) schemes which seem to be more common in large-scale networks. Reactive or on-demand discovery schemes generally involve flooding which renders them less efficient to implement at large scales.

From an algorithmic standpoint, a generalized discovery scheme provides the following operations:

- Discovery operations: encapsulate the interface that the players P use to communicate with the mechanism and include two operations:
  - *join(i, level)*: allows player i to request a discovery service possibly expressing a desired service level (and potentially her valuation of some service level).
  - *discover(i, j)*: allows player i to discover player j.

- Service operations: are implemented on the service nodes and dictate a set of rules for maintaining state about the namespace and for handling the above queries.

In general, the discovery scheme utilizes a point-to-point location-based *forward(locators)* operation which could be a simple next-hop forwarding or forwarding to arbitrary locators.

The main reason that we separate the two routing functions f_d and f_p is because there are instances where the two functions are managed by different entities that can minimally collaborate to jointly optimize the two functions. For example, with the current Internet where BGP implements some form of f_p (routing on IP addresses), discovery schemes are being introduced in a separate plane that is not necessarily provisioned by ISPs (players) but rather by other economic entities (as in DNS, and recently [17]). On the other hand, in name-independent compact routing design [6], it is assumed that the two functions are being jointly optimized to achieve a single global goal of efficient communication/discovery. This requirement has motivated us to study discovery mechanisms separately and to deviate from the pure algorithmic treatment of the topic towards *solution concepts* that are based in economics.

15.3.2 Discovery versus Search: Why Receiver-Based Discovery?

In order to frame our work, we introduce the notions of *advertisers* and *seekers*. In identifier-based discovery, advertisers are the entities that wish to be discoverable by the rest of the network using their identifiers. They utilize the *join(i, level)* interface to express their wish to the mechanism. Seekers, who could be advertisers as well, wish to locate the advertisers and they utilize the *discover(i, j)* operation to do so. In our model players are advertisers who may simultaneously be seekers (think of a node in a Distributed Hash Table (DHT) for example as in [40]).

It is important to distinguish two different classes of problems that relate to discovery and that have been considered in the literature. The first, distributed information retrieval, is that of locating information without prior knowledge of the providers or the location of the information (information could be located anywhere in the network). This problem is generally referred to as unstructured search (as in Gnutella, Freenet P2P networks, social networks, etc.). One key idea here is that in order for the requester to find the requested information, she must search for it and be willing to invest in the search. The provider either can not or is not willing to do so. Some prominent work in this vein that addresses cost and incentive structures includes the work by Kleinberg [26].

The second class of problems, which we are more interested in and which we refer to as identifier based discovery, aim at discovering a path to a uniquely identified entity assuming the seeker is given the identifier(s) of the destination beforehand. This problem is common in service centric networks where there generally exists many competing providers for the same service. Within this class of problems, we distinguish two subclasses based on the cost model employed. The first subclass deals with routing problems and focuses on the transit or forwarding cost which is to be bore by the seeker. Several proposals fall under this subclass and many utilize economic tools based in mechanism design [13,35,41]. We distinguish another flavor of the problem by noticing that in service-centric network environments, the seeker gets no utility from the discovery part but rather gets the utility from consuming the service itself. In this sense, the utility of discovery is mainly to the provider or the advertiser: the provider wishes to sell the service and can efficiently do so only when the service is "discoverable". This is the main point that distinguishes our work from the literature on routing and forwarding incentives. The players may be thought of as providers that receive a utility from being discoverable by the rest of the network, the utility of being famous, the latter being inevitably related to the player's business. Hence, in the receiver-based business model, the player does not care about whether other players are discoverable or not, whereas with general P2P resource sharing applications the player's utility is to share the resources of other players and hence to be able to discover the rest of the network (originator-based).

15.3.3 A Taxonomy of Discovery Schemes

Figure 15.4 shows some classic models used by current discovery schemes (and proposals) following the two-layer model. Big circles (light and dark) represent nodes used by f_p at the lower layer (nodes V). At the upper layer, big dark circles represent a subset of those nodes that maintains state about the virtual namespace (service nodes V_D where $V_D \subseteq V$); small dark circles are the objects that wish to be discovered or the players (players P). Figure 15.4 tries to illuminate the relationship between the players P (who receive the discovery service), and the nodes V_D (who provide the service and incur the cost). This relationship is important in an economic setting, such as when studying pricing schemes and when devising a strategic model (and solution concept) for the problem at hand. For example, service nodes in model I (described shortly) may be generally considered to be obedient (i.e. to follow the protocol) as they belong to the same administrative entity (or to multiple competing entities each providing the same service). In models II and III however one needs to consider strategic service nodes in addition to the strategic agents where the two sets could be the same. Some of the representative schemes in the literature that follow these service models are listed in Table 15.1.

In model (I) [$V_D \neq P$], there is a dedicated set of nodes V_D (possibly infrastructure) that keep the state information about the virtual namespace while the players P reside on different nodes. DNS is one example of a centralized scheme that follows this model. In DNS, V_D is the set of root/gTLD (for global Top Level Domain) servers and the players are domain servers that keep zone files. Another scheme that uses this model and that is distributed is the recent DONA proposal [27] where V_D is the set of resolution handlers, and the players are generally objects on edge nodes.

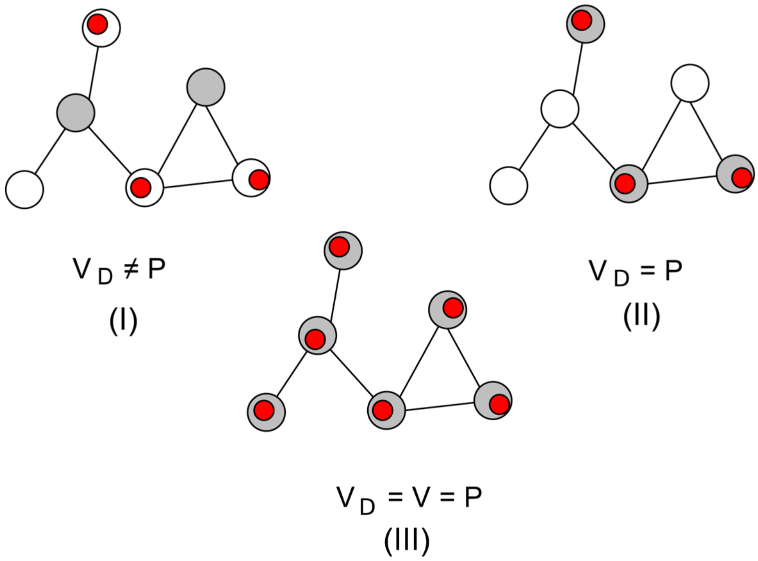

Fig. 15.4 Representation of some common models for discovery.

Table 15.1 Identifier-based discovery schemes.

Model	**Representative Schemes**
Model I	DNS, DONA [27], eFIT [32], LIS ([17], etc.)
Model II	DHTs (Chord [40], etc.)
Model III	NICR ([4,6], etc.), BGP-DA, ROFL [11]

Another set of proposals that fits under this model is embodied by the Locator-ID-Split (LIS) work which aims at providing discoverability to edge sites (e.g. [17]) or nodes (e.g. [34]) in the Internet.

In model (II) $[V_D = P]$, the state is kept on the same set of nodes that the players reside on. In such a model, the players themselves have a common interest in implementing the discovery scheme f_d. The typical example here is Distributed Hash Tables (DHT).

In model (III) $[V_D = V = P]$, the state is maintained on all the nodes V and the players are all the nodes. This model is common to proposals that perform native routing on flat identifiers. One class of schemes that fits under this model is represented by the Name Independent Compact Routing (NICR) [6]. In NICR, the upper and lower layer functions are jointly designed and closely related (more details on NICR later). Another class of schemes that belong to this model does not utilize an underlying f_p i.e. f_d is basically a simultaneous discovery and forwarding scheme. BGP-DA is the representative scheme here where the players are the prefixes advertised by ASes V and where it is necessary for all nodes V to keep the state in order for prefix path discovery (i.e. routing in this case) to succeed. Another recent scheme is the DHT-based ROFL [11], in which the routers are the nodes (if we ignore objects here) that are themselves the players identified by flat identifiers (hashes). Note that models (II) and (III) are the same for our purposes and we shall not make the distinction between the two hereafter.

It is worth noting that each of the schemes in Table 15.1 is designed to satisfy a set of requirements and is based on a set of assumptions about the two-layer functions. Some of the common requirements observed in the literature include *efficiency*, *scalability*, *trust*, *user-control*, *robustness*, *economic requirements*, etc. Some of assumptions address the underlying graph structure (e.g. *scale-free*, or *small-world*) assumptions, or more specific structural assumptions of underlying metric embeddings.

15.3.4 Incentives and Pricing

Having introduced the discovery problem and overviewed different discovery models used in the literature, we now proceed to motivate the need for incentives in discovery. Recall that in order for a node to be discoverable, a cost must be incurred by the set of service nodes V_D generally for maintaining *state* about the node's identifier. The term *state* in this context refers to the information stored on the service nodes to allow the players to be discoverable. The per-node state may be thought of as simply the node's routing table which is generally comprised of mappings from identifier to location information. The question that arises then is who pays for maintaining the state, and what incentive models are suitable for the different discovery models. In this section, we present some solution concepts that are applicable to each of the discovery models, and set the stage for the BGP incentive model which will be discussed in the next section.

15.3.4.1 Model I: $V_D \neq P$

Recall that in this setting, the players P are requesting a discovery service from a set of infrastructure service nodes V_D. When $V_D \neq P$, mechanism design and particularly Distributed Algorithmic Mechanism Design (DAMD) [15] in addition to general cost-sharing models [36] seem to be intuitive frameworks for modeling incentives and pricing. Different situations may arise based on whether the service nodes are obedient or not (obedient service nodes will not try to manipulate the protocol), belong to multiple competing economic entities or not, and on whether the mechanism is subsidized or not. Note that when the mechanism is subsidized, the designer of the mechanism does not have to worry about budget-balance where the latter means that the total payments made by the players must offset the total cost of providing the service.

Assume the service nodes to be obedient and no competition dynamics present, and consider the following DAMD model: each player has a valuation of being discoverable, which she presents to the mechanism. The mechanism logically controls (and operates on) the service nodes collecting all the players' valuations, the demand, and allocating payments back to the players to achieve a mix of goals. These goals could potentially include incentive-compatibility (or strategy-proofness), welfare maximization (or efficiency), and/or budget-balance. When the mechanism is subsidized, the goal of the mechanism is to maximize the social welfare (instead of budget-balance) under the constraint that a cost is associated with providing the service. In this sense, truthful valuations of the service need to be declared by the players, and hence the goal of incentive-compatibility (especially when the mechanism is able to provide different levels of the service). We have presented such a DAMD model that accounts for service differentiation in a recent work [24]. A one-shot VCG variant [36] is a natural solution here that could achieve efficiency and incentive-compatibility again assuming that the mechanism could be subsidized in other ways. The VCG pricing scheme is a cost-sharing scheme i.e. it shares the total cost of providing the service among the participating players. The mechanism will always maximize the social welfare of all the players and will pick prices (cost shares) such that a player i pays an amount equal to the difference in the total welfare of the other players with and without player i's participation - the damage caused by player i's participation.[3] Note that the budget-balance requirement becomes essential when the subsidization assumption does not hold since the total cost must be collected so that service nodes are paid for participating. For example, if a node j is not compensated for the cost of keeping state about the rest of the network, the node will have no incentive to participate. It has been proven by Laffont and Green and later by Satterwaite impossibility theorems [36] that cost-sharing mechanisms can be either strategy-proof and efficient, or strategy-proof and budget-balanced but not both.

When competition among the service providers is present, then the one-shot mechanism design framework seems less practical. This case is more representative of model (I) than the no-competition case. The main idea here is that multiple competing Discovery Service Providers (DSPs) offer the service to the players. Each DSP is assumed to be owned and operated by an autonomous economic entity and DSPs compete for service or market share. Dynamic pricing is more suitable in such a model and a realistic strategic model for this setting based on repeated games was introduced by Afergan [5]. The model discusses price strategies at Internet interchange locations, such as multiple ISPs providing service to a customer (e.g. a CDN). The same model may potentially apply to the discovery mechanism pricing where multiple competing DSPs compete for market share.

15.3.4.2 Models II, III: $V_D = P$

When the set of service nodes $V_D = P$, players incur a cost due to participation of other players and the issue of incentive and pricing becomes even more challenging.

[3] This VCG pricing scheme is referred to as the Clark Pivot rule [36].

In this distributed setting, the traditional game theoretic and economic tools seem to be more applicable, since the centralized designer and the obedient service nodes assumptions inherent to the mechanism design framework no longer hold. Consider BGP for example where every node that wishes to be discoverable introduces state about its identifier on every other node in the DFZ. NICR [4, 6] schemes on the other hand are less costly as they try to optimize the tradeoff between state and *stretch* (stretch is defined in the context of routing as the ratio between the cost of the path taken by the routing scheme, to the minimum cost path where cost could be defined differently based on the setting (e.g. hops or delay); the maximum of the ratio for all source destination pairs is generally referred to as the stretch [6]). In this sense, a node that wishes to be discoverable must introduce state on a subset of other nodes in the network. In both examples above, one can directly recognize the incentive mismatch issue and the challenges inherent to the design of incentive and pricing models that are suitable for this setting. In the next section, we present one such model for BGP.

15.4 An Incentive Model for Route Distribution and Discovery in Path Vector Protocols

The main motivator for devising a model to account for the cost of distribution in BGP is the recent attention in the research community to the incentive mismatch when it comes to the cost of discovery in BGP. Herrin has analyzed in [21] the non-trivial cost of maintaining a BGP route and has highlighted the inherent incentive mismatch in the current BGP system where the rest of the network pays for a node's route advertisement.

BGP is intrinsically about distributing route information to destinations (which are IP prefixes) to establish paths in the network (route distribution and route computation). Path discovery is the outcome of route distribution and route computation. A large body of work has focused on putting the right incentives in place knowing that ASes are economic agents that act selfishly in order to maximize their utilities. In dealing with the incentive problem, previous work has ignored the control plane incentives (route distribution) and focused on the forwarding plane incentives (e.g. transit costs) when trying to compute routes. One possible explanation for this situation is based on the following assumption: a node will have an incentive to distribute routes to destinations since the node will get paid for transiting traffic to these destinations and hence route distribution becomes an artifact of the transit process and is ignored. The majority of previous work that tries to introduce the required incentive models do so by introducing per-packet transit costs. Nodes declare these costs to a mechanism and receive payments from the latter. The mechanism design framework is generally employed here and the mechanism is generally assumed to be subsidized (hence budget-balance is not a design goal). In this work, we conjecture that forwarding is an artifact of route distribution (and definitely computation) where the latter happens first in the process and hence our main focus is on incentivizing nodes to distribute route information. Clearly, we separate the BGP distribution game from the forwarding game and we focus solely on the former. Whether the two games can be combined and studied simultaneously is an open question at this point.

In this section, we synthesize many of the ideas and results from [13, 19, 26, 30] into a coherent model for studying BGP route distribution incentives. The model we employ is influenced by the query propagation model studied by Kleinberg [26] in the context of social networks.

15.4.1 A Simple Distribution Model

A destination d advertises its prefix and wishes to invest some initial amount of money r_d in order to be globally discoverable (or so that the information about d be globally distributed). Since d can distribute its information to its direct neighbors only, d needs to provide incentives to get the information to propagate deeper into the network. d wants to incentivize its neighbors to be distributors of its route who then incentivize their neighbors to be distributors and so on. A transit node i will be rewarded based on the role it plays in the outcome routing tree to d, T_d (whether the outcome is a tree should become clear later in the discussion). The utility of the transit node i from distributing d's route, as we shall describe shortly, increases with the number of nodes that route to d through i – hence the incentive to distribute.

The model seems to correctly capture many of the details behind how policy-based BGP (and in general path-vector protocols) works and the inherent incentives required. Additionally, the model is consistent with the simple path vector formulation introduced by Griffin in [19]. More clearly, it is widely accepted that each AS participating in BGP has as part of its decision space, the following decisions to make:

- import policy: a decision on which routes to d to consider,
- route selection: a decision on what route to d to pick among the multiple possible routes,
- export policy: a decision on who to forward the advertisement to among its direct neighbors.

All three policies are captured in the game model we describe next.

There are two main properties of interest in when it comes to the BGP game model: *convergence*, and *incentives*. The BGP inter-domain routing protocol handles complex interactions between autonomous, competing economic entities that can express local preferences over the different routes. Given the asynchronous interactions among the ASs and the partial information, convergence of BGP to a stable solution becomes an essential property to aim for when studying policies. Griffin et al. [19] defined the stable paths problem which is widely accepted as the general problem that BGP is solving. The authors formulated a general sufficient condition under which the protocol converges to an equilibrium state, mainly the "no dispute wheels" condition. A game theoretic model was recently developed by Levin et al. [29] that enhances the stable paths formalization and studies the incentive-compatibility question. In addition to convergence, incentive issues are crucial to the success and stability of BGP mainly since nodes are assumed to be selfish entities that will act strategically to optimize their utility. In this sense, any distribution and route computation mechanism or policy can only benefit from aligning the incentives of the players to achieve the mechanism's goals [13, 14, 29, 36].

15.4.2 Related Work

The Simple Path Vector Protocol (SPVP) formalism [19] develops sufficient conditions for the outcome of a path vector protocol to be stable. The two main components of the formalism are *permitted paths* and local strict *preference* relations over alternate paths to some destination. A respective game-theoretic model was developed by Levin [29] that captures these conditions in addition to incentives in a game theoretic setting. Other traditional BGP inventive models have not accounted for distribution or discovery costs and incentives and have assumed that every BGP speaker has value in knowing about all destinations and is hence willing to tolerate the cost of such assumption. Our work is fundamentally different than previous models particularly in regard to the incentive structure. The aim in our model is for a destination d to become discoverable by the rest of the network.

Feigenbaum et al. study incentive issues in BGP by considering least cost path (LCP) policies [13] and more general poilicies [14]. The ROUTING-GAME presented in Section 15.2 describes [14]. Our model is fundamentally different from [13] (and other works based in mechanism design) in that the prices are strategic, and it does not assume the existence of a bank (or a central authority) that allocates payments to the players but is rather completely distributed as in real markets. The main element of [14] is payments made by the bank to nodes. The model assumes that the route to d is of value to a source node where the latter will strategically pick among the multiple routes to d. A bank is required to make sure payments are correctly allocated to nodes based on their contribution to the outcome. The bank assumption is troublesome in a distributed setting such as the Internet, and an important question posed in [14] is whether the bank can be eliminated and replaced by direct payments by the nodes.

Li et al. [30] study an incentive model for query relaying in peer-to-peer (p2p) networks based on rewards, on which Kleinberg et al. [26] build to model a more general class of trees. We have introduced the latter model in Section 15.2 with the QUERY-GAME. Both of these models do not account for competition. Similar to the problem setting of [30], an advertiser does not know in advance the full topology neither the resulting outcome of route distribution. Designing payment schemes for such settings generally requires revelation of "non-private value" information such as topology information [39] which might not be available to the players neither to the mechanism designer. The dynamic pricing scheme introduced in [30] avoids such revelation by pricing only based on local information. While we borrow the basic idea from [30] and [26], we address a totally different problem which is that of route distribution versus information seeking.

In addition, our work relates to price determination in network markets with intermediaries (refer to the work by Blume et al. [9] and the references therein). We have introduced the TRADE-GAME of [9] as well in Section 15.2. A main differentiator of this class of work from other work on market pricing is its consideration of intermediaries and the emergence of prices as a result of strategic behavior rather than competitive analysis or truthful mechanisms. Our work specifically involves cascading of traders (or distributors) on complex network structures mainly the Internet.

15.4.3 The General Game

We focus in this work on path-vector protocols and we reuse notation from [14, 30]. We are given a graph $G = (V, E)$ where V consists of a set of n nodes (alternatively termed players, or agents) each identified by a unique index $i = \{1, \ldots, n\}$, and a destination d, and E is the set of edges or links. Without loss of generality, we will study the BGP discovery/route distribution problem for some fixed destination AS with prefix d (as in [14, 19] and [26, 30]). The model is extendable to all possible destinations (BGP speakers) by noticing that route distribution and computation is performed independently per prefix. The destination d is referred to as the *advertiser* and the set of players in the network are termed *seekers* in the discovery model. Two classes of seekers will be distinguished in our model, *distributors* and *retailers*. As the name suggests, distributors actively participate in distributing d's route information to other seeker nodes while retailers simply consume the route (leaf nodes in the outcome distribution tree). For each seeker node j, Let $P(j)$ be the set of all routes to d that are known to j through advertisements, $P(j) \subseteq \mathcal{P}(j)$, the latter being the set of all simple routes from j. The empty route $\phi \in \mathcal{P}(j)$. Denote by $R_j \in P(j)$ a simple route from j to the destination d with $R_j = \phi$ when no route exists at j, and let $(k, j)R_j$ be the route formed by concatenating link (k, j) with R_j, where $(k, j) \in E$. Denote by $B(i)$ the set of direct neighbors of node i and let $next(R_i)$ be the next hop node on the route R_i from i to d. Define node j to be an *upstream* node relative to node i, $j = next(R_i)$. The opposite holds for *downstream* node. Finally, let D_i denote the degree of node i, $D_i \in \mathbb{N}$.

The general discovery game is simple: destination d will first export its prefix (identifier) information to its neighbors promising them a reward r_d ($r_d = 10$ in Figure 15.5) which directly depends on d's utility of being discoverable. A distributor node j (a player) in turn strategizes by selecting a route among the possibly multiple advertised routes to d, and deciding on a reward $r_{jl} < r_{ij}$ to send to each *candidate* neighbor $l \in B(j)$ that it has not received a competing offer from (i.e. s.t. $r_{lj} < r_{jl}$ where $r_{lj} = 0$ means that j did not receive an offer from neighbor l) pocketing the difference $r_{ij} - r_{jl}$. The process repeats up to some depth that is

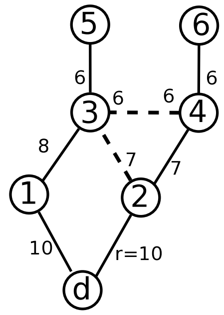

Fig. 15.5 Sample network (not at equilibrium): Solid lines indicate an outcome tree T_d under the advertised rewards.

directly dependent on the initial investment r_d as well as on the strategies of the nodes. A reward r_{ij} that a node i promises to some direct neighbor $j \in B(i)$ is a contract stating that i will pay j an amount that is a function of r_{ij} and of the set of downstream nodes k that decide to route to d through j (i.e. $j \in R_k$ and $R_j = (j, i)R_i$). Note that such a reward model requires that the downstream nodes k notify j of their best route so that the latter can claim its reward from its upstream parent. We intentionally keep this reward model abstract at this point and we shall revisit it later in the discussion when we define more specific utility functions. For example, in Figure 15.5, node d promises $\{1, 2\}$ a reward $r_d = 10$. Node 1 exports route $(1, d)$ to its neighbor promising a reward $r_{13} = 8$. Similarly node 2 exports the route $(2, d)$ to its neighbor set $\{3, 4\}$ with $r_{23} = r_{24} = 7$ and so on. Clearly in this model, we assume that a player can strategize per neighbor, presenting different rewards to different neighbors. We take such assumption based on the autonomous nature of the nodes and the current practice in BGP where policies may differ significantly across neighbors (as with the widely accepted Gao–Rexford policies [18] for example).

15.4.3.1 Assumptions

To keep our model tractable, we take several simplifying assumptions. In particular, we assume that:

1. the graph is at steady state for the duration of the game i.e. we do not consider topology dynamics;
2. the destination d (the advertiser) does not differentiate among the different ASes in the network;
3. the advertised rewards are integers i.e. $r_{ij} \in \mathbb{Z}^+$ and that $r_{ij} < r_{\text{next}(R_i)}$, where the notation $r_{\text{next}(R_i)}$ refers to the reward that the upstream node from i offers to i. A similar assumption was taken in [26] and is important to avoid the degenerate case of never running out of rewards, referred to as “Zeno’s Paradox”;

4. a node that does not participate will have a utility of zero; additionally, when the best strategy of a node results in a utility of zero, we assume that the node will prefer to participate than to default as this could lead to future business opportunities for the node;
5. finally, we only study the game for a class of policies which we refer to as the Highest Reward Policy (HRP) and we accordingly define a utility function for the players. As the name suggests, HRP policies incentivize players to choose the path that promises the highest reward. Such class of policies may be defined general enough to account for complex cost structures as part of the decision space. Despite the fact that the distribution model we devise is general, we assume for the scope of this work that transit costs are extraneous to the model and we refer to resulting preference function as *homogeneous preferences*. This is a restrictive assumption at this point given that BGP allows for arbitrary and complex policies among the players. Such policies are generally modeled with a valuation or preference function that assigns strict preference to the different routes to d. Transit cost is one form of such functions [14], and more complex ones (for example next-hop preferences or metric based preferences) have been studied and modeled [14, 19]. In BGP, such preferences are reflected in contractual agreements between the ASes.

15.4.3.2 Strategy Space, Cost, and Utility

Strategy Space

We now proceed to define the strategy space. Given a set of advertised routes $P(i)$ where each route $R_i \in P(i)$ is associated with a promised reward $r_{\text{next}(R_i)} \in \mathbb{Z}^+$, the strategy $s_i \in S_i$ of an autonomous node i comprises two decisions:

- After receiving update messages from neighboring nodes, pick a single best route $R_i \in P(i)$;
- Pick a reward vector $r_i = [r_{ij}]$ promising a reward r_{ij} to each candidate neighbor j (and export route and respective reward to *all* candidate neighbors).

A strategy profile $\mathbf{s} = (s_1, \ldots, s_n)$ defines an outcome of the game, and a utility function $u_i(\mathbf{s})$ associates every outcome with a real value in $\mathbb{R}$. We shall show shortly that for a certain class of utility functions, every outcome uniquely determines a set of paths to destination d given by $O_d = (R_1, \ldots, R_n)$ and that O_d is always a tree T_d. We use the notation s_{-i} to refer to the strategy profile of all players excluding i. For a given utility function, the Nash equilibrium is defined as follows:

Definition 1 *A Nash Equilibrium is a strategy profile* $\mathbf{s}^* = (s_1^*, \ldots, s_n^*)$ *such that no player can move profitably by changing her strategy, i.e. for each player* i, $u_i(s_i^*, s_{-i}^*) \geq u_i(s_i, s_{-i}^*), \forall s_i \in S_i$.

Cost

There are two classes of cost within the distribution model. The first class defines the cost of participation while the second defines the "per-sale" costs. More clearly, cost of participation is local to the node and includes for example the cost associated with the effort that a node spends in maintaining and distributing the route information to its neighbors. The participation cost may additionally include the amortized per-route operational cost of the hardware (the DFZ router). Herrin [21] estimates this cost to be $0.04 per route/router/year for a total cost of at least $6,200 per year for each advertised route assuming there are around 150,000 DFZ routers that need to be updated. Per-sale costs, on the other hand, are incurred by the node for each sale that it makes and is generally proportional to the number of its downstream nodes in the outcome O_d. As mentioned earlier in the assumptions, we ignore this class of cost in the current model leaving it as part of our future work. Hence, while in general the cost function may be more complicated, we simply assume that the distribution cost c_i is composed of two components: c_i^{dist} representing the distribution cost and is only incurred by the distributors, and c_i^{state} represents the cost of maintaining the state and is incurred by all participating players.

Utility

Earlier in the discussion, we briefly alluded to a rewarding model in which node i rewards a neighbor node j based on some function of r_{ij} and of the set of downstream nodes of j (the latter corresponding to the number of sales node j made). Defining a concrete rewarding function (and hence utility function) for the players is a questions that the game modeler is left with. Specifically, we seek to identify the classes of utility functions and the underlying network structure for which equilibria exist. As a first step, we experiment with a simple class of functions which rewards a node linearly based on the number of sales that the node makes. This first model incentivizes distribution and potentially requires a large initial investment from d. More clearly, define the set $N_i(\mathbf{s}) = \{j \in N \backslash \{i\} | i \in R_j\}$ to be the set of nodes that pick their best route to d going through i (nodes downstream of i) and let $\delta_i(\mathbf{s}) = |N_i(\mathbf{s})|$. Additionally, let $I(x)$ denote the indicator function which evaluates to 1 when $x > 0$ and to 0 otherwise. Thus, $I(\delta_i(\mathbf{s}))$ indicates whether i is a distributor or not. We are now ready to define the utility of a node i from an outcome or strategy profile $\mathbf{s}$ as follows:

$$u_i(\mathbf{s}) = (r_{\text{next}(R_i)} - c_i^{\text{state}}) - c_i^{\text{dist}} I(\delta_i(\mathbf{s})) + \sum_{\{j:i=\text{next}(R_j)\}} (r_{\text{next}(R_i)} - r_{ij})(\delta_j(\mathbf{s}) + 1). \tag{15.1}$$

The first term in the utility function $(r_{\text{next}(R_i)} - c_i^{\text{state}})$ is incurred by every participating node and is the one unit of reward from the upstream parent on the chosen best path minus the local state cost. The second and third terms are only incurred by distributors. The second term $c_i^{\text{dist}} I(\delta_i(\mathbf{s}))$ denotes the distribution cost while the last term given by the summation is the total profit made by i where $(r_{\text{next}(R_i)} - r_{ij})(\delta_j(\mathbf{s}) + 1)$ is i's profit from the sale to j (which depends on the size of j's subtree given by δ_j). We assume here that node i gets no utility from an oscillating route and gets positive utility when R_i is stable. A rational selfish node will always try to maximize its utility by picking $s_i = (R_i, [r_{ij}])$. Equation (15.1) indicates that a node increases its utility linearly in the number of downstream seekers it can recruit, given by the summation. However, to increase the third term given by the summation, node i should carefully pick its rewards r_{ij} given that there might be other nodes competing with i for the route. There is an inherent tradeoff between $(r_{\text{next}(R_i)} - r_{ij})$ and $(\delta_j(\mathbf{s}))$ s.t. $i = next(R_j)$ when trying to maximize the utility in Equation (15.1) in the face of competition as shall become clear shortly. A lower promised reward allows the node to compete but will cut the profit margin. Finally, we assume implicitly that the destination node d gets a fixed incremental utility of r_d for each distinct player that maintains a route to d – the incremental value of being discoverable by any seeker.

15.5 HRP: Convergence, and Equilibria

Before discussing BGP convergence and equilibria under our assumptions and the utility function defined in Equation (15.1), we first prove the following lemma which results in the Highest Reward Path (HRP) policy:

Lemma 1. *In order to maximize its utility, node i must always pick the route R_i with the highest promised reward i.e. $r_{\text{next}(R_i)} \geq r_{\text{next}(R_l)}, \forall R_l \in P(i)$.*

The proof of Lemma 1 is given in Appendix A. The lemma implies that a player could perform her two actions sequentially, by first choosing the highest reward route R_i, then deciding on the reward vector r_{ij} to export to its neighbors. When the rewards are equal however, we assume that a node breaks ties uniformly.

15.5.1 Convergence

A standard model for studying the convergence of BGP protocol dynamics was introduced by Griffin et al. [19] (and later studied by Levin et al. [29]), and assumes BGP is an infinite round game in which a *scheduler* entity decides on which players participate at each round (the *schedule*). Any schedule must be fair allowing each player to play indefinitely and to participate in an infinite number of rounds. Convergence here refers to the convergence of BGP protocol dynamics to a unique outcome tree T_d under some strategy profile $\mathbf{s}$. The "no dispute wheels" condition, introduced by Griffin et al. [19], is the most general condition known to guarantee convergence of possibly "conflicting" BGP policies to a unique stable solution (tree). From Lemma 1, it may easily be shown that "no dispute wheels" exist under

HRP policy i.e. when the nodes choose highest reward path breaking ties uniformly. No dispute wheel can exist under HRP policy simply because any dispute wheel violates the assumption of strictly decreasing rewards on the reward structure induced by the wheel. Hence, the BGP outcome converges to a unique tree T_d [19] under any strategy profile **s**. The tree is stable given **s** which itself is only stable at equilibrium. Note that this is true for every strategy profile (i.e. independent of how the nodes pick their rewards) as long as the strictly decreasing rewards assumption holds, $r_{ij} < r_{\text{next}(R_i)}, \forall i, j$.

Lemma 2 ([20]). *The equilibrium outcome O_d under $\mathbf{s}^*$ is a stable routing tree T_d.*

Having said that, the next set of questions is targeted at finding the equilibrium profile $\mathbf{s}^*$. Particularly, does such an equilibrium exist and under what conditions? Is it unique? And how hard is it to find? In this work, we study the existence of equilibria on special network topologies leaving the other questions for future work.

15.5.2 Equilibria

In the game defined thus far, notice first that every outcome (including the equilibrium) depends on the initial reward/utility r_d of the advertiser as well as on the tie-breaking preferences of the nodes, where both of these are defining properties of the game. Studying the equilibria of the general game for different classes of utility functions and for different underlying graph structures is not an easy problem due to the complexity of the strategic dependencies and the competition dynamics. We are not aware of general equilibria existence results that apply to this game. Hence, we start by studying the game on the simplest possible class of graphs with and without competition. Particularly, we present existence results for the simplest two graphs: 1) the line which involves no competition, and 2) the ring which involves competition. We additionally assume in this discussion that the costs are constant with $c_i^{\text{dist}} = c_i^{\text{state}} = 1$.

Before trying to understand the equilibria of the game on these simple graphs, there is an inherent order of play to capture in the model in order to apply the right solution concept. Recall that the advertiser d starts by advertising itself and promising a reward r_d. The game starts at stage 1 where the direct neighbors of d, i.e. the nodes at distance 1 from d, observe r_d and play simultaneously by picking their rewards while the rest of the nodes "do-nothing". At stage 2, nodes at distance 2 from d observe the stage 1 strategies and then play simultaneously by picking their rewards. At stage 3, nodes at distance 3 from d observe the stage 1 and stage 2 strategies and then play simultaneously and so on. Stages in this multi-stage game with observed actions [16] have no temporal semantics. Rather, they identify the network positions which have strategic significance. The closer a node is to the advertiser, the more power such a node has due to the strictly decreasing rewards assumption. The multi-stage game model seems intuitive based on the assumptions of strictly decreasing rewards and the ability of the node to strategize independently

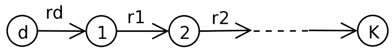

Fig. 15.6 Line netowk: a node's index is the stage at which the node plays; d plays at stage 0.

on each of the downstream links. We resort to the multi-stage model on these simple graphs simply because any equilibrium in the multi-stage game is a stable outcome in BGP no matter how the scheduler schedules the nodes as long as the schedule is fair and infinite.

Formally, each node or player i plays only once at stage $k > 0$ where the latter is the distance from i to d in number of hops; at every other stage the node plays the "do nothing" action. The set of player actions at stage k is the stage-k action profile, denoted by $a^k = (a_1^k, \ldots, a_n^k)$. Further, denote by $h^{k+1} = (a^1, \ldots, a^k)$, the *history* at the end of stage k which is simply the sequence of actions at all previous stages. We let $h^1 = (r_d)$. Finally, $h^{k+1} \subset H^{k+1}$ the latter being the set of all possible stage-k histories. When the game has a finite number of stages, say $K + 1$, then a terminal history h^{K+1} is equivalent to an outcome of the game (which is a tree T_d) and the set of all outcomes is H^{K+1}.

The strategy of node i who plays at stage $k > 0$ is then $s_i : H^k \to \mathbb{R}^{m_i}$ where m_i is the number of node i's direct neighbors at stage $k+1$. It is important to notice in this multi-stage setting that the node's strategy is explicitly defined to account for the order of play given by the graph structure (i.e. a pure strategy of a player is a function of the history). Starting with r_d (which is h^1), it is clear how the game produces actions at every later stage given by the node strategies resulting in a terminal action profile or outcome. Hence an outcome in H^{K+1} may be associated with every strategy profile **s**. We now proceed to study the equilibria on the line and the ring network topologies.

15.5.2.1 No Competition: The Line

In the same spirit as [26] we inductively construct the NE for the line network of Figure 15.6 given the utility function of Equation (15.1). Before proceeding with the construction, notice that for the line network, $m_i = 1$ for all nodes except the leaf node since each of those nodes has a single downstream child. In addition, $\delta_i(\mathbf{s}) = \delta_j(\mathbf{s})+1, \forall i, j$ where j is i's child ($\delta_i = 0$ when i is a leaf). We shall refer to both the node and the stage using the same index since our intention should be clear from the context. For example, the child of node i is $i+1$ and its parent is $i-1$ where node i is the node at stage i. Additionally, without loosing any information, we simply represent the history $h^{k+1} = (r_k)$ for $k > 0$ where r_k is the reward promised by node k (node k's action), and hence the strategy of node k is $s_k(h^k) = s_k(r_{k-1})$.

We construct the equilibrium strategy s^* as follows: first, for all players i, let $s_i^*(x) = 0$ when $x \leq c_i^{\text{state}}$ (where c_i^{state} is assumed to be 1). Then assume that $s_i^*(x)$ is defined for all $x < r$ and for all i. Obviously, with this information, every node i

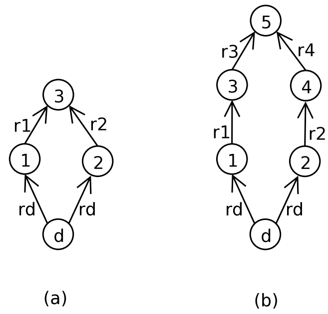

Fig. 15.7 Ring network with even number of nodes: (a) 2-stage game, (b) 3-stage game.

can compute $\delta_i(x, s^*_{-i})$ for all $x < r$. This is simply due to the fact that δ_i depends on the downstream nodes from i who must play an action or reward strictly less than r. Finally, for all i we let $s^*_i(r) = \arg\max_x (r_{i-1} - x)\delta_i(x, s^*_{-i})$ where $x < r$.

Claim 1 *The strategy profile s^* is a Nash equilibrium.*

Proof. The proof of is straightforward: given the utility function defined in Equation (15.1), no node can move profitably. Notice that in general when $r_{\text{next}(R_i)} \leq c^{\text{state}}$, propagation of the reward will stop simply because at equilibrium no node will accept to make negative utility and will prefer to not participate instead (the case with the leaf node). □

Clearly, the NE is not unique since different strategies could result in the same utility. This occurs on the line particularly when a node could get the same utility from being a distributor or not due to the incurred distribution cost. If we assume that a node will always prefer to distribute when the utility is the same, then it can be shown that the NE is unique.

Notice that in the line network, the NE exists for all values of r_d and in any equilibrium node 1 will always be able to make the maximum profit given r_d due to its strategic network position as the first and only distributor.

15.5.2.2 Competition: The Ring

Contrary to the line network, we will present a negative result in this section for the ring network. In a ring, each node has a degree of 2 and $m_i = 1$ again for all nodes except the leaf node. We will consider rings with an even number of nodes due to the direct competition dynamics. In the multi-stage game, after observing r_d, nodes 1 and 2 play simultaneously at stage 1 promising rewards r_1 and r_2 respectively to their downstream children, and so on. Figure 15.7 shows the 2-stage and the 3-stage versions of the game.

For the 2-stage game in Figure 15.7a, it is easy to show that an equilibrium always exists in which $s_1(r_d) = s_2(r_d) = (r_d - 1)$ when $r_d > 1$ and 0 otherwise. This means that node 3 enjoys the benefits of *perfect competition* due to the Bertrand-style competition [16] between nodes 1 and 2 which drives their profit to the minimum possible profit (and hence drives 3's reward to the maximum possible reward). The equilibrium in this game is independent of 3's preference for breaking ties which is not the case with the 3-stage game as we shall show next.

We now present the following negative result when the utility function is given by Equation (15.1):

Claim 2 *For the multi-stage game induced by the ring network, a Nash equilibrium does not always exist, i.e. there exists a value of r_d for which no equilibrium exists.*

Proof. The proof makes use of a counterexample. Particularly, the 3-stage game of Figure 15.7b does not have an equilibrium for $r_d > 5$. This is mainly due to oscillation of the best response dynamics and may be shown by examining the strategic form game, in matrix form, between players 1 and 2. We leave this as an exercise for the interested reader. Briefly, and assuming that node 5 breaks ties by picking route $(5,3)R_3$, $r_d > 5$ signifies the breaking point of equilibrium or the reward at which node 2, when maximizing its utility $(r_d - r_2)\delta_2$, will always oscillate between competing for 5 (large r_2) or not (small r_2). □

Note that under the linear utility given in Equation (15.1), the NE is not guaranteed to exist on the simple ring network. This result is an artifact of the utility function. Finding the classes of utility functions for which equilibria always exist, for the ring network initially and for more general topologies as well, is part of our ongoing work.

15.6 Conclusions and Ongoing Work

In this chapter we have presented a general treatment of the incentive issues that might arise in the context of identifier-based discovery. The BGP incentive model presented has several advantages, mainly providing a dynamic, distributed pricing scheme for route distribution that is partially immune to manipulation and does not require a centralized bank. However, several forms of manipulation may occur in the rewards model. For example, node i may declare a reward $r_i > r_{\text{next}(R_i)}$ when competing on a route to possibly increase its utility, or nodes may lie about the real values of δ when declaring these values to their upstream nodes. Such forms of manipulation may be avoided by route verification and secure cryptographic mechanisms (check secure BGP for example [10]). We have not considered such issues in this chapter. The cascaded rewarding model is similar in spirit to network marketing in economics. One of the main pitfalls of network marketing (alternatively referred to as Multi-Level Marketing or MLM) is that it may put more incentives on recruiting distributors rather than on making a sale. In our model, recruiting a downstream distributor is equivalent to making a sale since a downstream node to i must route to d through i. In addition, the assumption $r_i < r_{\text{next}(R_i)}$ eliminates the pyramid effect common to network marketing schemes.

We are currently working on establishing equilibria results for general classes of utility functions and for general graph structures. The natural next step after that would be to study distributed algorithms that converge to the equilibria, particularly focusing on scalable extensions to BGP [23]. Additionally, we plan to quantify the cost of being discoverable, or in other words the initial investment r_d required by d to guarantee *global reachability* – as a function of the network structure. Interestingly here, for the Internet AS level topology, it was shown by Krioukov et al. [28] that the average distance between any two nodes is small (around 3.5 hops). This property lends itself to the *small world* phenomenon in complex networks [7].

Appendix A: Proof of Lemma 1

Proof. The proof is straightforward. The case for $|B(i)| = 1$ is trivial. The case for $|B(i)| = 2$ is trivial as well since i will not be able to make a sale to the higher reward neighbor by picking the lower reward offer. Assume that node i has more than 2 neighbors and that any two neighbors, say k, l advertise routes $R_k, R_l \in P(i)$ s.t. $k = next(R_k), l = next(R_l)$ and $r_{ki} < r_{li}$, and assume that i's utility for choosing route R_k over R_l either increases or remains the same i.e. $u_i^{R_k} \geq u_i^{R_l}$. We will show by contradiction that neither of these two scenarios could happen.

Scenario 1: $u_i^{R_k} > u_i^{R_l}$

From Equation (15.1), it must be the case that either (case 1) node i was able to make at least one more sale to some neighbor j who would otherwise not buy, or (case 2) some neighbor j who picks $(j, i)R_i$ can strictly increase her $\delta_j(\mathbf{s})$ when i chooses the lower reward path R_k. For case 1, and assuming that r_{ij} is the same when i chooses either route, it is simple to show that we arrive at a contradiction in the case when $j \in \{k, l\}$ (mainly due to the strictly decreasing reward assumption i.e. $r_i < r_{\text{next}(.)}$); and in the case when $j \notin \{k, l\}$, it must be the case that j's utility increases with i's route choice i.e. $u_j^{(j,i)R_k} > u_j^{(j,i)R_l}$. This contradicts with Equation (15.1) since w.r.t. j, both routes have the same next hop node i. The same analogy holds for case 2.

Scenario 2: $u_i^{R_k} = u_i^{R_l}$

Using the same analogy of scenario 1, there must exist at least one neighbor j of i that would buy i's offer only when the latter picks R_k, or otherwise node i will be able to strictly increase its utility by picking R_l pocketing more profit. □

References

1. GENI: Global environment for network innovations. http://www.geni.org/
2. NSF Nets FIND initiative. http://www.nets-find.net/
3. Akamai Technologies. http://www.akamai.com, 2008.
4. I. Abraham, C. Gavoille, D. Malkhi, N. Nisan and M. Thorup, Compact name-independent routing with minimum stretch, in *Proceedings of ACM SPAA '04*, ACM Press, New York, pp. 20–24, 2004.
5. M. Afergan, Using repeated games to design incentive-based routing systems, in *Proceedings of 25th IEEE International Conference on Computer Communications, INFOCOM 2006*, pp. 1–13, 2006.
6. B. Awerbuch, A. Bar-Noy, N. Linial and D. Peleg, Compact distributed data structures for adaptive routing, in *Proceedings of ACM STOC '89*, ACM, New York, pp. 479–489, 1989.
7. A.L. Barabasi, *Linked*, Perseus Publishing, 2002.
8. L. Bent and G.M. Voelker, Whole page performance, Workshop on Web Content Caching and Distribution, Boulder CO, 2002.
9. L. Blume, D. Easley, J. Kleinberg and E. Tardos, Trading networks with price-setting agents, in *EC '07, Proceedings of the 8th ACM Conference on Electronic Commerce*, ACM, New York, pp. 143–151, 2007.
10. K. Butler, T. Farley, P. McDaniel and J. Rexford, A survey of bgp security issues and solutions, Tech. Report, AT&T Labs, 2004.
11. M. Caesar, T. Condie, J. Kannan, K. Lakshminarayanan and I. Stoica, Rofl: routing on flat labels, in *Proceedings of SIGCOMM 2006*, ACM Press, New York, pp. 363–374, 2006.
12. M. Caesar and J. Rexford, Bgp routing policies in isp networks, *IEEE Network* **19**(6), 5–11, 2005.
13. J. Feigenbaum, C. Papadimitriou, R. Sami and S. Shenker, A bgp-based mechanism for lowest-cost routing, *Distrib. Comput.* **18**(1), 61–72, 2005.
14. J. Feigenbaum, V. Ramachandran and M. Schapira, Incentive-compatible interdomain routing, in *EC '06: Proceedings of the 7th ACM conference on Electronic Commerce*, ACM, New York, pp. 130–139, 2006.
15. J. Feigenbaum and S. Shenker, Distributed algorithmic mechanism design: Recent results and future directions, in *Proceedings of the 6th International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications*, ACM Press, pp. 1–13, 2002.
16. D. Fudenberg and J. Tirole, *Game Theory*, MIT Press, 1991.
17. V. Fuller, D. Meyer and D. Farinacci, Lisp alternative topology (lisp+alt), http://tools.ietf.org/html/draft-fuller-lisp-alt-03.txt.
18. L. Gao, On inferring autonomous system relationships in the internet, *IEEE/ACM Trans. Netw.* **9**(6), 733–745, 2001.
19. T.G. Griffin, F.B. Shepherd and G. Wilfong, Policy disputes in path-vector protocols, in *ICNP '99: Proceedings of the Seventh Annual International Conference on Network Protocols*, IEEE Computer Society, Washington, DC, p. 21, 1999.
20. T.G. Griffin, F.B. Shepherd and G. Wilfong, The stable paths problem and interdomain routing, *IEEE/ACM Trans. Netw.* **10**(2), 232–243, 2002.
21. W. Herrin, What does a bgp route cost?, http://bill.herrin.us/network/bgpcost.html, 2008.
22. C. Huitema and S. Weerahandi, Internet measurements: The rising tide and the dns snag, in *Proceedings of the 13th ITC Specialist Seminar on IP Traffic Measurement Modeling and Management*, IPseminar, ITC, Monterrey, CA, 2000.
23. J. Khoury, Discovery in large-scale networks, Ph.D. Thesis, University of New Mexico, 2009 (to appear).
24. J. Khoury and C.T. Abdallah, Identifier-based discovery mechanism design in large-scale networks, in *Future-Net'09: Proceedings of the International Workshop on the Network of the Future, with IEEE ICC'09*, Dresden, Germany, 2009.
25. J. Kleinberg, The convergence of social and technological networks, *Commun. ACM* **51**(11), 66–72, 2008.

26. J. Kleinberg and P. Raghavan, Query incentive networks, in *FOCS '05: Proceedings of the 46th Annual IEEE Symposium on Foundations of Computer Science*, IEEE Computer Society, Washington, DC, pp. 132–141, 2005.
27. T. Koponen, M. Chawla, B.G. Chun, A. Ermolinskiy, K.H. Kim, S. Shenker and I. Stoica, A data-oriented network architecture, in *Proceedings of SIGCOMM'07*, ACM, Kyoto, Japan, 2007.
28. D. Krioukov, k.c. claffy, K. Fall and A. Brady, On compact routing for the internet, *SIGCOMM Comput. Commun. Rev.* **37**(3), 41–52, 2007.
29. H. Levin, M. Schapira and A. Zohar, Interdomain routing and games, in *STOC '08: Proceedings of the 40th Annual ACM Symposium on Theory of Computing*, ACM, New York, NY, pp. 57–66, 2008.
30. C. Li, B. Yu and K. Sycara, An incentive mechanism for message relaying in unstructured peer-to-peer systems, in *AAMAS '07: Proceedings of the 6th International Joint Conference on Autonomous Agents and Multiagent Systems*, ACM, New York, pp. 1–8, 2007.
31. A. Mas-Colell, M.D. Whinston and J.R. Green, *Microeconomic Theory*, Oxford University Press, 1995.
32. D. Massey, L. Wang, B. Zhang and L. Zhang, A scalable routing system design for future internet, in *ACM SIGCOMM Workshop on IPv6 and the Future of the Internet*, ACM Press, New York, 2007.
33. D. Meyer, L. Zhang and K. Fall, Report from the IAB Workshop on Routing and Addressing, Internet RFC 4984, Septenber 2007.
34. R. Moskowitz, P. Nikander and P. Jokela, Host identity protocol architecture, RFC 4423, 2006.
35. N. Nisan and A. Ronen, Algorithmic mechanism design, in *Proceedings of the 31st ACM Symposium on Theory of Computing*, Atlanta, GA, 1999.
36. N. Nisan, T. Roughgarden, E. Tardos and V.V. Vazirani, *Algorithmic Game Theory*, Cambridge University Press, New York, 2007.
37. Papadimitriou, C.: Algorithms, games, and the internet, in *STOC '01: Proceedings of the Thirty-Third Annual ACM Symposium on Theory of Computing*, ACM, New York, pp. 749–753, 2001.
38. Y. Rekhter, T. Li and S. Hares, RFC 4271: A border gateway protocol 4 (bgp-4), 2006.
39. J. Shneidman, D.C. Parkes and M. Seltzer, Overcoming rational manipulation in distributed mechanism implementations, Tech. Rep., Harvard University, 2003.
40. I. Stoica, R. Morris, D. Liben-Nowell, D.R. Karger, M.F. Kaashoek, F. Dabek and H. Balakrishnan, Chord: A scalable peer-to-peer lookup protocol for internet applications, *IEEE/ACM Trans. Netw.* **11**(1), 17–32, 2003.
41. S. Yuen and B. Li, Strategyproof mechanisms towards dynamic topology formation in autonomous networks, *Mob. Netw. Appl.* **10**(6), 961–970, 2005.