



DIGITAL ACCESS TO SCHOLARSHIP AT HARVARD

Enabling Sharing in Auctions for Short-Term Spectrum Licenses

The Harvard community has made this article openly available.
[Please share](#) how this access benefits you. Your story matters.

Citation	Kash, I. A., Rohan Narayana Murty, and David C. Parkes. 2013. "Enabling Sharing in Auctions for Short-Term Spectrum Licenses." In <i>Mechanisms and Games for Dynamic Spectrum Allocation</i> , ed. Tansu Alpcan, Holger Boche, Michael L. Honig, and H. Vincent Poor, 464–494. Cambridge: Cambridge University Press.
Published Version	doi:10.1017/cbo9781139524421.020
Accessed	February 19, 2015 5:17:06 PM EST
Citable Link	http://nrs.harvard.edu/urn-3:HUL.InstRepos:13497110
Terms of Use	This article was downloaded from Harvard University's DASH repository, and is made available under the terms and conditions applicable to Other Posted Material, as set forth at http://nrs.harvard.edu/urn-3:HUL.InstRepos:dash.current.terms-of-use#LAA

(Article begins on next page)

16 Enabling Sharing in Auctions for Short-term Spectrum Licenses

Ian A. Kash, Rohan Murty, and David C. Parkes

Wireless spectrum is a valuable and scarce resource that currently suffers from under-use because of the dominant paradigm of exclusive-use licensing. We propose the SATYA auction (Sanskrit for truth), which allows short-term leases to be auctioned and supports diverse bidder types, including those willing to share access and those who require exclusive-use access. Thus, unlike unlicensed spectrum such as Wi-Fi, which can be shared by any device, and exclusive-use licensed spectrum, where sharing is precluded, SATYA improves efficiency through supporting sharing alongside quality-of-service protections. The auction is designed to be scalable, and also strategyproof, so that simple bidding protocols are optimal. The primary challenge is to handle the externalities created by allocating shared-use alongside exclusive-use bidders. Using realistic Longley-Rice based propagation modeling and data from the FCC's CDBS database, we conduct extensive simulations that demonstrate SATYA's ability to handle heterogeneous bidders involving different transmit powers and spectrum needs.

16.1 Introduction

Currently, spectrum is licensed by governments in units covering large areas at high prices and for long periods of time, which creates a large barrier to entry for new applications. The main alternative, unlicensed bands such as Wi-Fi, has offered tremendous benefit, but is subject to a “tragedy of the commons” where these bands become congested and performance suffers [6].

Many researchers and firms (e.g., [4, 18, 35]) have proposed creating a *secondary market* for dynamic spectrum access to provide a new way to access spectrum. The idea is that primary spectrum owners would be able to sell short-term leases. The same technology could also be used by the government to provide a new approach to the licensing of government owned spectrum, by selling short-term licenses in a primary market. This is made possible by recent advances in building spectrum registries [18, 29].

Rather than just sell exclusive-use access to spectrum, we advocate the adoption of auction technology to allocate spectrum to both exclusive-use and shared-use. An exclusive-use license guarantees a winner no interference but can be inefficient. For example, devices such as wireless microphones are only used occasionally, and other devices can use the same spectrum when the wireless microphone is not in use. This

heterogeneity of devices and demand patterns presents an opportunity for sharing. In addition, many devices are capable of using a *medium access controller* (MAC) to share spectrum when there is contention.

Auctions generate revenue and also enable efficient spectrum use through the dynamic reprovisioning of spectrum. Auctions improve efficiency relative to fixed price schemes by allowing prices to adapt dynamically in response to varying demand. In addition, auctions provide incentives for different users to describe through bids their access requirements, for example specifying exclusive-use or allowing sharing.

We describe the SATYA auction, which is designed to allocate short-term spectrum access across a wide range of scenarios, embracing different access technologies and different types of users, including individuals and service providers. This makes SATYA well suited to handle the mixture of users found today in settings such as Wi-Fi while still providing a better service than unlicensed spectrum. In determining the allocation of spectrum, SATYA considers the effect of interference on the value of the allocation to different bidders. Because of the possibility of sharing spectrum, bidders in SATYA care about how spectrum is allocated to other bidders, along with the dynamic access patterns of those bidders. Most existing auction designs for wireless spectrum either fail to allow sharing, or fail to scale to realistic problem sizes.

In order to make the algorithms for winner determination and determining payments scalable we impose structure on the bidding language with which bidders describe the effect of allocation to others on their value. The language allows bidders to express their value for different allocations, given probabilistic activation patterns, an interference model, and under different requirements expressed by bidders for shared vs exclusive-access spectrum. In determining the the value of an allocation, the auction must determine the fraction of each bidder's demand that is satisfied in expectation, considering sharing and interference patterns. For this purpose, we adopt a model for resolving contention by devices and assume knowledge of which devices will interfere with each other given allocation (based on device location), and represented through a conflict graph.

The SATYA auction is *strategyproof*, which is a property that makes simple bidding protocols optimal for users or the devices representing users. In particular, the utility-maximizing (the utility to a user is modeled as the difference between the user's value and the price) strategy is to bid truthfully, regardless of the bids, and regardless of the kinds of activation patterns and sharing or exclusive-use preferences of other users.

Strategyproofness is a property that is desirable for large-scale, distributed systems involving self-interested parties because it promotes stability—the optimal bid is invariant to changes in bids from other users. In comparison, bidders would need to keep changing their bids in a non-strategyproof auction in order to maximize utility. This continual churn in bids imposes an overhead on system infrastructure as well as participants.

Even without sharing, the problem of finding a value-maximizing (and thus efficient) allocation of spectrum is NP-hard [19]. In obtaining scalability, we adopt a greedy algorithm for determining the winners and the spectrum allocation. A crucial difficulty that arises because of externalities is that a straightforward greedy approach to allocation

fails to be monotonic. What this means is that a user can submit a larger bid but receive less spectrum (in expectation, given the interference, sharing and activation patterns). Monotonicity is sufficient, and essentially necessary, for strategyproof auctions [30]. This leads to a significant design challenge.

In recovering monotonicity, SATYA modifies the greedy algorithm through a novel combination of bucketing bids into value intervals in which they are treated equally (this idea was employed in Ghosh and Mahdian [12]), along with a “computational ironing” procedure that is used to validate the monotonicity of an allocation and perturb the outcome as necessary to ensure monotonicity (this idea was introduced by Parkes and Duong [31]). These techniques prevent cases in which an increase in bid can change the decision of the greedy algorithm to something that looks just as good given decisions made so far, but turns out to be worse for the bidder because of interference with other bids that are subsequently allocated.

In evaluating SATYA, we use real-world data sources to determine participants in the auction, along with the Longley-Rice propagation model [3] and high-resolution terrain information, to generate conflict graphs. We compare the performance of SATYA against other auction algorithms and baselines. Our results show that, when spectrum is scarce, allowing sharing through the SATYA auction increases efficiency by 40% over previous approaches while generating revenue for spectrum owners. The baseline also serves to provide an upper bound on the potential cost of requiring strategyproofness relative to a protocol that is designed to be efficient, but where participants in any case choose to behave in a way that is approximately truthful.

16.1.1 Related Work

Most proposed auction designs for the sale of short-term spectrum licenses preclude sharing amongst auction participants [5, 10, 23, 33, 35, 36]. From amongst these, VERITAS [35] was the first strategyproof design. However, VERITAS does not support sharing. We compare SATYA to VERITAS in the empirical analysis.

Ileri et al. [16, 17] consider models where users have exclusive access but only for short time periods, which effectively permits some amount of sharing. Kasbekar and Sarkar [24] propose a strategyproof auction and allow for sharing amongst winners. But the winner determination algorithm in their proposed auction is not scaleable because bids are explicitly represented on different joint spectrum allocations, which requires considering an exponential (in the number of participants) number of allocations. In contrast, our use of a structured bidding language allows us to achieve good efficiency while considering a polynomial number of allocations. Huang et al. [15] propose an auction design where bidders bid to share a single channel, but their design is not strategyproof and they do not address the issue of how to assign channels when multiple are available.

Gandhi *et al.* [11] propose an auction that allows sharing amongst winners, but differs from our approach in that it does not provide strategyproofness (and thus lacks an equilibrium analysis and is otherwise hard to evaluate), and precludes sharing between users who want exclusive-use when active but are only intermittently active (e.g., wire-

less microphone devices) and other users. Externalities have also been considered in auction theory [21, 22], but without the combinatorial aspect of our allocation problem and the difficulties this implies for achieving strategyproofness. A number of papers have considered externalities in online advertising [8, 12, 13, 25, 32]. However, this work (and similarly that of Krysta *et al.* [26] on the problem of externalities in general combinatorial auctions) is not directly relevant, as the externalities in spectrum auctions have a special structure, of which SATYA takes advantage in order to achieve compact bid representations and scalable winner determination.

16.2 Challenges in Auction Design

In this section we describe in some more detail the challenges that arise when designing a spectrum auction that permits sharing. First, we discuss the general form of an auction and define strategyproofness. Second, we present a central result due to Myerson [30] that provides a general framework for designing strategyproof auctions through the use of a monotone allocation rule. Finally, we introduce *reserve prices*, which are a standard approach to increasing the revenue from an auction.

In the simplest type of auction, a single item is for sale. Each bidder i has private information about his value $V_i > 0$. Let $B_i \geq 0$ denote the bid from bidder i . Each bidder receives an allocation $A_i \in \{0, 1\}$, where $A_i = 1$ if the bidder gets the item and 0 otherwise. Feasibility insists that $\sum_i A_i \leq 1$. Writing $B = (B_1, \dots, B_n)$ for bids from n bidders, then we can write the allocation selected as a function $A(B) = (A_1(B), \dots, A_n(B))$. Finally, each bidder makes some payment $P_i \geq 0$, that depends on the bids, so we write $P_i(B)$.

In a standard model, a bidder's utility, which captures his preference for the outcome of an auction, is

$$U_i(B) = V_i A_i(B) - P_i(B), \quad (16.1)$$

and represents the true value for the allocation minus the payment. There are many ways such an auction can be run.

One approach, known as a *first-price auction*, is that each bidder names a price and the bidder who bids the most wins the item and pays their bid with $P_i(B) = B_i$ for the winner. With perfect knowledge, a bidder should bid slightly more than the highest bid of other bidders (to a maximum of V_i), in order to pay as little as possible. Thus bidders try to anticipate how much others will bid, and bid accordingly. This gives a first-price auction high strategic complexity.

Another approach, due to Vickrey [34], is a *second-price auction*, where each bidder names a price and the bidder who bids the most wins the item. However, instead of paying the bid price, the payment is equal to the bid of the second highest bidder. In such an auction, a bidder has a simple strategy that is (weakly) optimal no matter what bid true value $B_i = V_i$. The Vickrey auction is strategyproof.

DEFINITION 16.1 An auction defined with allocation rule A and payment rule P is

strategyproof if

$$U_i(V_i, B_{-i}) \geq U_i(B'_i, B_{-i}) \tag{16.2}$$

for all bid profiles B , all agents i , and all alternate bids B'_i , where $B_{-i} = (B_1, \dots, B_{i-1}, B_{i+1}, \dots, B_n)$ is the bid profile without agent i .

As explained in the introduction, a strategyproof auction is desirable because of the effect it has in simplifying bidding strategies and because of the overhead it removes from the infrastructure by precluding the need for bids to be continually updated as bids from others change.

But how to design such an auction in our setting? One thing to recognize is that the allocation will be much more complicated: analogous to an item is a channel \times location (where the location depends on the location of the bidder's device and the channel is a range of frequencies) In addition to there being multiple items to allocate, there will be interference such that the value of an item depends on the other bidders allocated similar items. In particular, bidders that are geographically close to each other and are allocated the same channel will interfere with each other.

Part of the challenge is to describe a concise language to represent a bidder's value for different possible allocations. Another part of the challenge is to ensure that the allocation can be computed in polynomial time. The NP-hardness of the winner determination problem precludes a general auction design due to Vickrey, Clarke, and Groves [7, 14, 34], that would be strategyproof and efficient in our domain.

In achieving strategyproofness, an important property is that an allocation algorithm be *monotone*, which requires that $A_i(B_i, B_{-i})$, where $B_{-i} = (B_1, \dots, B_{i-1}, B_{i+1}, \dots, B_n)$, is weakly increasing in the bid of bidder i , fixing the bids of others, so that $A_i(B_i, B_{-i}) \geq A_i(B'_i, B_{-i})$ for $B_i \geq B'_i$.

Theorem 16.2.1 (Myerson [30]) An auction is strategyproof if and only if for all bidders i , and fixed bids of other bidders B_{-i} ,

- 1 $A_i(B)$ is a **monotone** function of B_i (increasing B_i does not decrease $A_i(B)$), and
- 2 $P_i(B) = B_i A_i(B) - \int_{z=0}^{B_i} A_i(z, B_{-i}) dz$.

Even beyond strategyproofness, monotonicity is still a worthwhile goal because it guarantees that participants attempting to optimize their bid will only increase the amount they receive and the amount they pay when they increase their bid.

In the case of an auction for a single good, the nature of monotonicity is simple: a bidder must continue to win the good when bidding a higher price. However, in our setting, winning a channel alone is not sufficient to make a bidder happy. In particular, if the channel is heavily used by others in a bidder's neighborhood it may have little value. Thus, a bidder cares not only about whether or not he is allocated a channel, but also who else is allocated the same channel. The effects of the allocations of other bidders on the value of winning a good are known as *externalities*. This complicates the auction design because the allocation rule must be monotone not only in whether a bidder gets a channel, but also the amount of sharing that occurs on that channel. But

once an allocation rule has been developed that is monotone in this sense, the auction can be made strategyproof through standard methods.

While determining the prices bidders pay requires computing an integral, in many cases this integral has a simple form. For example, in the (deterministic) single good case the allocation to a bidder, A_i , only takes on two values: 0 when the bidder does not get the good and 1 when he does. Since the allocation must be monotone, it is entirely determined by the *critical value* where it changes from 0 to 1. Thus, computing the integral reduces to the problem of determining the minimum bid that the bidder could have made and still been allocated.

In addition to strategyproofness, the proposed auction designs for the allocation of short-term licenses and spectrum sharing can be evaluated in terms of the twin goals of:

- *Allocative efficiency*: rather than maximize throughput or spectral efficiency, allocate resources to maximize the total utility from the allocation. Thus, in addition to traditional metrics we also report the total value from the allocations determined at the outcome of SATYA.
- *Revenue*: good revenue properties are important in order to provide an incentive for spectrum owners to participate in the market.

Efficiency is often held to be of primary importance when designing a marketplace because it provides a competitive advantage over other markets, and encourages participation by buyers. Maximizing revenue can be at odds with efficiency because it can be useful to create scarcity in order to boost revenue. One way to do this is to adopt a *reserve price*. We will examine the tradeoff between efficiency and revenue that can be achieved by adjusting the reserve price in SATYA.

16.3 The Model of Shared Spectrum and Externalities

16.3.1 User Model

In order to find opportunities to share among heterogeneous users (e.g., a user with a wireless device, or a TV station), we need a language to describe the requirements of each possible *type* of user.

Our model uses discrete intervals of time (called *epochs*), with auctions clearing periodically and granting the right to users to contend for access to particular channels over multiple epochs. Thus our approach models participants who regularly want spectrum in a particular location over a period of time. Participants who wish to enter or leave need to wait until the next time the auction is run. The ultimate allocation of spectrum arises through random activation patterns of users and interference effects, and depends on specifics of the medium-access control (MAC) contention protocol. The effect of this MAC protocol is modeled within SATYA in determining the allocation.

The interference between users and their associated devices is modeled through a *conflict graph*, $G = (V, E)$, such that each user i is associated with a vertex ($i \in V$) and an edge, $e = (i, j) \in E$ exists whenever users i and j would interfere with each other if

they are both active in the same epoch and on the same channel. Note that, for service providers such as TV stations, defining the conflict graph may be complex as it requires making decisions about the acceptability of interference over some portion of the served area.

We allow for both exclusive-use and “willing to share” users, where the former must receive access to a channel without contention from interfering devices whenever they are active, while the latter can still obtain value through contending for a fraction of the channel with other interfering devices.

We say that a channel is *free*, from the perspective of user i in a particular epoch, if no exclusive-use user j , who interferes with i and is assigned the right to the same channel as i , is active in the epoch.

Formally, we denote the set of user types T . Each type $t_i \in T$ is a tuple $t_i = (x_i, a_i, d_i, p_i, C_i, v_i)$, where:

- $x_i \in \{0, 1\}$ denotes whether the user requires *exclusive-use* of a channel in order to make use of it ($x_i = 1$) or willing to share with another user while both are active on the channel ($x_i = 0$).
- $a_i \in (0, 1]$ denotes the *activation probability* of the user: the probability that the user will want to use the channel, and be active, in an epoch.
- $d_i \in (0, 1]$ is the *fractional demand* of the channel that a user who is willing to share access requires in order to achieve full value when active.
- $p_i \geq 0$ denotes the per-epoch penalty incurred by the user when active and the assigned channel is not free. Both exclusive-use and non exclusive-use users can have a penalty.
- $C_i \subseteq C = \{1, 2, \dots\}$, where C is the set of channels to allocate, each corresponding to a particular spectrum frequency, denotes the channels that user i is able to use (the user is indifferent across any such channel.)
- $v_i \geq 0$ denotes the per-epoch value received by the user in an epoch in which it is active, the channel is free, and in the case of non exclusive-use types, the user receives at least a share d_i of the available spectrum.

In this model, each user demands a single channel. We discuss an extension to multiple channels in Section 16.4.6.

Some of the parameters that describe a user’s type are a direct implication of the user’s technology and application domain. For example, whether or not a user requires exclusive-use when active and is unwilling or unable to share falls into this category. Users that can use a MAC will tend to be able to share, other users will tend not to be able to. As we explain below, users operating low-power TV stations or with wireless microphone devices would likely be in this category. The set of channels C_i on which a user’s device can legally broadcast will tend to be easy to define.

For parameters such as the activation probability (how often the user makes use of the channel), and the fractional demand (how much of the channel is used when active), we assume that these can be estimated by the device, and then monitored by the network environment upon the outcome of an auction with the user punished if this information

is mischaracterized. For example, a user could be banned from participating in future auctions. But certainly, the fractional demand d_i and activation probability a_i may be difficult to estimate in some cases, and especially when first bidding, due to uncertainty arising from the effects of interference, anticipated traffic, and propagation.

Examples

- A user who wishes to run a low-power (local) TV station on a channel would be unable to share it with others when active ($x_i = 1$), would be constantly broadcasting ($a_i = 1$), and would have a very large penalty p_i since it is unacceptable for the broadcast to be interrupted by someone turning on another (exclusive-use) device.
- A user with a wireless microphone cannot share a channel when active ($x_i = 1$), but is used only occasionally ($a_i = 0.05$) and has a smaller value of p_i since it may be acceptable if the user is occasionally unable to be used when there is another exclusive user also trying to use the channel.¹
- A bidder may want to run a wireless network. Such a user would have constant traffic ($a_i = 1$), consume a large portion of the channel ($d_i = 0.9$), and might have a large penalty similar to a TV station for being completely disconnected. However, such a user is willing to share the channel with other non-exclusive types ($x_i = 0$), and will pay proportionately less for a smaller fraction of the bandwidth.
- A bidder representing a delay tolerant network [20], who occasionally ($a_i = 0.2$) would like to send a small amount of information ($d_i = 0.4$) if the channel is available. Such bidders might have a low or even no penalty as their use is opportunistic.

The per-epoch penalty is the cost to a user that is incurred in an epoch when the user is active (wants to use the network) but the channel is encumbered by an exclusive-use device. This can represent the known cost of using an alternate network or a contractual rebate.

The per-epoch value of a user v_i represents the dollar value that a user assigns to being able to access the channel when active, that is, in an epoch when the user wants to use the network. For an exclusive-use user, it is the per-epoch value for gaining exclusive access during that epoch. For a user willing to share, it is the per-epoch value for gaining a fraction d_i of the channel (as long as the channel is unencumbered by an exclusive-use device), and the assumption is that the value falls off linearly for a share below d_i . We will design a strategy-proof auction in which it is optimal for users to report their true value of v_i when bidding in the auction.

16.3.2 Allocation Model

Let $A_i \in C_i \cup \{\perp\}$ denote the channel allocated to each user i , where \perp indicates the user has not been assigned a channel. Let $A = (A_1, \dots, A_n)$ denote the joint allocation to n users. To allocate a channel means that the user has the right to contend for the channel

¹ Indeed, it might make sense from an efficiency perspective to have several such devices share a channel if they interfere with each other sufficiently rarely.

when active, along with other users that interfere with the user and are allocated the same channel.

Exclusive-use users take priority over non exclusive-use users, and only experience interference when multiple exclusive-use users are simultaneously active. Non exclusive-use users share the channel when active simultaneously, and when the channel is free of exclusive-use users.

Let $V_i(A, t)$ denote the *expected value* to user i for allocation A given type profile $t = (t_1, \dots, t_n)$. The value also depends on the conflict graph G , since this affects the interference between users. But we omit this term for notational simplicity.

An efficient allocation of spectrum maximizes the expected total value across the user population, that is

$$A^* \in \arg \max_A \sum_i V_i(A, t) \quad (16.3)$$

All allocations are feasible in our setting, since the expected value captures the negative externality due to interference. For this, we define the expected value $V_i(A, t)$ as,

$$= \begin{cases} 0 & \text{if } A_i = \perp, \text{ otherwise} \\ v_i \cdot a_i \Pr_i(F|A, t) \frac{E_A[S_i|F, t]}{d_i} - p_i \cdot a_i (1 - \Pr_i(F|A, t)). \end{cases} \quad (16.4)$$

A user's value depends first on the expected fraction of the user's request that can be satisfied. The user can only use the channel when it is not in use by another exclusive-use user, so we let $\Pr_i(F|A, t) \in [0, 1]$ denote the probability that the channel is *free* (F), with no exclusive-use user interfering with the allocated channel. Given that the channel is free, the user may still have to share with other users. For this, $E_A[S_i|F, t] \in [0, 1]$ denotes the maximum of the expected fraction of a channel that is available to user i given an epoch in which the channel is unobstructed by an exclusive-use user, the user is active, and given user i 's demand. For an exclusive-use user, this amount is always $E_A[S_i|F, t] = 1$, because such a user receives complete access to the channel when active and the channel is otherwise free.

Thus, the first term in (16.4) takes the expected fraction of channel capacity (necessarily less than d_i) supplied in an epoch in which the user is active, and in which the channel is free from exclusive-use users, and multiplies this by the probability the channel is free and the user is active $a_i \Pr_i(F|A, t)$, and the user's value for receiving d_i fraction of the channel in an epoch. This assumes that a user's value is linear in the available bandwidth (up to max-demand d_i .) The second term in (16.4) calculates the expected per-epoch *penalty* due to the channel not being free when a user is active (the probability of which is $a_i \cdot (1 - \Pr_i(F|A, t))$).

To complete this, we need to also define the probability that the channel allocated to user i is free, given allocation A and type profile t . This is given by the expression,

$$\Pr_i(F|A, t) = \prod_{j \in N_i \text{ s.t. } A_j = A_j \wedge x_j = 1} (1 - a_j), \quad (16.5)$$

where N_i is the set of neighbors of i in G . This is the joint probability that no exclusive-use neighbor in the conflict graph, allocated the same channel as i , is active in an epoch.

Finally, we require an expression for $E_A[S_i|F, t] \leq d_i$, the expected fraction of a channel available to a user in an epoch when it is active and the channel is free. For this, we first consider the effect of a fixed number of active (non exclusive-use) neighbors in such an epoch.

For this, we assume a *Carrier Sense Multiple Access* (CSMA) style MAC, in which bandwidth is shared as equally as possible among active (non-exclusive-use) users, subject to the constraint that no user i receives more than its demand d_i . Formally, if N_a is a set containing i and the active neighbors of i with whom i shares a channel in the allocation, and $N_f = \{j \in N_a \mid d_j < f\}$, then user i receives a share of the available bandwidth on the channel equal to,

$$share_i(N_a, t) = \min \left(d_i, \max_{f \in [0,1]} \frac{1 - \sum_{j \in N_f} d_j}{|N_a - N_f|} \right) \quad (16.6)$$

The user either gets the full demand d_i or, failing that, the fair share (which the max in the equation determines). If all users have the same demand d_i , this reduces to each either the full demand being satisfied if $d_i \leq 1/|N_a|$ or receiving a $1/|N_a|$ share of the channel capacity otherwise. If some users demand less than their fair share, the remainder is split evenly among the others.

In completing an expression for $E_A[S_i|F, t]$, we adopt $v_i(A, c)$ to denote the set of neighbors of i on conflict graph G that, in allocation A , are allocated channel c . In particular, $v_i(A)$ denotes the set of neighbors allocated the same channel as i . The probability that a particular set, $N' \subseteq v_i(A)$ is active in any epoch is,

$$active_i(N', t) = \left(\prod_{j \in N'} a_j \right) \left(\prod_{\ell \in v_i(A) - N'} (1 - a_\ell) \right) \quad (16.7)$$

From this, a user's expected share of the channel, given that the user is active and the channel is free (where the expectation is computed with respect to random activation patterns of interfering neighbors) is given by,

$$E_A[S_i|F, t] = \begin{cases} 0 & \text{if } \Pr_t(F|A, t) = 0 \\ 1 & \text{if } x_i = 1 \\ \sum_{N' \subseteq v_i(A)} active_i(N', t) share_i(N', t), & \text{o.w.} \end{cases} \quad (16.8)$$

The two special cases cover exclusive-use users (who always receive their full demand when active, conditioned on the channel being otherwise free), and users for whom the channel is never free (for whom we arbitrarily define it to be 0, because the value in this case turns out to be irrelevant).

In general, computing $E_A[S_i|F, t]$ requires time exponential in the number of neighbors $v_i(A)$ with which i shares a channel. In making this practical, sharing can be limited to $d_{\max} \ll n$ neighbors, and the calculation can be completed in time that scales as $O(2^{d_{\max}})$. Alternatively, it may turn out that d_{\max} is already small due to the nature of the conflict graph. Indeed, in our experiments for practical models of signal propagation, and even with hundreds of users participating in the auction, we did not need to impose such a limitation.

16.4 Auction Algorithm

Turning to the design of SATYA, we assume that the only component of a user's type that can be misreported is v_i , which represents the per-epoch value of the user when active. Designing an auction that is strategy-proof in regard to per-epoch value v_i is the focus of this section.

As explained in Section 16.3.1, it seems reasonable to assume that many of the parameters of a user's type can be checked by the network at the outcome of an auction and enforced through punishment (e.g., kicking out of the auction environment in the future), or are fundamental to the operation of a user's device (e.g., whether or not it can share the channel or needs exclusive-use) and thus not useful to manipulate. Misreporting them (for example by remaining active and sending junk data to increase a_i), implementing such manipulations would be costly to users and in many cases would simply result in higher payments. Thus, for simplicity we ignore this possibility.

There are a number of interpretations of the penalty incurred by a user when a channel is encumbered by an exclusive-use device under which it seems reasonable to treat it as known. For example, the penalty could represent the cost to gain access to a fallback network when this network is unavailable or a contractual payment the spectrum owner is obligated to make. The exact value may also not be important, as long as it is large. In practical terms, the value of the penalty controls the level of sharing that is permissible with exclusive users who are occasionally active, and getting this to the correct order of magnitude may be sufficient. In the extreme, SATYA works fine taking $p_i = 1$ for all i , which forbids such sharing.

Even if no users are permitted to share channels, finding an optimal allocation is NP-Hard [19]. Assigning bidders to channels such that no two neighbors have the same channel is a *graph coloring problem*. Therefore we adopt a greedy algorithm for allocation, modified to achieve monotonicity.

16.4.1 Externalities and Monotonicity

Let us first define *monotonicity* in our setting. For this, it is convenient to drop the complete type profile t from notation and write $\Pr_i(F|A)$ and $E_A[S_i|F]$ in place of $\Pr_i(F|A, t)$ and $E_A[S_i|F, t]$ respectively. In addition, let $b = (b_1, \dots, b_n)$ denote the joint bid vector received from users, with $b_j \geq 0$ for all j .

Fixing the bids $b_{-i} = (b_1, \dots, b_{i-1}, b_{i+1}, \dots, b_n)$ of other users, an allocation algorithm $\mathcal{A}(b)$ (defining an allocation for every bid profile) is *monotone* if,

$$\begin{aligned} \Pr_i(F|\mathcal{A}(b'_i, b_{-i}))E_{\mathcal{A}(b'_i, b_{-i})}[S_i|F] \\ \geq \Pr_i(F|\mathcal{A}(b))E_{\mathcal{A}(b)}[S_i|F], \end{aligned} \quad (16.9)$$

for all bids $b'_i \geq b_i$. This insists that the *expected share* of a channel available to a user, conditioned on being active, weakly increases as the user's bid increases.

Figure 16.1 shows how monotonicity can fail for simple greedy algorithms. The greedy algorithm considers each user in (decreasing) order of bids and allocates the

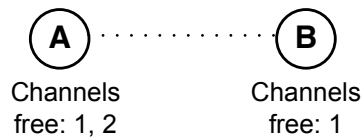


Figure 16.1 A potential violation of monotonicity. Nodes A and B are in contention range. At node A's location channels 1 and 2 are free; at B only channel 1 is free.

user to the best available channel in terms of maximizing value (or no channel if that is better). If there is a tie, the algorithm uses some tie-breaking rule, such as the lowest channel number. If user A has a lower bid than user B, the algorithm assigns user B to channel 1, then user A to channel 2, and both are fully satisfied. If user A raises its bid above that of user B, user A will be assigned to channel 1. Then, assuming sharing is better than leaving B unassigned, the algorithm assigns user B to channel 1, and user A receives less value due to interference.

16.4.2 High level approach

The monotonicity violation from Figure 16.1 would be prevented if the algorithm was not allowed to assign user B to channel 1 in the second case. We do this for many cases by assigning each user to a “bucket” based on his bid, such that the more a user bids the higher the bucket to which he is assigned. Users are not allowed to share with a user from a higher bucket. Thus, if user B is in a lower bucket than user A, user B will simply not be assigned a channel. If both users are in the same bucket, we will consider them in some order independent of their actual bids, and adopt in place of their bid value the minimal possible value associated with the bucket. The effect is that the allocation decision is invariant to a user's bid while the bid is in the same bucket. Since users are only allowed to share with other users within their buckets, the way buckets are chosen is an important parameter of our algorithm. Larger buckets create more possibilities for sharing. However, they also mean that the algorithm pays less attention to user's bids, so they may decrease the social welfare (the total value of the allocation) and revenue.

Bucketing prevents many violations of monotonicity, but it is not sufficient to prevent all of them. In particular, the example from Figure 16.1 can still occur if user A is in a lower bucket than user B and then raises his bid so they are in the same bucket (if he raises it to be in a *higher* bucket there is no problem). To deal with this case we adapt a technique known as “ironing” [31] to this domain. This is a post-processing step in which allocations that might violate monotonicity are undone. Given an input (a set of bids) to an allocation algorithm, the basic idea is to check the sensitivity of the allocation with respect to unilateral changes in the bid value by each bidder. In Parkes and Duong [31] this is applied to a problem of stochastic optimization, and a failure in regard to checking higher bids is addressed by unallocating the bidder at the current input. In the context of the SATYA auction, sensitivity is checked in regard to lower bids, and failure is addressed by unallocating other bidders that are sharing a channel with the user (improving the allocation for the bidder at the current input).

For each user allocated in the current bucket, we ask the counterfactual question “If this user were instead in the next lower bucket, is it possible he would be allocated?” If so, we guarantee that the user is satisfied in the current bucket by canceling (or “ironing”) the allocations of other users with whom he shares. In Figure 16.1, if user A were in a lower bucket he would be allocated a channel. Therefore, in the ironing step, the algorithm would change user B’s allocation and not allocate a channel in the current bucket. It will be important, though, that a channel allocation that is canceled in this way will be considered unavailable for future allocation. This prevents the need for nested arguments involving the effect of ironing on future allocations, future ironing of future allocations, and so on.

In this high level description, we have assumed that any two users who interfere with each other cannot share a channel without harming each other. In reality, this is not the case; users capable of using a MAC and sending at sufficiently low rates will have a negligible effect on each other. Many of the more intricate details of our algorithm come from adapting the general approach to take advantage of this fact and allow more efficient use of wireless spectrum.

16.4.3 The SATYA Algorithm

SATYA begins by assigning each user i to a bucket based on the user’s bid value b_i . There are many ways this can be done as long as it is monotone in the user’s bid. For example, user i with an activity-normalized bid $a_i b_i$ could be assigned to value bucket with bounds $[2^\ell, 2^{\ell+1})$. To be general, we assume that bucketing of values is done according to some function $\beta(k)$, such that bucket k contains all users with (normalized) bids $a_i b_i$ in the range $[\beta(k), \beta(k+1))$.

Once users are assigned to buckets they are assigned channels greedily, in descending order of buckets. The order of assignment across users within the same bucket is determined randomly. Let K_i denote the bucket associated with user i . A channel c is considered to be available to allocate user i at some step in the algorithm, and given the intermediate allocation A , if,

- the channel c is in C_i ;
- assigning i would not cause an externality to a neighbor from a higher bucket: for all $j \in N_i$, with $K_j < K_i$,

$$\sum_{\ell \in \{v_j(A,c) \cup \{i\}\}} d_\ell \leq 1 \quad (16.10)$$

- and, the combined demands of i and the neighbors of i from higher buckets assigned to c are less than 1:

$$d_i + \sum_{j \in v_i(A,c), K_j > K_i} d_j \leq 1 \quad (16.11)$$

We refer to the second condition as requiring that the demands of each neighbor of user i from a higher bucket be satisfied. The third condition requires that the demand of

user i is satisfied. This does not preclude allocations where some user has $E[S_i|F, t] < d_i$. It simply requires that, in such cases, the user is sharing with others in the user's own bucket.

Suppose i is the next user to be considered for allocation. SATYA will identify the channel for which assigning i to the channel has the maximum marginal effect on the total value of all currently allocated users along with user i itself. To do so, for every channel c that is available to the user, and including \perp (and thus not allocating any spectrum to the user), SATYA estimates the expected value to some user j after assigning i to c as

$$e_j(A, b) = \beta(K_j) \Pr_j(F|A, b) \frac{E_A[S_j|F, b]}{d_j} - a_j \cdot p_j(1 - \Pr_j(F|A, b)) \quad (16.12)$$

This estimate differs from the user's actual bid by assuming that each user in a given bucket shares the same value. This is important for achieving monotonicity, because we need to ensure the decision for a user depends on the bucket associated with a user's bid value and not in more detail on a user's value.

Given this, user i is assigned to the channel that maximizes the sum of the expected bid values of each user already allocated and including its own value, and without leaving any user with a negative expected value. The optimal greedy decision might allocate \perp to user i , and thus no spectrum. In the event of a tie, the user is assigned to the lowest numbered among the tied channels (including preferring \perp , all else equal).

After all users in a bucket are assigned channels, there is an *ironing* step in which monotonicity of the allocation is verified, and the allocation perturbed if this fails. Recall that monotonicity violations occur when the greedy allocation makes a "bad" decision for the user and would make a better one had the user been considered later. Bucketing prevents users from being able to move themselves later while staying in the same bucket, but they could still lower their bid enough to drop into the next bucket. To rule out this possibility, the ironing procedure re-runs the allocation procedure for each user with the user placed instead in the next lower bucket. If this counterfactual shows that the final allocation would be *better* for the user, then there is a potential monotonicity violation, and the provisional allocation is modified by changing the assignments of the neighbors with whom the user shared a channel to \perp . Checking only the next bucket is sufficient because if the user can be assigned in any lower bucket he can be assigned in the next bucket.

The complete algorithm is specified in pseudocode as Algorithm 5. In the specification, we use distinct names to be able to refer to allocations created along the way. The variable $A(k, i, j)$ denotes the state of the allocation in bucket k after considering the j th user in the order given by random permutation π on users. Some of these allocations will be used for the counterfactual questions asked by ironing, so i is the user currently being omitted ($i = 0$ if there is no such user).

Lemma 16.4.1 Algorithm 5 is monotone.

The proof of Lemma 16.4.1 is presented in the appendix.