Economic Mechanisms
For Efficient Wireless Coexistence

Technical Report #MIT- LCS-TR-876
August, 2002

Omar Aftab

Abstract

Unchecked greed can adversely impact the efficiency of wireless systems. Individual devices have no motivation to conserve shared resources, and this inevitably leads to poor utilization of spectrum. Indeed, it has been oft demonstrated that there is a real risk of spiraling performance losses, culminating in a classic Tragedy of the Commons. Recognizing that efficient coexistence requires allocating spectrum resources optimally – a classic economic problem – we explore *economic* mechanisms for wireless coexistence. In doing so, we develop a novel approach that allows for more efficient coexistence in wireless systems. Each node is assigned an artificial budget, and nodes intelligently use this wealth to dynamically *bid* for the right to transmit. We explore the workings of such an artificial economy, and end with an overview of decentralized implementations for our system.

Table Of Contents

I Introduction

> "Ruin is the destination toward which all men rush, each pursuing his *own* best interest… Freedom in a commons brings ruin to all"
> [Hardin, G. *The Tragedy of the Commons*, 1968]

Wireless spectrum is a shared resource, and like all shared resources suffers from the age-old problem of greed. Since transmissions on the same frequency interfere with each other, only one node can access the airwaves at a given time. A greedy device could continuously transmit, rendering the system unusable for all others. Even in systems that support simultaneous transmissions, such as CDMA, devices can still exhibit greed by hoarding bandwidth and network resources.

Indeed, mutual interference became such a severe problem that early in the twentieth century the government was forced to step in with a simple, if draconian solution: no one could transmit on a given wireless frequency without a license. The FCC, accordingly, divvied up wireless spectrum into non-overlapping bands and doled them out to users.

> "Before 1927 the allocation of frequencies was left entirely to the private sector and the result was chaos. It quickly became apparent that broadcast frequencies constituted a scarce resource whose use could be regulated and rationalized only by the Government…"
> [The Supreme Court, *Red Lion Broadcasting Co. v. FCC*, 1969. Quoted in Hazlett, 2001]

This continues to be the state of affairs today. Each technology has a license, for exclusive access to its own little wireless band: AM radio operates between 535KHz to 1.7Mhz, and FM ranges from 88 to 108MHz. Garage door openers operate around 40Mhz, cell phones between 824-849Mhz. Even radio-controlled cars have a little niche at 75Mhz, while radio-controlled planes are relegated to 72Mhz.

The exclusive access[1] offered by such centralized licensing schemes clearly solves the problem of mutual interference, but it does so at a *price*. Granting exclusive access to a frequency is perfectly fine for certain applications, such as television, but is quite inefficient for others. In particular, reserving frequencies for systems that under-utilize them is wasteful – spectrum sits idle when they are not transmitting. The concept is analogous to the waste of bandwidth in circuit-switched vs. packet-switched networks[2].

Consider, for example, applications where the demand for bandwidth is bursty: wireless networks where the primary load is browsing, or sending e-mail. Since the load is sporadic, a licensed frequency channel devoted to such a network will be unused most of the time. It would be far better to allow several networks to share the spectrum so that when one is not utilizing it, others can. In addition to the possibility of such inefficient utilization, the centralized licensing scheme involves time-consuming bureaucratic procedures, complicated auction arrangements – and the cost of spectrum is often prohibitive.

Exclusive access thus, clearly has its limitations. Fortunately, there is an alternative: bands of 'unlicensed' wireless spectrum, in which any device is allowed to transmit. Access protocols and sharing schemes allow multiple wireless devices to co-exist within such bands, which include[3] the *Industry, Science, and Medicine band,* the *Millimeter Wave band*, the *National Information Infrastructure band*, and the *Personal Communications Services band*.

Unlicensed spectrum has a number of advantages, particularly for applications needing real-time access to the airwaves. Since there are no licenses – and hence, no licensing costs – wireless systems operating in the band are inexpensive, and can be mass marketed. In addition, the lack of lengthy and complicated licensing procedures means that the technology can be readily adopted, and quickly deployed. It is no surprise then that wireless technologies that utilize unlicensed spectrum, such as 802.11, are rapidly proliferating. Indeed, by the end of 2001 there were more than 11 million 802.11b

[1] Of course, having a single license-holder does not necessarily imply that there will be a single *device*. However, even in cases where the owner deploys multiple devices to share his band, he has a strong economic motivation to ensure that these devices coexist without interference – using the spectrum as efficiently as possible.

[2] The background in this chapter draws repeatedly and heavily from the sundry works by Satapathy and Peha listed in the references.

[3] Peha, J.M, 1998

devices and more than 4,000 public wireless access points in operation. Unlicensed spectrum is particularly suited for ubiquitous and mobile wireless applications, a fact that has greatly contributed to this growth. Users of portable devices no longer need to obtain licenses for every location where they use them, a tremendous advantage.

Further, since multiple devices coexist in a band, and any of these can transmit while the others are inactive, unlicensed spectrum promotes efficient usage of bandwidth: spectrum does not sit idle. In fact, a number of studies[4] have shown that sharing spectrum enhances efficiency – cellular networks, for example, could carry substantially more traffic if they *dynamically* shared spectrum[5].

This efficiency gain is particularly large for applications that transmit at varying rates, generating bursty[6] traffic, such as wireless LANs or PBXs. Wireless e-mail, for example, can accept arbitrary delays and only requires sporadic access to the airwaves[7]. Granting an exclusive band to such applications would be wasteful: they can share spectrum with minimal difficulty.

Challenges For Coexistence

As we have discussed, unlicensed spectrum has a number of advantages over licensed spectrum – the rapid growth of mobile, wireless applications is a testament to that fact. At the same time, however, the technology faces a number of significant challenges.

Paradoxically, these stem from the very facts that make unlicensed spectrum so appealing: there is no exclusive access and devices share frequencies. Since each device can transmit at will, *coexistence schemes* are essential to prevent catastrophic interference. Since each device differs in the rate and duration of transmissions it wishes to send, and this information is *local* to each device, it is challenging to devise a single, uniform coexistence scheme whilst ensuring efficient utilization of spectrum

In licensed bands, the owner has a strong incentive to use his hard-won spectrum efficiently: he wants all his devices to function optimally. Conserving spectrum is as important in unlicensed spectrum – but since there is no distinct owner, there is very little incentive for an *individual* device to conserve shared spectrum[8]. Indeed, it is in the self-interest of each to do the very opposite: it can waste spectrum in order to improve its own performance. Of course, doing so degrades the performance of others and – as more and more devices greedily begin to follow suit – the spectrum eventually becomes clogged and unusable.

The Tragedy of the Commons

The scenario sketched above is a case of the Tragedy of the Commons[9], a famous economic result that describes how unpunished greed can lead to a shared resource becoming unusable. As we discussed above, unlicensed wireless systems are prone to tragedy – consider the following scenario:

Each node in the system considers the question of whether or not to use up an additional unit of spectrum. In doing so, the node rationally seeks to maximize its individual utility, which consists of two components – a gain and a loss.

1. The gain is the performance improvement from having an additional unit of spectrum, +1.
2. The loss is the performance loss caused by additional congestion in the system, but this is a loss *shared by all nodes in the system*. Thus, the loss for the node is a mere *fraction* of –1.

Adding these together, the node decides that using an additional unit of spectrum increases its utility. Since each node in the system reaches this same

[4] Salagado-Galicia, 1995 & 1997; Peha, 1997 etc.
[5] Salgado, Sirbu & Peha. "*A Narrow Band Approach to Efficient PCS Spectrum Sharing",* Feb 1997.
[6] Consider the following example, from Peha, 1997: 8 wireless Personal Branch Exchanges have enough spectrum to support 32 simultaneous calls. The calls are exponentially distributed, and arrive according to a Poisson process. We wish to keep the percentage of calls that can not be handled – and hence, must be blocked – low, say at 1%. If the PBXs share spectrum they can sustain a total load of 68.9%, while if the spectrum is divvied up between the PBXs – each getting four channels – the total load that can be handled is only 21.7%. Sharing, thus, makes far more efficient use of available spectrum.
[7] Satapathy, D.P. and Peha, J.M, 1996
[8] Satapathy, D.P. and Peha, J.M, 1998
[9] Hardin, G. "The Tragedy of the Commons," 1968.

conclusion, the system rapidly becomes hopelessly overloaded: the sum of individual optimums fails to create a global optimum.

> Therein is the tragedy. Each is locked into a system that compels him to increase (usage)… without limit - in a world that is limited. Ruin is the destination toward which all men rush, each pursuing his own best interest in a society that believes in the freedom of the commons. Freedom in a commons brings ruin to all.
> [Hardin, G. *The Tragedy of the Commons*, 1968]

The two essential ingredients for the tragedy are *greed*, and the *freedom* to satisfy that greed without penalty. Consider the classic case of Citizen's Band radios, which operate in an unlicensed band to facilitate mobility. When band usage became high and there was a lot of interference, frustrated users simply purchased more powerful radios. These used up even more spectrum, causing more interference and encouraging even more users to boost their transmission power – eventually clogging up the system. America On-line provides another example: when it offered a flat monthly rate for unlimited Internet access, customers immediately increased usage. Many would remain logged on continuously to avoid having to reconnect, which caused the entire pool of AOL customers to experience performance losses[10].

These examples have their parallels in the wireless domain. A device can exhibit greed by using more bandwidth than it needs, or by simply holding on to a channel longer than necessary to save time re-accessing it later. As more and more devices begin to do so, performance rapidly plummets – particularly where spectrum utilization is high.

Properties of an Ideal Solution

As we have seen, unlicensed spectrum requires a sharing, or access control scheme to decide which device gets access to the spectrum. Ideally, such a scheme should:

1. **Prevent interference:** First and foremost, the scheme must ensure that devices can coexist and transmit without interfering with each other. The traditional wireless CSMA/CA scheme, for example, has a "Listen Before Transmit" policy in which each node tests the airwaves to see if they are in use before initiating transmission.

2. **Curb Greed:** In order to avoid a Tragedy of the Commons, the scheme must provide a penalty for greed: nodes must be given a strong disincentive to squander bandwidth.

3. **Promote Economic Efficiency:** Unlicensed spectrum is a shared resource, one that should be shared in an economically efficient, Pareto-optimal manner. Ideally, the device that most *needs* access to the airwaves should be the one to get it.

In addition, the scheme should be feasible: the technology required should be economical, easily adopted and easily adapted to new challenges and applications.

Existing Schemes

There are a number of ways for avoiding interference with unlicensed devices. One obvious method is simply to keep usage low – either by limiting demand for spectrum by imposing high fees on customers, or by allocating large amounts of excess spectrum.

Of course, this is highly inefficient, and with the rapid increase in unlicensed wireless devices – particularly in the 2.4GHz ISM band – things are only going to get worse. The expense and difficulty of getting licenses may also compel companies to use the unlicensed bands for applications better served by licensed spectrum, increasing utilization and exacerbating the problem.

Clearly, other solutions are needed, and a few have been proposed. Let us very briefly consider some wireless coexistence schemes.

"Listen Before Talk" & CSMA/CA Schemes

As we mentioned earlier, the unlicensed Personal Communication Services band employs a 'Listen Before Talk' *etiquette* to avoid congestion. An etiquette is merely a framework of rules governing access to airwaves, to which all devices using the

[10] These and other examples are in Peha, J.M. "The Path Towards Efficient Coexistence", April 2000.

band must comply. Devices in the UPCS band avoid collisions by sampling the airwaves and deferring transmission until they are clear: each device waits until no one else is talking, thus avoiding interference.

The widely popular IEEE standards for wireless networking, 802.11 and 802.11b use an essentially identical scheme on the ISM band – **C**arrier **S**ense **M**ultiple **A**ccess with **C**ollision **A**voidance[11]. As in Ethernet, each device that wishes to transmit first checks the medium to see if another node is talking. If it finds that the channel busy, the node waits for a random time before trying again; if it is free, the transmission can proceed.

Such schemes do successfully avoid interference, and are currently used in practice. However, they neither provide a disincentive for greed, nor do they address the issue of economic efficiency: they thus **fail to meet criteria (2) & (3)** above. Indeed, the potential for a Tragedy of the Commons with such schemes has been repeatedly demonstrated[12].

For example, consider a wireless node following such a 'Listen Before Talk' etiquette. Whenever the device wishes to transmit a packet, it has to wait for an appropriate monitoring time to verify that the channel is indeed free. When it is done transmitting it should relinquish the channel. However, the device may decide to hold on to the channel even if it has *nothing* to send – thereby saving its future packets the monitoring period delay. By doing so, however, the node is wasting spectrum and preventing others from transmitting. Indeed, it may selfishly hold the channel for the maximum period permitted by the channel etiquette. All the while, other devices will be building up longer and longer queues of packets to be sent. When our greedy device finally does relinquish the channel, there is likely to be a much longer wait before it can access it again[13]. Studies have shown that while such greed may benefit the device, it always worsens the performance of all others[14]. Struggling to regain performance, they too are forced to become greedy – paving the way for a Tragedy of the Commons. Clearly then, we must look to alternative schemes for our solution.

Penalty Schemes

As we have seen, the simple channel access schemes currently used fail to address the issue of greed, and are hence prone to tragedy. Recently Prof. Peha at Carnegie Melon proposed adding explicit incentives[15] to conserve bandwidth into the etiquette. The idea is simple: curb greed by imposing a *penalty* on devices, based on the amount of spectrum they consume.

For example, in a listen-before-talk scheme, a device that recently transmitted should have a lower priority to transmit again. Consider such a device that has just finished a transmission: normally, it could transmit again just as soon as the channel was idle. In Peha's penalty scheme, however, the channel would have to be idle for a 'penalty time' before the device would be permitted to transmit again. This delay would depend on how long the device had kept the channel busy during its last transmission: the longer it had held the channel, preventing others from transmitting, the longer it would have to wait for its next transmission. To effectively counter greed, the penalty imposed must be proportional to the amount of spectrum used.

Of course, assigning a penalty on subsequent transmissions tends to limit throughput, causing performance losses: a device cannot continuously transmit a stream of data, since it is forced to wait out its 'penalty time' even when the channel is freely available.

An optimal penalty must thus balance the

[11] 802.11 uses Collision *Avoidance* instead of Ethernet's Collision *Detection*. In wired Ethernet, a node starts transmitting when it thinks no one else is doing so. As it transmits, it listens to see if others are transmitting simultaneously – if so, it backs off for a random time and retries. In contrast, a wireless node cannot transmit and receive at the same time: its own transmission would drown out any incoming signals on the same frequency. Thus, it cannot *detect* collisions – they must be *avoided.* The transmitter thus starts by submitting a 'request to send' message. It then waits for a 'clear to send' from the receiver before sending date. Other nodes that hear the 'clear to send' message realize that there is about to be a transmission, and refrain from sending for the indicated length of time. This technique avoids (most) collisions – others are dealt with by using a random back-off time between retries. It also helps address the so-called 'hidden station' problem that wireless systems face: where *A* and *C* can both hear *B*, but cannot hear each other. We utilize such an RTS/CTS scheme in the final chapter.

[12] See, for example, Satapathy, 1996 or Peha, 1997.

[13] This example is from Satapathy & Peha, 1997

[14] Satapathy & Peha, "Etiquette Modifications" 1998, quoting Satapathy & Peha "Spectrum Sharing Without Licenses"

[15] Satapathy & Peha, *supra*

performance loss it causes, with the need to avert a Tragedy of the Commons. This optimal penalty will typically depend both on the population in the band, which determines the risk of a tragedy – and the application in use: streaming video, for example, cannot tolerate a large drop in throughput[16].

Penalty schemes avoid interference, and penalize greed – and are thus promising. However, as we have seen, the 'penalty time' imposed tends to limit throughput and cripples performance. Secondly, penalty schemes do not address the issue of economic efficiency: they provide no mechanism for nodes that need real-time access to the spectrum *more* urgently than others – say for video-streaming – to get that access. They thus **fail to meet criterion (3)** above.

Technology As a Panacea

A broader solution oft-touted for the spectrum-sharing problem envisages a future where new technology will make spectrum plentiful. Faced with such wide availability and low utilization, the possibility of a Tragedy of the Commons will dwindle. The need for licensing will also disappear, as devices will access the airwaves using frequency-hopping techniques that will allow them to find and allocate spectrum in real-time, without having an exclusive band set aside for them. This glorious future, however, is yet to arrive. The technology required for such a scheme is still prohibitively expensive, and further, would require the dismantling of existing licensing schemes that provide federal revenue. Additionally, the *demand* for wireless applications is increasing as rapidly as technological advancements have been increasing the effective *supply* of spectrum – and there are no indications of supply outstripping demand in the near future. Sharing schemes, thus, remain essential.

Economic Solutions

None of the schemes we have seen so far satisfies all three of our goals: the prevention of interference, the curbing of greed, and the economically efficient allocation of spectrum.

The essential problem concerns the right of access to a common resource: spectrum. Only one device can transmit at a time, and in doing so, it denies access to others in its vicinity. This is an allocation problem, and one that is economic in nature. An economic good has two fundamental characteristics: it is *excludable*, as people other than the consumer are excluded from enjoying its benefits – and it is *rival*, meaning that it is consumed and using it means that others can not. A sandwich is a typical example: only the person who eats it benefits from it, and in doing so, it is no longer available for others.

Consider the right to transmit at a particular moment in time. This right is a classic economic good. It is both excludable – transmitting exclusively benefits the device that does so, and rival – only one device[17] may transmit at that given moment. Our problem then, reduces to the allocation of this right, an economic good. Any economist will tell you that the most efficient method of doing so is the market price mechanism. The first one to do so was Nobel Laureate Ronald Coase, who in 1959 asserted that spectrum, like other resources should be allocated "by the forces of the market"[18]. Indeed a market mechanism can help meet each of our three goals:

1. **Interference is avoided**, simply because each device must buy the right to transmit. As we have seen, the right to transmit at a given time is an exclusive good, thus there can be no contention – devices will be able to coexist without suffering from mutual interference.

[16] Different functions penalize greed to differing degrees. As we have seen, the 'Listen Before Talk' etiquette, while offering excellent throughput, does nothing to curb greed. It is thus feasible only if isolated operation can be guaranteed. On the other hand, a linear penalty function, say one that forces the device to wait for as long as it previously held the channel, completely avoids greed – but causes throughput to drop in **half**. Such a drastic measure is only necessary if the band is over-populated and contention is very likely. Peha '98 has suggested a compromise: the penalty time should depend on the *square root* of how long the channel was previously held. This allows for high throughput, and can curb greed when utilization is low. However, a Tragedy of the Commons can still strike if utilization is high. As with all compromises, no matter which penalty function is chosen, it will be sub-optimal for some applications. This is yet another disadvantage of such schemes.

[17] In the case of frequency sharing systems, such as CDMA, a limited *number* of devices can transmit at a given time. Since the number of concurrent transmissions is restricted, however, the good remains rival: if a group of devices choose to transmit at a given moment, others cannot.

[18] Coase, 1959; in Benkler, *Overcoming Agoraphobia*, 1998

2. **Greed is curbed**, since each device *pays* for the right to transmit. As we have seen, the essential ingredient in a Tragedy of the Commons is the freedom to satisfy greed without penalty. Recall that penalty schemes attempt to suppress this freedom by introducing 'penalty time' delays. While such delays can reduce the chance of a tragedy, they also inflict substantial performance losses in the system. Economics tells us, however, that the ideal penalty is a market *price*, which provides a single, complete signal of how much others value the transmissions they are being forced to forego. Charging a price forces devices to conserve spectrum, and curbs rampant greed – without the attendant performance loss that other penalty schemes entail.

3. **Economic efficiency is ensured** since market-clearing mechanisms, such as auctions, ensure that the node most wishing to transmit is the one that gets access to spectrum. Indeed, there are auction mechanisms that are guaranteed to generate Pareto optimal results – even in the face of strategic behavior. We shall discuss this in detail in the next chapter.

We see then, that an economic market-mechanism has great potential to provide a robust, optimal solution to the problem of efficient coexistence in unlicensed spectrum. In the words of Coase speaking in 1959:

> …The allocation of resources should be determined by the forces of the *market* rather than as a result of government decisions. Quite apart from the misallocations which are the result of political pressures, any central agency which attempts to perform the function normally carried out by the pricing mechanism operates under two handicaps. First of all, it lacks the precise monetary measure of benefit and cost provided by the market. Second, it cannot, by the nature of things, be in possession of all the relevant information possessed by the managers of every business which uses or might use radio frequencies, to say nothing of the preferences of consumers for the various goods and services in the production of which radio frequencies could be used…
> [Coase, R. *The Federal Communications Commission*, 1959. Quoted in Hazlett, 2001]

In subsequent chapters, we explore economic solutions to the problem of coexistence. In doing so, we encounter a number of interesting issues: what mechanisms are suited for distributing the right to transmit? Should the mechanism employ actual money, or should money-equivalents such as transmission-right credits be substituted? Can secondary markets exist to trade such credits? How should such credits be renewed? Can such schemes be implemented in a distributed fashion, and what are their overheads and limitations?

This is a truly vast area, full of questions to chart and investigate. The goal of this work is to serve as an initial, exploratory step. Others are sure to follow.

II Pricing

"Mr. Justice Frankfurter seems to think that federal regulation is needed because radio frequencies are limited in number and people want to use more of them than are available. But it is a commonplace of economics that almost all resources in the economic system (and not simply radio and television frequencies) are limited in amount and scarce, in that people would like to use more than exists… It is true that some mechanism has to be employed to decide who, out of many claimants, should be allowed to use the scarce resource. But the way this is usually done in the American economic system is to employ the price mechanism, and this allocates resources to users without the need for government regulation…"
[Coase, R. *The Federal Communications Commission*, 1959. Quoted in Hazlett, 2001]

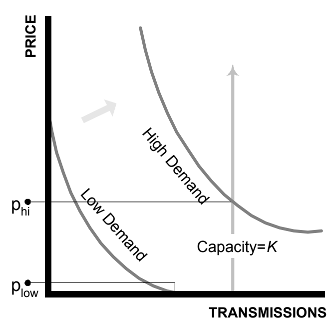
One of the primary concerns of economics is the allocation of scarce resources. A scarce resource is one for which demand exceeds supply when its price is zero; virtually all the products we use in our everyday lives are scarce resources. Over centuries, economists have developed, refined and perfected a solution to the allocation of scarce resources, namely markets. There is a vast volume of research that shows that markets can reach optimal allocations with no central guidance; each node making individually selfish optimizations that lead to the greater good. Market mechanisms are distributed and require minimum coordination, since each node makes its own individual decisions. They can respond dynamically to correct disturbances in demand or supply, and have enjoyed great success in the real world since time immemorial. All of this encourages us to design a market-mechanism for our particular allocation problem: that of the right-to-transmit.

Ideally, users consuming a resource should pay a price representing the costs that their usage creates. This results in intelligent utilization of resources, and is an essential requirement for an optimal distribution. In the context of wireless systems, the primary cost that needs to be addressed is the *social cost of creating delay for other nodes*.

Recall that a Tragedy of the Commons occurs when each node fails to take into account the social cost its transmission imposes on others. Ideally, the node that most wishes to transmit should be the one to do so, but if its transmission delays or prevents other nodes from transmitting, it should pay a *price* equivalent to the cost it imposes on these users.

Consider a wireless network that uses CDMA or similar techniques to support K simultaneous transmissions. While the network is uncongested – when the number of concurrent transmissions is less than K – an additional user does not cause any inconvenience to the others. He should, therefore, be allowed to access the airwaves without penalty.

On the other hand, if the network is operating at or close to capacity, an additional user prevents some other node from transmitting. In this case, he should be charged a price for his transmission corresponding to the inconvenience it imposes on others. The more congested the network – the more the number of nodes that wish to transmit at a given moment – the higher this price will be, creating an automatic disincentive to squander bandwidth when it is more precious. On the other hand, if the network is not congested, and the node's transmission does not preempt or delay others, the price will be close to zero[19].



We can see this argument graphically in the figure below. When network utilization is low, nodes are quiescent and the demand for access to the airwaves can be met by network capacity. This results in the zero market-clearing price, p_{low} depicted in the figure above.

When overall network utilization is high, each node wishes to access the airwaves. Since capacity is limited, the nodes that get to transmit are the ones that most want to – those willing to pay the most. The high demand results in a high market-clearing price p_{hi}, which acts as a disincentive for excessive consumption.

[19] This section summarizes work on the economics of congestible resources by MacKie-Mason & Varian, 1995-96.

This, of course, is an extreme case where the capacity of the network is fixed, and transmissions beyond capacity are blocked. In the more general case, when additional transmissions merely cause delays, the result remains the same: the optimal price should reflect the marginal cost of the delays introduced by the additional transmission. Facing this price, each user transmits *only if the benefit from his usage exceeds the costs imposed by it on all others.*

Optimality of Pricing

For a more mathematical exposition of this principle[20], consider a network capable of supporting simultaneous transmissions, where each incremental transmission creates delay for all others[21]. A typical node in this network is 'greedy': it is happier when it consumes more network resources, such as bandwidth. At the same time, each node faces 'delay' due to congestion in the network, and this detracts from its individual welfare. This 'delay' should be viewed as the comprehensive cost of congestion – being delayed or dropped etc.

Thus, each node's utility function can be expressed as: $u_i(b_i^+, D^-)$, where b_i are the resources consumed by node i, and D is the total delay in the system. This delay, in turn, arbitrarily depends on the total load, $f(L)$ on the system, where:

$$L = \sum_{i=1}^{n} b_i \tag{2.1}$$

For simplicity, let us rewrite the utility function as a direct function of b and L. Thus, our utility function is:

$$u_i(b_i^+, L^-) \tag{2.2}$$

If there were no price for consumption, each node would greedily choose b_i to maximize its own *individual* utility. This yields the first order condition:

$$u_i'(b_i) = 0 \tag{2.3}$$

As we have seen, this is the current state of affairs in most wireless systems. Each node maximizes its own utility without regard to the costs it imposes on others, creating a Tragedy of the Commons.

Now consider the most *efficient* use of the network. A benevolent central planner would limit each node's consumption, b_i such that the total welfare of the system is maximized. His problem, thus, is to determine the set of *optimal* individual consumptions $b_1^* \cdots b_n^*$, which maximize the *sum* of utilities for all nodes.

$$\max_{b_1 \ldots b_n} \sum_{i=1}^{n} u_i(b_i, L) \tag{2.4}$$

Differentiating using the chain rule, and equating with zero:

$$\begin{aligned} &\frac{\delta u_i(b_i, L)}{\delta b_i} + \frac{\delta L}{\delta b_i} \cdot \sum_{j=1}^{n} \frac{\delta u_j(b_j, L)}{\delta L} \\ &= \frac{\delta u_i(b_i, L)}{\delta b_i} + \sum_{j=1}^{n} \frac{\delta u_j(b_j, L)}{\delta L} = 0 \end{aligned} \tag{2.5}$$

This finally results in n first-order conditions of the form[22]:

$$\frac{\delta u_i(b_i, L)}{\delta b_i} = -\sum_{j=1}^{n} \frac{\delta u_j(b_j, L)}{\delta L} \tag{2.6}$$

These conditions simply formalize our intuitive result that each node should consume resources until the marginal benefit from its usage equals the marginal cost it imposes on all the others. Our central planner then solves these n constraints, yielding the optimal individual consumptions, $b_1^* \ldots b_n^*$, which result in the most efficient usage of the network.

A *decentralized* method of attaining this efficient distribution is to set a *price* that each node must face.

$$p_{opt} = -\sum_{j=1}^{n} \frac{\delta u_j(b_j, L)}{\delta L} \tag{2.7}$$

Where p_{opt} is simply the right-hand-side of equation (2.6): the marginal cost imposed on all nodes by an additional transmission. Note that this price is *independent of i* and is thus the *same* for all nodes.

---

[20] The theory in this section draws **extensively** on models in MacKie-Mason & Varian, 1995-96. We summarize their model and tailor it for our needs.

[21] We begin by considering the most general case. The results derived are then applied to the simpler case of a network with fixed capacity and no strictly incremental inconvenience.

[22] Note that all terms $\left.\frac{\delta u_j(b_j, L)}{\delta b_i}\right|_{i \neq j}$ are zero

Suppose each node is now charged p_{opt} for the network resources it consumes. Each individual maximization problem thus becomes:

$$\max_{b_i} u_i(b_i, L) - b_i p_{opt} \tag{2.8}$$

Which implies the first order condition:

$$\begin{aligned} &\frac{\delta u_i(b_i,L)}{\delta b_i} + \frac{\delta u_i(b_i,L)}{\delta L}\cdot\frac{\delta L}{\delta b_i} - p_{opt} = 0 \\ &= \frac{\delta u_i(b_i,L)}{\delta b_i} + \frac{\delta u_i(b_i,L)}{\delta L} - p_{opt} = 0 \end{aligned} \tag{2.9}$$

The middle term represents the delay imposed on node i's transmission by its *own* usage, and is zero[23]. Thus, each node i automatically limits its consumption b_i to satisfy:

$$\frac{\delta u_i(b_i,L)}{\delta b_i} = p_{opt} = \sum_{j=1}^{n}\frac{\delta u_j(b_j,L)}{\delta L} \tag{2.10}$$

Which is the **same as** the central planner's optimal allocation in **(2.6)**

Each node consumes resources until the marginal benefit from its usage equals the price it faces – which is set to the total inconvenience caused to others in the system. We can see, thus, that *charging* each node for the social costs its transmissions create results in the efficient usage of the network.

Optimality of Pricing: Specifics

Let us now consider the more specific case of a system with a fixed capacity for K simultaneous transmissions. An 802.11 cell, for example, has a capacity of 1 – only one node may transmit at a time. While such a network is operating below capacity, the inconvenience caused by an additional transmission is *zero*. When the network is at capacity however, any additional transmissions are blocked – only K concurrent transmissions can continue, and this imposes social costs on the nodes that cannot transmit.

Since each node only makes the decision of whether to transmit or not, each b_i is now a binary variable, $b_i \in [0,1]$. Each node's utility function $u_i(b_i, L)$ is positive for $b_i = 1$ and zero otherwise: a node only gains if it transmits[24].

Without a price, each node promptly maximizes its individual utility by simply setting $b_i = 1$. This, of course, quickly causes network capacity to be exceeded, setting the scene for a Tragedy of the Commons. A benevolent central planner, on the other hand, would seek to maximize the total welfare in the system:

$$\max_{b_1 \ldots b_n} \sum_{i=1}^{n} u_i(b_i, L) \tag{2.11}$$

Which is the same as (2.4). In this particular case, however, the solution to this maximization is quite simple. The central planner can only choose K nodes – since the total network capacity is K. To maximize welfare, these should be the nodes that gain the most 'happiness' from transmitting. He therefore simply sorts all the $u_i(b_i, L)\big|_{b_i=1}$, and chooses the highest K – setting b_i to 1 for each such node.

Our preferred solution is to finesse the central planner and to arrive at the same solution using a price. Recall that this ideal price should reflect the inconvenience caused by a node's transmission.

$$p_{opt} = -\sum_{j=1}^{n}\frac{\delta u_j(b_j,L)}{\delta L}$$

While the network is operating below capacity, an additional transmission has no social cost, and should thus be priced at zero. When demand is high and the network is at capacity, a transmitting node blocks some other node from transmitting. Thus, the inconvenience caused is the utility foregone by this $K+1^{th}$ node.

$$p_{opt} = \begin{cases} 0\big|_{n \le K} \\ -\displaystyle\sum_{j=1}^{n}\frac{\delta u_j(b_j,L)}{\delta L} = u_{K+1}(b_{K+1}, L)\big|_{n>K} \end{cases} \tag{2.12}$$

Facing this price each node's individual maximization is, as before:

[23] Even if this were, for some inexplicable reason, non-zero the definition of p_{opt} in (2.7) shows that p_{opt} dominates over this middle term for large n.

[24] $D = f(L) = f(\sum_{i=1}^{n} b_i)$ impacts utility only when $\sum_{i=1}^{n} b_i > K$. If usage exceeds capacity, there is no utility from transmission.

$$\max_{b_i \in [1,0]} u_i(b_i, L) - p_{opt} \tag{2.13}$$

Referring to (2.12), we see that when demand is low and fewer than K nodes wish to transmit, each is free to do so at no cost – maximizing welfare. When demand is greater than K, each node transmits only if:

$$u_i(b_i, L) > u_{K+1}(b_{K+1}, L)\Big|_{b_i, b_{K+1}=1} \tag{2.14}$$

Once again, this is the same as the central planner's solution: the K nodes that gain the most from transmitting are the ones that get to do so: welfare is maximized, resulting in the most efficient use of the network.

The Elusive Price

We have seen that a transmission price based on the inconvenience caused to other nodes results in welfare maximization and the optimal use of network resources. We have said nothing so far about *how* this price should be determined and levied.

The optimal price is extremely dynamic[25]; it depends on the social costs a transmission creates, and these vary considerably with time. An interfering transmission at a moment where another node is accessing real-time data, for example, is far more inconvenient relative to when it is merely checking email.

One simplistic approach[26] would be to construct a 'congestion map' of the network, tabulating congestion levels at different times. A fixed price could then be charged, depending on the time of day. An obvious problem with such posted prices is their inflexibility: they cannot adapt to rapidly changing network conditions. As we have seen, if the network is under-utilized, a transmitting node causes no inconvenience to others. The optimal price, thus, is zero. Using posted prices however, the node will be charged the standard fixed rate, which is inefficient. Similarly, when demand exceeds capacity, nodes willing to pay more for access may not get it, while others with a lower willingness to pay continue to transmit at the fixed price.

The posted price is *constant*, and thus cannot adjust to reach an economically efficient equilibrium. In general, since there is little hope that a posted price will reflect the true inconvenience faced by other nodes: it will invariably result in sub-optimal allocations, though it is likely to be an improvement on the status quo.

Indeed, the posted price scheme is representative of another, more fundamental problem – the lack of omniscience. No *centralized* arbiter can ever hope to gauge the individual inconvenience imposed on each node at a given moment. Ever since Adam Smith's analysis of markets in the late 18th century, it has been known that distributed self-optimizing agents perform resource allocation more efficiently than centralized systems. The nodes themselves, after all, are the ones that best know how much inconvenience they face[27]. What is needed is a method that allows them to use this *local* information to determine the optimal price in a more distributed fashion.

One such mechanism is the time-honored tâtonnement process[28], devised by Esprit Léon Walras. The tâtonnement, literally 'groping', mechanism begins by setting an arbitrary tentative price, which is broadcast. Nodes examine the tentative price, and indicate if they wish to transmit. If the tentative demand exceeds the capacity of the network, the price is adjusted upward; if there is excess capacity, the price is lowered. In this way the market 'gropes' towards price equilibrium in an iterative manner. Once the final price is reached, transactions can commence. Since competitive markets are well known to be efficient, the tâtonnement process should eventually reach an optimal price.

Tâtonnement schemes, at least in the classical sense described above, are plagued by a number of issues that make them cumbersome in practice. Users must first observe the current price and indicate whether or not they wish to transmit, and they must do so for a *number* of rounds till the price settles and they are allowed to proceed. This incremental process is time-consuming and inefficient, requiring multiple rounds of communication between nodes and the clearing-

[25] This section draws upon MacKie-Mason & Varian 94, 95, 96
[26] From MacKie-Mason & Varian, 1996.

[27] Coase himself makes this point far more eloquently in the quote at the end of Chapter I
[28] This discussion on tâtonnement is based on MacKie-Mason and Varian, 1996.

house. Under some conditions the tâtonnement can fail to yield converge to an efficient price. Nevertheless a suitably modified tâtonnement scheme is useful, and we review such an approach in the next chapter.

Auctions

> Auctions are one of oldest surviving classes of economic institutions…As impressive as their historical longevity is the remarkable range of situations in which they are currently used…
> [Milgrom, R. *Auction Theory*. Advances in Economic Theory, 1987]

The other popular mechanism for determining prices is, of course, the auction: one of the oldest forms of market. Herodotus describes the auction of women in Babylon, circa 500 B.C. Ancient Rome used auctions regularly to sell property, booty and in one famous case, even the Empire itself[29]. In 7th century China, the property of dead monks was auctioned off to the highest bidder[30]. Indeed, auctions have been favored throughout history as efficient methods for allocating resources. Their popularity has only increased with time: with millions of goods being auctioned off everyday, auctions are now widely recognized as extremely efficient economic mechanisms. The United States Treasury alone, for example, sells billions of dollars of promissory notes at auction.

Auctions come in an array of flavors, classified by the rules of bidding. We briefly consider the general types of auctions to determine which are best suited for our particular application.

English Auction

English auctions are the ones that most people are familiar with. Nodes compete by successively raising their bids. When no one is willing to bid further, the good is awarded to the highest bidder. Each node's strategy is to bid slightly higher than the current highest bid, and stop whenever the price exceeds its personal valuation of the item. Clearly, an English auction results in a Pareto efficient outcome: the good is awarded to the person who values it most. It should also be apparent that the price[31] the winner pays is marginally higher than the valuation of the *second-highest* bidder.

The primary problem with the familiar English Auction is its iterative nature. The overhead of repeated communication between the nodes and the auctioneer, and the long time to convergence make it ill-suited for determining prices for network consumption. Throughput suffers, as the winner has to wait for the auction to end before transmitting.

Dutch Auctions & Strategic Issues

In the Dutch auction, the seller continually lowers the asking price until one of the bidders wins the item. Since the auctioneer can reduce the price swiftly, Dutch auctions can proceed at a rapid pace; they are often used to sell perishable items such as fresh flowers or produce. Dutch auctions, however, are susceptible to strategic behavior that can result in sub-optimal allocations. Since an agent wants to win the auction at the lowest price possible, his bids will be dictated by his *beliefs* on how much others value the item. Strategizing with erroneous beliefs can lead to inefficient[32] outcomes.

Suppose that *A* values a good at $100 while *B* values it at $75. An optimal allocation must clearly award the item to *A*. Now suppose *A* erroneously believes that it is only worth $70 to *B*; *A* thus plans to wait till just before the price drops to $70 before making his bid. By then, however, it is too late: *B* will have claimed the item at $75. In general, there is no *guarantee* that the item will be awarded to the bidder with the highest valuation, though it is likely in practice, since gains from strategic behavior fall rapidly with the number of players.

Both English and Dutch auctions involve iteration, and are thus unsuitable for real-time network

[29] Didius Julianus' winning bid of 6,250 drachmas made him Emperor of Rome in 173AD (as quoted in Reynolds, K. 1996)

[30] All historical references and background from Milgrom. P. 1989 and Cassady, R. 1967. The description of auctions largely draws upon graduate textbooks, such as Varian *Intermediate Microeconomics*, as well as the sundry auction papers in the bibliography.

[31] Say *A* values an item at $100. *B* values it at $75. Once the bid reaches $75, *B* doesn't raise, leaving *A* free to claim it at $76.

[32] If nodes do not engage in speculation – or if do, and their beliefs are accurate – all the auctions described here yield efficient outcomes. Expected gains from strategizing tend to fall rapidly with the number of players, thus strategic issues are less pertinent for networks, which typically have more than 3 or 4 nodes. Furthermore, it may be viable to prevent fully computerized nodes from strategizing altogether.

purposes. We now consider the two auctions that we use throughout this work, the *first-price* and *second-price* auctions.

First-Price Auction

In a first-price sealed bid auction, each node submits a single bid without knowing the bid of others. The item is awarded to the highest bidder. Sealed bid auctions thus avoid wasteful iteration, but are susceptible to issues similar to those that plague their Dutch counterparts, viz. there is an incentive to strategize. The optimal bid for each node depends on its beliefs about the valuations of other nodes: if those beliefs are inaccurate, the good may be allocated to someone who does not strictly value it the most. As we mentioned above, however, gains from strategic behavior fall rapidly with the number of bidders, thus the possibility of inefficiency is only relevant for small systems with only three or four players. First-price auctions are a viable option for wireless systems, which are likely to have a reasonable number of nodes. Moreover, their simplicity makes them particularly well suited for distributed, decentralized applications, as we shall see in the final chapter.

Vickrey Auction

Of the auctions we have seen so far, the English is the only one that absolutely guarantees a Pareto-optimal allocation. Its primary shortcoming however, is its iterative nature: what we ideally need is an auction that only requires nodes to submit a *single* bid, yet provides the correct incentives to ensure they bid truthfully, without engaging in destructive strategic behavior.

Such an auction is a Vickrey auction, named after William Vickrey, winner of the 1996 Nobel Prize in Economics, who described it[33] in the 1960s. As in the sealed bid auction, each bidder is unaware of the bids of his competitors. As usual, the good is awarded to the highest bidder, but he is charged the *second-highest* bid. Note that if each node bids truthfully, the result is identical to that of the English auction discussed above: the node with the highest valuation receives the good, and pays the valuation of the second highest.

We have seen, however, that nodes in small sealed-bid auctions may not always bid their true valuations: there is an incentive to guess what the next-highest bid will be, and to bid only slightly more. Should the guess be wrong, there can be trouble. In a Vickrey auction, however, it is optimal for each node to bid its *true* valuation. Intuitively, an agent never under-bids: doing so only lowers the probability of winning, but does not affect the amount paid should it win. Over-bidding, similarly, is a bad idea. If an agent has to overbid to win, someone else must have placed a bid higher than his valuation. If he overbids and wins, he will end up paying more for the good than its value to him. Thus both over-bidding and under-bidding are counter-productive; honesty is always the best policy[34].

A second-price sealed bid auction, thus, results in an efficient outcome: the agent who most desires the good receives it. Moreover, the auction provides the correct incentives for users to bid their true willingness to pay. And it does all this without costly iteration. In general, should multiple goods need to be allocated, the second-price auction can be extended to a $K+1^{th}$ price auction. Analogously, the K highest bidders receive a unit each, at the $K+1^{th}$ price. Such auctions have similar efficiency and truth-telling incentives.

We have already seen that the price for a transmission should reflect the inconvenience it causes to others. Recall that for a network with the capacity for K simultaneous transmissions, this inconvenience is the utility foregone by the $K+1^{th}$ node, which is forced to relinquish its transmission. Consider equation (2.12), summarized below:

$$p_{opt} = u_{K+1}(b_{K+1}, L)$$

[33] Vickrey, W. "Counter-speculation, auctions, and Competitive Sealed Tenders," March, 1961

[34] More formally, consider the simple case of two bidders who value a good at v_A and v_B, and submit bids b_A and b_B respectively. Let us assume that $v_A \geq b_B$; A values the item more than B's bid for it, which is the price A will face. A, thus, wishes to *maximize* his probability of winning, $p(b_A \geq b_B)$. Since $v_A \geq b_B$, he can make p=1 by setting $b_A = v_A$ and bidding his true valuation.

Now suppose that $v_A < b_B$; if he wins, A would have to pay more than he values for the item. He thus wishes to *minimize* his chances of winning. Since $v_A < b_B$, he can make $p = 0$ by setting $b_A = v_A$, again bidding his true valuation. Thus the optimal strategy for a bidder in a Vickrey auction is to always bid truthfully. This simple textbook example is from *Intermediate Microeconomics*. A more thorough treatment can be found in Varian, H. 1994.

Transmitting would have given the $K+1^{th}$ node a utility of u_{K+1}. The welfare-maximizing price is the dollar valuation of this utility. But this is also the clearing price for a $K+1^{th}$ Vickrey auction!

To see this, consider holding a $K+1^{th}$ price Vickrey auction for the right-to-transmit in a network with the capacity for K concurrent transmissions. The auction would grant the K highest bidders the right-to-transmit, charging them the $K+1^{th}$ highest bid. If less than K nodes wish to transmit, no one is inconvenienced; as we have seen, the optimal price for access is then zero. Correspondingly, since less than K nodes bid, the $K+1^{th}$ bid in the Vickrey is also necessarily zero.

Now assume that there is more demand than capacity. An additional transmission must necessarily prevent someone else's, and this preempted node suffers inconvenience. As we have seen, in a Vickrey auction each node truthfully bids exactly what the item is worth to it. The $K+1^{\text{th}}$ node, thus, will bid u_{K+1} - the benefit it would reap from transmitting, and this will be the clearing price for the auction. But this is exactly the same as the welfare-maximizing price in (2.12) above.

We see, thus, that the auction always results in the *optimal price and allocation*.

Observations

The Vickrey auction has a number of properties that make it particularly well suited for allocating the right to transmit:

- Auctions provide for an economically distributed solution: there is no need for an omniscient central planner. Nodes make their own decisions on how to bid based on purely *local* information such as their private valuation of goods.

- The auction yields the optimal price and a welfare-maximizing allocation, and does so without costly iteration that would delay transmissions and waste network bandwidth.

- There are no perverse incentives in the auction; it is the dominant strategy for each node to bid truthfully, and thus no effort is wasted strategizing in attempts to fool the system.

- Computation complexity is limited. A $K+1$ Vickrey auction essentially amounts to determining the largest K bids, which that can easily be done in $O(n \log K)$.

- The auction is information efficient. The only communication involves exchanging bids and prices between the nodes and the market[35]. This is particularly important for wireless networks, where bandwidth is precious. Indeed, it has been shown that market price systems, such as auctions, "*minimize* the dimensionality of information required to determine Pareto optimal allocations."[36]

- Finally, the auction results in the classic, competitive 'supply equals demand' market-clearing equilibrium that is known to have a host of additional nice economic properties; some of which we shall call upon later. Note also that the equilibrium price at any point is the amount the *marginal* user bid. This means that each infra-marginal user enjoys a consumer surplus – he is paying less than he would be willing to, which increases his welfare[37].

A Note on Bandwidth Auctions

In addition to being used for allocating the indivisible right to transmit, a generalized form of the Vickrey auction can be used to allocate *bandwidth* shares, using the same inconvenience-compensation principle we have espoused. For example, such auctions could be used for wholesale allotments of spectrum amongst competing service providers. The design of such an auction is discussed in Lazar & Semret's seminal *"Design and Analysis of the Progressive Second Price Auction"*

[35] Wellman, M. 1998
[36] Wellman, M. 1998. See also J.S Jordon "*The competitive allocation process is informationally efficient uniquely*"
[37] MacKie-Mason & Varian, 1996

III Usage-based Pricing

> …All users of those spectrum bands should pay an access fee that is continuously and automatically determined by the demand and supply conditions at the time, i.e. by the existing congestion…
> [Noam, E. *Taking the Next Step to Open Spectrum Access*, 1995]

We now have all the elements in place to begin considering the problems of coexistence. True to economic tradition, we begin with a highly simplified 'model' of the problem. We abstract away technical complexities and some very real implementation issues, to try and gain insights and grasp essential concepts. We will attempt to address such concerns in due course.

Let us start by outlining a simple, fee-based approach to the problem. Consider granting control over the spectrum to a central 'clearing-house', which would allow nodes to acquire access in real-time. We assume that:

- A maximum of *K* simultaneous transmissions can be supported.
- All nodes can communicate with the clearing-house to place bids.

By doing so we are examining the problem in its simplest incarnation: that of a single access point controlling a set of nodes that collectively interfere with each other. In this idealized world, each node would negotiate a separate price for each transmission depending on usage and congestion.

The price for each transmission will be determined by a *K+1* Vickrey auction. Each node submits a bid to the clearing-house, which grants explicit permission to transmit to the winning node(s). Such a scheme can also be viewed as an extension of the licensing mechanism. A license is granted to a central agent, who then 'rents' the license out to transmitting devices for a fee.

Consider an 802.11 cell as an illustrative example. In this case, clearing-house functionality is best implemented in the hub. This could be accomplished using a MAC protocol similar to DQRUMA or DSA, but the technical details of how this is accomplished are not pertinent here; we focus on the economics and defer consideration of how this could be done to a later point.

Since this is an 802.11 cell, *K* is 1. Only one node may transmit at a time[38]. The highest bidder gets to transmit, and is charged the second-highest bid. When the network is not busy, the transmission would generally be uncontested – there will be no competing bids. Our winning node will thus be able to transmit without being charged.

When the network is heavily loaded, there will be many competing bids. The highest bidder will get to transmit, but will be charged the second-highest bid, which will be high. Thus, when resources are constrained, they will go to the node that wants them most desperately. At the same time, the price charged will be higher when there is more demand – and hence more congestion – thus acting as an automatic stabilizer.

We have seen that the free-for-all, random access schemes prevalent today are clearly inefficient: delay is far more costly for some nodes than others. They are also susceptible to greed and the Tragedy of the Commons. Other, centralized allocation schemes *sans* pricing are often subject to the biases of planners, who have to judge the social 'value' of a transmission. In the words of MacKie-Mason, charging a price "allows users to decide for *themselves* whether their transmissions are more or less valuable"[39] than the social costs they engender. Indeed "pricing directly provides the information needed to allocate scarce resources to users who value them the most. There is no need to arbitrarily assign priorities, or to force users with higher valuations to…be stuck behind low-value users[40]"

In such a usage-based scheme, the price charged depends purely on congestion, and thus gives each device an incentive to conserve spectrum and curb

[38] Hubs can be set to three different frequencies in 802.11b, but only one is active at a time. Neighboring hubs are typically set to different frequencies to avoid interference. If a different protocol were available that allowed multiple concurrent transmissions at different frequencies the solution would be identical to the one above – *K* would be simply be higher.
[39] MacKie-Mason and Varian, 1995
[40] MacKie-Mason and Varian, 1994

greed. We have already shown that the resulting allocation maximizes welfare. Indeed, this scheme satisfies all of the criteria we presented in Chapter 1: it prevents interference, curbs greed, and ensures economic efficiency.

However, it has a serious shortcoming: it involves money. The scheme requires complex infrastructure to allow wireless, often mobile, devices to perform real, monetary transactions with the agent. Devices would need to make bids, and keep account of real dollar payments. Secure systems would also be needed to transfer funds, and for ensuring that nodes actually *pay* for bandwidth they use. The costs of accounting, billing and policing each transmission could become prohibitive[41]. All this can lead to high transaction costs that would make such a scheme difficult to implement.

Issues with Usage-based Pricing

Even in the simplified context that we have presented, the scheme raises a host of interesting issues specific to usage-based pricing. We discuss some of these here[42].

Real-time Bidding

Since bidding is continuous in such a scheme, users cannot be expected to run it manually. The bidding system could be implemented as a separate, intelligent layer that would shield users from most of the complexities of bidding. A simple UI could allow them to set standard bids for transmissions from different applications, and override these defaults should circumstances warrant it. For example, real-time video or audio access could be assigned a higher priority than e-mail.

Variability of Price

Generally, users like to have semi-predictable prices, so that they can get a feel of how much a good will cost them. Since the clearing price depends on the $K+1^{th}$ bid, the price for a transmission is quite fluid, and the spot price may fluctuate. In our scheme however, this is not a major issue. A user's Vickrey bid sets the *maximum* that he could be charged for a transmission; he can be confident that the actual cost will never be higher. In addition, he will get virtually instant feedback should his bid be successful, so he should have no difficulty in making budgetary decisions[43].

Finally, if a node does not wish to bear the risk of price fluctuations, it can contract with a broker, such as the clearing-house itself, to purchase options contracts. When the contract is due, the brokers would fulfill it by bidding in the spot market. We discuss this further in Chapter VII.

Wealth Issues

MacKie-Mason and Varian state that pricing is often "opposed on the grounds that 'poor' users will be deprived of access[44]". They also point out that "this is not a problem with pricing itself, but with distribution of wealth". Wealth distribution problems can be addressed through standard non-distortionary economic means: taxes, subsidies or grants etc. Also, recall that when there is sufficient capacity, *no* price is charged. Thus users could avoid payment altogether, simply by setting their default bids to zero. Most of the time their transmissions will go through with reasonable delays, but when the network is congested, such users would 'pay' in units of delay, rather than money. Since many users do not require real-time access, they are capable of tolerating delays for applications such as e-mail.

Transaction Costs

The bane of usage-based pricing, at least at the granularity that we are considering, are the transaction costs raised by charging each transmission. These include the real-time delays introduced by soliciting bids, and the accounting and billing of transmissions. As the number of nodes increases, the clearing-house has to maintain an increasing number of accounts. Since nodes are actually expected to pay real money to the clearing-house, billing each individual transmission introduces major complexity. Each node must be billed, and be made to pay for its usage – as in the cellular system, which is quite complex. It may be possible to mitigate the problem using pricing based on statistical *sample* rather than dynamic current bids, though this would raise other optimality issues.

[41] Peha, J.M, 1998

[42] This discussion summarizes sections from MacKie-Mason & Varian, 1994 & 1995, who discuss similar usage-based issues in the context of *wired* Internet.

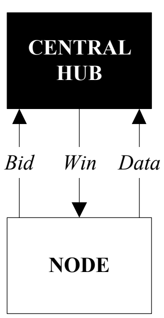
[43] MacKie-Mason & Varian *supra*

[44] MacKie-Mason & Varian *supra*

Another possibility would be to allow nodes to maintain their own balances, and report them at the end of the day to the node. Since these balances are in real dollars, however, this raises a whole slew of security issues – with corresponding transaction costs for addressing them.

Even if these sundry transaction costs can be lowered, money is not a feasible solution for many applications. Consider university wireless LANS, or home networks where there is no central agent who owns the rights to the airwaves, and it doesn't seem reasonable that there even should be one. One obvious solution is to use a money-substitute, such as transmission-credits. We do so in the next chapter, but for now we consider how the real-time latency of such a centralized scheme can be improved.

Reducing Overhead: The Tâtonnement Revisited



In the simple scheme we have described, the clearing-house must first solicit bids from every node, process them, and explicitly inform the winners that they may proceed to transmit. There are thus three phases in the system: a *request* phase, where nodes place bids; a *scheduling/permission* phase where the clearing-house determines the winners, debits their accounts, and informs them; and finally a *transmission* phase where winners can finally transmit data.

These overheads are similar to those in well-established demand-assignment MAC protocols designed to support QoS in wireless systems, such as *Distributed-Queuing Request Update Multiple Access* and *Dynamic Slot Assignment++* etc[45]. Satisfying QoS standards using purely random-access protocols is problematic, since each packet must compete with others to access the medium.

Protocols that guarantee a particular QoS function in a manner conceptually similar to our Vickrey scheme, and involve the same three phases: request, permission, and data transmission.

In Bell Lab's DQRUMA, for example, there is a separate random-access channel reserved for requests Nodes use this request-channel to send their QoS requirements to the base station. Having done so, a node monitors downlink activity from the base., waiting to receive its 'Packet Transmit Permission'. The base generates permissions based on an optimizing algorithm. Once a particular node receives permission, it can access the data channel without contention. If nodes are transmitting continuously, their subsequent requests for transmission are piggy-backed with data – reducing contention on the request channel. DQRUMA thus allows for efficient scheduling of various transmissions with differing QoS requirements. Moreover, while not addressing the problem of greed[46], DQRUMA does achieve acceptable throughput[47].

It should be apparent that the *overhead* in our scheme is comparable to DQRUMA and others of its ilk: the delay involved in the first two phases of receiving bids, and notifying winners. The performance of such protocols is encouraging, but it still worth considering what can be done to reduce overhead.

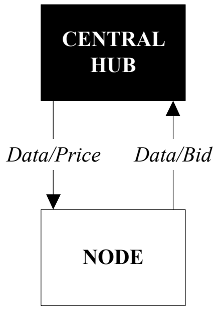
If it is felt that the latency caused by holding an auction for every transmission is too high, there is a straightforward solution: trade off some minimal economic efficiency and simply do not hold auctions for every single transmission.

Let us assume, as usual, that the network supports K concurrent transmissions and that there is a listen-before-talk or equivalent standard mechanism available for nodes to access channels. The hub holds an auction and broadcasts a price every once in a

[45] The following description is based on Gumalla, C.V. & Limb, J.O. 2000, who describe the DQRUMA protocol and its properties.

[46] DQRUMA, like other existing QoS protocols, **does *not* curb greed**. There is nothing to prevent nodes from arbitrarily requesting the highest quality of service for each and every packet, drastically degrading system performance. Indeed it is individually *rational* for each node to do so, resulting in the usual tragedy. Our economic scheme not only provides for efficient usage, but also curbs greed – it is always individually optimal for the node to conserve network resources and declare its true valuation for the packet.

[47] At light loads DQRUMA achieves the delay performance of slotted ALOHA, and at heavy loads it approximates an appropriately weighted round-robinesque algorithm favoring higher QoS traffic.



while, and subsequently *only* the nodes willing to pay that price compete to randomly access the channels. Thus nodes face no auction overhead most of the time, yet economic efficiency is (mostly) ensured.

More explicitly, say that the current asking price is p_0; this was determined by an auction and broadcast sometime ago. Any node that transmits pays this price. Let us assume that it has been a while since the auction, and demand has risen. There are now N nodes with bids higher than p_0, where $N > K$. These N nodes contend for the random-access channels, and necessarily only K of them succeed. Note that the allocation at this point is not strictly optimal. The K transmissions the hub is servicing are from high-bidders, but not necessarily the *highest* bidders; this is the primary source of economic inefficiency.

When should the hub update prices? The obvious solution would be to do so after a fixed period of time: a longer period would further reduce auction overhead, but at the risk of further economic inefficiency. There is, however, a better solution. The hub could require that each node that accesses a channel continue to head all data transmissions with a bid. By monitoring these bids, the hub could glean that the current asking price was becoming too inefficient and it was time to hold another auction.

Are there any incentive problems with this approach? Could clever nodes try and 'cheat' by setting false low bids in an attempt to save money? First of all, note that no node could submit a bid below the current asking price, p; since only nodes with bids higher than p can access the channel. Second, recall that if demand exceeds capacity at the asking price, there will be $N>K$ nodes contending for access and only a random K will succeed. The N-K nodes remaining will be unsatisfied, and at every transmission there will be a different set of N-K nodes who wish to transmit and are willing to pay more, but aren't able to. These frustrated nodes will respond by submitting successively higher bids. The hub can note these increasing bids as a signal that its asking price is too *low*, and readjust when the situation becomes dire enough. Conversely, too high a price can be detected simply if there are fewer than K transmissions; the fewer there are, the more the price needs to be reduced.

Note that all of this happens *continuously*. Since all transmissions are headed with bids, the hub can detect the need for a new auction, hold it, and simply piggyback the new price on one of the downlink transmissions. There is no more waiting while auctions are held, and no additional transmissions are required, greatly improving overall system overhead.

In effect, the solution described is a combination of a tâtonnement with an auction. Recall that the primary problem with a classic tâtonnement is the large wait as the price converges. The tâtonnement issues a tentative price, nodes respond with tentative demands, if total demand is more than capacity, the price is raised – otherwise it is lowered. No transmission takes place until the price settles, which can take multiple rounds. The difference here is that transmissions *can* occur, even if they are not charged the strictly optimal fee. The market no longer 'gropes' for the equilibriating price – it floats near it, and should it drift too far another auction is held which pins it back again[48].

[48] We stress that one should not be terribly concerned if a high-valuation node occasionally transmits before one that would have valued transmitting slightly more. The efficiency loss is infinitesimal, and is far less than transaction costs suffered by uncompromisingly pursing the optimal allocation. The point of pricing is to curb greed, and to ensure that when network conditions are congested, resources preferably go to the nodes that need them and are willing to pay. These objectives remain satisfied.

IV Token Economy: Overview

"Money is the root of all evil"
[The Bible, *I Timothy* 6:10]

We have seen that the major issue with usage-based pricing is transaction costs associated with the use of money. In a large usage-based pricing system, such as cellular, the accounting and billing issues may become severe enough to warrant shifting focus from end-users to wholesale applications.

There is, however, an alternative. We explore a usage-based pricing solution that seeks to avoid the issues of using money within the system by replacing cash with 'tokens'. Tokens are the equivalent of actual money *within* the market for wireless bandwidth; a token is essentially cash that can be used for one purpose, and one purpose alone: to bid and pay for transmissions.

The Token Economy

Throughout this work, we have stressed the justifications for using market mechanisms to allocate resources – no scheduling system or algorithm can truly duplicate the sophistication of pricing. Our problem, thus, is to continue to exploit the advantages of the market mechanism whilst avoiding the implementation overheads associated with using money.

Our solution is to finesse the problem and use a money-substitute, tokens. Tokens are entirely indistinguishable from electronic money, except that they are only useful within the wireless market. Tokens, like money, grant the ability to purchase a resources, and the price mechanism can still be used to allocate these efficiently. This concept of artificial, 'private money' is hardly unprecedented; Waldspurger[49] describes a number of microeconomic schedulers that use similar fake currency[50] to bid in auctions for resources. Notably, the *escalator algorithm* proposed by the seminal Agoric papers[51], and the *Spawn*[52] system, all rely upon price mechanisms that use 'fake money', and these have all performed well in practice.

Intuitively, by replacing money with tokens, we can use the usage-based pricing scheme exactly as before. We continue to determine and charge the winners for their transmissions based on *K+1* Vickrey auctions, or our mechanism of choice, but we do all this in the currency of tokens.

Consider a node that values being able to transmit both now, and in the future. Since the only way it can transmit is by paying in tokens, it has an incentive to treat them like cash: bidding too much now would mean too little left for later transmissions. In essence then, tokens are identical to electronic cash. Where do nodes get this cash? One simple approach is to provide them with some income at regular intervals. There are a number of other possibilities, and we consider sundry funding strategies in a later chapter.

Such an approach has a number of advantages. Nodes are free to decide, individually, how much they value their transmission and bid accordingly. Moreover, since we are creating an artificial economy, we are free to influence it in any manner we deem appropriate. Thus we can determine budget constraints, control each node's individual income, set taxes, mandate transfers etc. etc. We have a vast array of economic instruments at our disposal, and this allows for unprecedented configurability of the system.

Framework

In our economy, nodes have an endowment of tokens that they use to place bids and vie for transmission rights. A node gains happiness only from transmissions. It is useful to draw a loose parallel to the well-known economic problem of inter-temporal consumption: that of an agent trying to decide how much of his wealth to consume now, and how much in the future.

[49] Waldspurger, C.A. "Lottery and Stride Scheduling", 1995
[50] Drexler & Miller "Incentive engineering…" 1988.
Ferguson, D. et al. "Microeconomic algorithms…" 1988.
Ferguson, D. "Application of Microeconomics to the Design of Resource Allocation and Control" 1989.
Wellman, M. "Market-oriented programming…" 1993.
Cheng & Wellman "The WALRAS algorithm" 1998.

[51] Drexler & Miller "Markets and computation: Agoric open systems." 1988
[52] Waldspurger, C.A. et al. "Spawn: A distributed computational economy." 1992.

Consider such a node which wishes to maximize utility by choosing how much of its endowment to bid in each period. This optimal bid b_t^* determines the probability of success p for a transmission, and the node values this probability of success. We formalize this in the next chapter, but roughly the node wishes to determine the optimal bids to maximize overall utility:

$$\max_{\{b_t\}} \left\{ u_t(b_t) + u_{t+1}(b_{t+1}) + u_{t+2}(b_{t+2}) + \ldots \right\} \quad (4.1)$$

(4.1) is a simplification of the node's true problem, but for now it serves to build intuition. In performing this maximization, the node is subject to a constraint: it cannot bid more than it has. This budget constraint can take many forms. For example, if the node were granted an initial wealth W_0, it would perform the maximization above relative to the constraint:

$$W_{t+1} = W_t - \Phi(b_t)\big|_{0 \le b_t \le W_t} \quad (4.2)$$

Where $\Phi(b_t)$ is the Vickrey price the node expects to pay, given its bid of b_t.

Thus, there are two distinct problems that need to be considered. The optimization problem of (4.1) – that of determining the optimal bids in terms of tokens, which have purely nominal value – and the problem of choosing a *funding strategy* that will determine the form of the budget constraint in (4.2). In the following two chapters we discuss each of these issues in detail. Here we present a general overview, and discuss a simple solution in the special case of secondary markets.

Bidding

Rational economic agents clearly have a desire to spread out their consumption over time. Intuitively, wealth is only useful for one thing: consumption today and tomorrow. Nodes, inasmuch as they value being able to transmit in the future, will wish to curb greed and conserve wealth.

In usage-based pricing, the user could specify that a transmission had a certain value *\$v* to him. Given the Vickrey auctions, the optimal bid was then also simply *\$v*. Living in the real world, the user had a very good estimate of the *value* of a dollar to him. Using this information, deciding, say, that a transmission was worth 10¢ was natural. With tokens however, things are no longer that simple. Bidding in tokens presents a unique problem: a node has very little information about the *value* of its endowment. How much is a token worth? If a node has a hundred tokens is it 'rich' and can it afford to splurge? Or does it need a thousand? Determining the optimal bid *in the currency of tokens* presents a number of interesting challenges and we examine these in Chapter V.

Funding

The simple constraint in (4.2) above represents granting a single, initial endowment W_0 to each node, which it consumes over time. The next period wealth is simply current wealth less current consumption. Alternative funding strategies could include granting a node a steady stream of income every cycle, granting it income every k cycles, or allowing winning bids to be redistributed amongst losers. Each of these alternatives has its own attendant advantages and disadvantages, and we discuss such schemes in Chapter VI.

Before we move on with our exposition, however, let us consider the special case of introducing perfectly liquid secondary markets.

Secondary Markets

Consider giving all nodes an equal, fixed endowment of tokens. This endowment can be in the form of a regular income stream, or a large initial endowment to be consumed over time. While equitable, such a funding approach may not be always be optimal. For example, consider two nodes *A* and *B*, both of which vie for control of a single transmission channel. Both *A* and *B* generate legitimate and useful transmissions, but *A* services more traffic, much of which is delay sensitive. *B*, on the other hand, generates sporadic e-mail traffic only.

With equal endowments, *A* is forced to budget its wealth and place lower bids than *B*. Simply put, *A* has more packets to send, and the *same* amount of tokens: thus, on average, it can afford to pay less tokens per packet. This means that *B*'s low-value traffic will likely win versus *A*'s whenever there is contention, but *B*'s traffic can tolerate delays, and *A*'s cannot. In effect, setting equal endowments for *A* and *B* corresponds to saying that the net value of all their transmissions is equal. Or, equivalently, that they are entitled to an *equal* share of network resources. This may often be the case for large, homogenous networks, but need not always be so, as it is here.

The administrative solution here, of course, is obvious. *A* should simply be granted a higher 'budget' than *B*, since it has to handle more legitimate traffic. But it could be argued that *A* should preferably be allowed to determine its *own* budget. Node *A* could buy tokens in a secondary market. If *A* thinks that it has to make more transmissions, it should buy more tokens, and since it must pay a price for these tokens, this will automatically curb greed. More importantly, secondary markets will also allow users to assign *value* to their tokens: since tokens are freely traded, there will be a market-clearing price for them. Thus a user will know that a hundred tokens are worth, say, \$5. This makes our bidding problem very simple: since tokens now have a real dollar valuation, it is now realistic to expect the user to specify a valuation for a transmission in terms of tokens. Recall the dilemma of a user who values a transmission at 10¢, but could not figure his bid in tokens. With a given market-clearing price for tokens his problem is trivial: his optimal bid is 2 tokens.

Indeed, secondary markets allow us to gain the benefits of usage-based pricing, whilst avoiding the problems of using money within the wireless system. All the accounting, billing and payment issues occur *outside* the system. And they do so infrequently: a user can buy tokens to last him for a month, or a year. Setting up secondary markets for a virtual item is not without precedent. Cell phone minutes aside, there are established secondary markets for characters and weapons for the popular online game EverQuest. Gamers buy and sell items, 'gold', and characters that can only be used within the game on E-Bay. Current market clearing price for a simple sword is about \$5. A high-level warrior character: \$100-\$600.

Secondary markets could be set up in a number of different ways. Users could explicitly *trade* tokens at clearing-houses. This, however, introduces some complexity into the system, since tokens must be conserved across transactions, and users must be able to add and remove them from their nodes. Tokens, since they have real dollar value, become electronic cash, with the security and enforcement headaches that attend it. Furthermore, tokens are *not* homogenous. Consider two networks, or bands, that have differing levels of demand. Each of these networks must have a *separate* market for tokens, and the clearing price for tokens for the high-load network will naturally be substantially higher. Thus tokens not only become artificial cash, they become artificial *currencies*, each with their own exchange rate to the dollar, and arrangements must be made to sell them separately.

An alternative to user-to-user trading is to require all nodes to directly purchase tokens from a band-manager[53], who acts as a 'central bank' and can produce more tokens. When a user utilizes a token, it is destroyed, just like a cell-phone minute. This would require users to have a method for adding to their existing token balance, but not subtracting. In addition, the revenue from the sale of tokens could go to the license holder for the band. A simpler, second-best approach is to allow wireless hardware manufacturers to sell equipment with different token 'funding-rates'. Demand for priority would then determine the clearing price for the equipment, and indirectly, for tokens; but with such a solution, users will not require the ability to modify their token accounts at all.

Clearly secondary markets could be useful, but can we do without them? Can we exorcise the demon of money entirely from our economy? We explore this question in detail in the following two chapters.

[53] Such a band manager would handle a given scope of interference. This could be the network administrator, in the case of an insolated network.

V Token Economy: Bidding

We posed two questions in the previous chapter: the problem of determining the optimal bid in the purely nominal currency of tokens, and that of determining how nodes get these tokens themselves. We consider the first of these here. The auction mechanism that we have devised ensures that a node bids its true valuation, but before it can bid a node must have a way to assess the 'value' of a transmission *in terms of tokens*. We begin by discussing simple bidding agents and build up an economic solution that allows nodes to accomplish this.

Bidding Agents

How should nodes decide what to bid? This fundamental question has not received much attention in existing artificial-currency systems. In *Spawn*[54], for example, the bidding strategy was simplistic. A job would receive a steady income stream, and would simply bid everything it had. This meant that if a job lost its bid, its wealth would increase and it would be able to bid higher the next time. Thus, a job would place successively higher bids until it won. While appropriate for the nature of *Spawn*, this simplistic bidding strategy led to the fluctuation in prices that Waldspurger et al reported. The system does not allow for consumption smoothing, which rational economic agents would naturally attempt.

What are the alternatives? One solution is to follow the approach of Steiglitz et al, in their work on artificial markets[55]. They posit using an arbitrary *bidding function* satisfying certain basic, common sense properties – which nodes use to determine their bids as a function of their wealth. Steiglitz et al report good behavior with such arbitrary bidding functions; asserting that the 'particular shape of the curve proves not to be critical'.

For us, such a bidding function would have the following attributes:

- Bids should be a fraction of the nodes current *wealth*, and should increase and decrease as the node becomes richer or poorer. They should also level off at some value.

- Bids should increase with the *priority* of the transmission. For example, transmissions generated by real-time applications should generate higher bids.

- Bids should increase depending on how *long* a transmission has been pending. The increase must not be indefinite, leveling off at some fraction of wealth for each priority class.

Recall our discussion from the section on usage-based pricing: fundamentally, nodes can either pay for a transmission in units of *money*, or in units of *delay*. The premises above merely state the common-sense observations that richer nodes can afford to pay more money and suffer less delay; that nodes will prefer to pay more to hasten delay-sensitive transmissions, and that nodes will react to transmissions that have successively failed and are suffering excessive delay by increasing their bids to expedite them.

Of course bidding functions don't necessarily need to follow these rules, though they are likely to do so in practice. Each user may specify an arbitrary bidding function for his node. In practice, it may be simpler to have a general bidding function with parameters that can be tweaked by the user, using an intuitive interface. One such simple bidding function could be:

$$B_t = [\overset{\min}{K_p} e^{-\alpha_p (t-t_0)} + \overset{\max}{K_p}(1-e^{-\alpha_p (t-t_0)})] \cdot W \qquad (5.1)$$

Where W is the node's initial wealth, $t-t_0$ is the time the transmission has been waiting, and $0 < \overset{\min}{K_p}, \overset{\max}{K_p}, \alpha_p < 1$ are all user-configurable variables. Note also that at all times, the first term is less than unity; thus the node can never bid more than its current wealth.

The resulting bidding strategy is quite simple: a node receives a transmission that has a priority *p*. Its initial bid is a fraction of its current wealth $\overset{\min}{K_p} W$; if the bid is unsuccessful, the node successively raises it to $\overset{\max}{K_p} W$ with an aggressiveness measured by the

[54] Waldspurger, C.A. et al. "Spawn: A distributed computational economy." 1992.
[55] Steiglitz, K. et al. "A computational market model based on individual action." 1996.

parameter α_p. Intuitively, packets that have being waiting for a while slowly gain priority. The user can pre-specify the parameters $\overset{\min}{K}_p, \overset{\max}{K}_p, \alpha_p$ for different classes of transmissions *p*, and the node will bid accordingly[56].

Each node can have this, or an entirely different bidding function. They will all compete to place bids in the economy and the price mechanism will automatically optimize the allocation. Additionally, the form of the bidding function can be improved incrementally using AI algorithms such as reinforcement learning. More intelligent bidding agents that strategically defer transmissions till later, when prices may be lower, will evolve as nodes try to squeeze the most out of tokens; and this is exactly what we *want*. We want nodes to conserve during congestion, and the token price system creates strong incentives for users to *create* such agents.

While such an approach may be better than *Spawn's* all-or-nothing method, it is still adhoc. A more systematic approach is to attempt to derive a bidding function from first principles, and this is precisely what we do in the next section. But first, some essential background.

The Value of Money

Recall that with secondary markets or usage-based pricing, the user could specify that a transmission had a certain value *\$v* to him. Since it is optimal for a user to bid his true valuation in a Vickrey auction, the optimal bid would also be *\$v*. Living in the real world, the user had a very good estimate of the *value* of a dollar to him. Using this information, deciding, say, that a transmission was worth 2¢ was natural. In a token economy, however, a node has very little information about the *value* of its endowment. How much is a token worth? What can it buy? If a node has a hundred tokens is it 'rich' and can it afford to splurge? Or does it need a lot more? Determining the optimal bid *in the currency of tokens* presents a number of interesting challenges and we examine these here.

Our problem, at a fundamental level, is simply this: tokens, in of themselves, do not have value. Only successful *transmissions*, both now and in the future, grant a node utility. Thus how is a node to decide how much to bid in terms of tokens?

This may seem intractable, but it actually isn't. Let us briefly consider the classic economic problem of inter-temporal choice: that of a consumer who wishes to determine how much of his wealth to consume over a period of time. The fact to note is, that just as in our token economy, *consumption* grants the consumer utility – wealth, in of itself, does not.

In the traditional set up, the consumer wishes to spend his initial endowment w_0 over his lifetime. A representative consumer wishes to maximize utility by choosing how much to consume, c_i^*, in each period:

$$\max_{\{c_t\}_{t=0}^{\infty}} \sum_{t=0}^{\infty} \beta^t u(c_t) \tag{5.2}$$

Here $\beta < 1$ is an arbitrary discount rate that reflects the consumer's myopia[57]. If β were 1, the consumer would value present consumption and future consumption equally.

In performing this maximization, the consumer faces a budget constraint: he cannot spend more than he possesses.

$$w_{t+1} = w_t - c_t \Big|_{0 \le c_t \le w_t} \tag{5.3}$$

Where w_0, the initial starting wealth, is given.
For all reasonable utility functions, such as $u = \ln(c_t)$ the maximization eventually results in the ubiquitous Euler equations:

$$c_{t+1}^* = \beta c_t^* \tag{5.4}$$

Maximizations such as (5.2), however, are difficult to perform in practice. Furthermore, no real consumer actually sits down and decides the optimal path of consumption over his lifetime. In practice, a rational consumer decides how much to consume based on

[56] While Steiglitz et al rely on common sense instead of a formal economic rationale in using such a function, this – roughly – corresponds to nodes having standard Cobb-Douglas utility preferences between the current transmission, and tokens to be used for future transmissions. Such preferences imply that node will always choose to spend a *constant fraction* of its wealth on the current transmission.

[57] β should generally be less than unity, since it is rational to prefer guaranteed consumption in the present, to expected consumption in the future.

his *wealth*, which he views as a proxy for consumption tomorrow. Thus we wish to find *c(w)*, the optimal consumption as a function of *wealth*.

Although we stated the problem above as choosing an infinite sequence for consumption and saving, the problem that faces the consumer in period *t* can be viewed simply as a matter of choosing today's optimal consumption, and tomorrow's starting wealth. The rest can wait until tomorrow. If a consumer continues to do this at every period, he should automatically attain the optimal path of consumption

Let *V(w)* be the value of having wealth *w* in hand. The consumer's problem, thus, can be restated as:

$$V(w) = \max_{c \le w} \{u(c) + \beta V(w-c)\} \tag{5.5}$$

This tells us that the total utility from the optimal path of consumption is attained simply by choosing the optimal consumption *today*, and following the same optimizing behavior in the future. This recursive form for the maximization problem is known as the Bellman equation, and is widely used in control theory and in advanced macroeconomics[58].

The value function *V(w)*, here has a very clear economic interpretation. It is the amount of utility a given level of wealth can purchase. Recall that wealth in of itself, even in this 'real' economy, has no value – its only use is its ability to purchase consumption. Formally then, *V(w)* is the total expected utility that can be attained from having wealth *w* and consuming it optimally in the future.
Note that this value function *V(w),* is *itself* the unknown. In the simple case of $u = \ln(c_t)$ for example, *V(w)* can be determined analytically[59]:

$$\begin{aligned} V(w) &= \frac{\ln(1-\beta)}{1-\beta} + \frac{\beta \ln \beta}{(1-\beta)^2} + \frac{\ln w}{1-\beta} \\ &= \kappa + \frac{\ln w}{1-\beta} \end{aligned} \tag{5.6}$$

Knowing *V*, one can solve for the optimal level of consumption for a given wealth, simply by substituting into (5.5), and maximizing. This yields:

$$c^*(w) = (1-\beta)w \tag{5.7}$$

The equation is the optimal *policy function* for our consumer: it determines how much he should consume, given his current wealth. (5.7) states that it is optimal for the consumer to spend a constant fraction of his remaining wealth *w* every period. Note that our recursive formulation has resulted in the **same** solution as the direct optimization in (5.4):

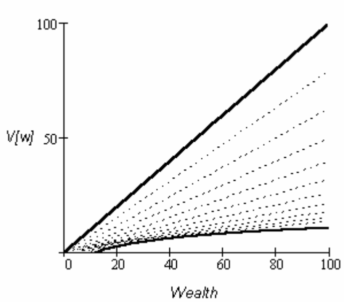
$$\begin{aligned} c_t^* &= (1-\beta)w_t \\ c_{t+1}^* &= (1-\beta)w_{t+1} = (1-\beta)(w_t - c_t^*) \\ &= (1-\beta)\left(\frac{c_t^*}{1-\beta} - c_t^*\right) \\ &= \beta c_t^* \equiv (5.4) \end{aligned} \tag{5.8}$$

Generally, however, *V* is complex enough that it must be estimated numerically, using dynamic programming, or value iteration. The recursive form of (5.5) lends itself directly to such an approach; starting from an initial 'guess' for *V*, repeated applications of (5.5) result in rapid convergence to the true value function. The optimal levels of consumption are automatically generated during the iteration. We outline the algorithm below.

$$\begin{array}{l} V \leftarrow V_{guess} \\ \text{while } (\Delta > \varepsilon) \\ \quad \left| \begin{array}{l} \Delta \leftarrow 0 \\ \text{for } (\forall\, w < w_{\max}) \\ \quad \left| \begin{array}{l} v \leftarrow V[w] \\ V[w] \leftarrow \max\limits_{0<c<w} \{\ln(c) + \beta \cdot V[w-c]\} \\ C[w] \leftarrow \arg\max\limits_{0<c<w} \{\ln(c) + \beta \cdot V[w-c]\} \\ \Delta \leftarrow \max(\Delta, |v - V[w]|) \end{array} \right. \end{array} \right. \end{array}$$

Value iteration works by maintaining two arrays, $V[w]$ and $C[w]$. $V[w]$ is initialized arbitrarily, and with each iteration, gets successively closer to its true analytic value. The algorithm runs till the incremental improvement in a round Δ, becomes negligible. When it quits, $V[w], C[w]$ contain the true utility of having wealth *w*, and the optimal consumption given wealth *w*, respectively.

[58] See, for example, Stokey, N. and Lucas, R. *Recursive Models for Macroeconomics* (1998).
[59] The proof is lengthy, irrelevant, and can be found in advanced graduate texts. See also Laibson, D. 2001.



The figure depicts the algorithm's first ten iterations for our example. We arbitrarily initialized $V[w]=w$; this is depicted as the 45º upper dark line. With each iteration, $V[w]$ gets successively closer to the exact, analytical logarithmic form predicted by (5.6): the lower dark line[60]. Note that by the tenth iteration, the value function has almost converged.

Bidding Agents: An Economic Approach

It is perhaps obvious, now, where we are headed. The technique of value iteration described above allows a rational consumer to derive the 'value' of having wealth *w*, and simultaneously provides him with the optimal level of consumption *c*(*w*).

If we view consumption as bids, and wealth as tokens, the parallel with our problem is clear. The analogy, of course, is not exact, since the simple model presented above is deterministic, whereas bidding, by its very nature, is stochastic. This, however, does not present any problems: almost all macroeconomic usage of value iteration involves stochastic processes. Value iteration thus, in one fell swoop, can provide nodes both with a means of discovering the 'value' of tokens, and a method for determining their optimal bids.

To explore this, we set up a simple model of the node's problem to aid in our exposition. We are more interested in the *technique*, than the specifics of this particular model. More complex and sophisticated models can be designed by building on the concepts we develop here.

A Simple Scenario

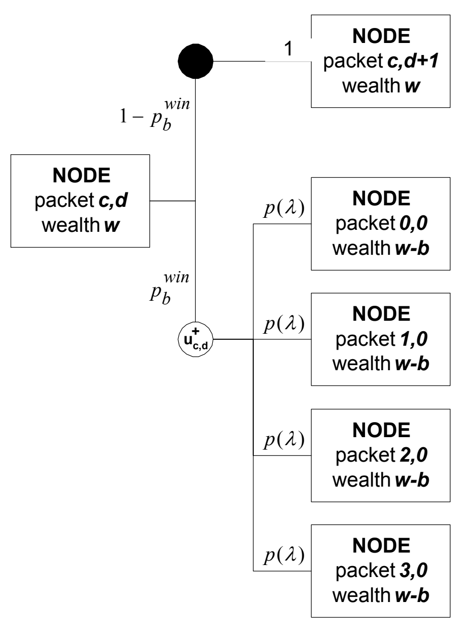
A node receives packets that it wishes to transmit; each such packet requires a *class* of service. The current packet the node has is denoted by *c*. For convenience, let *c=0* denote that the node is quiescent and has no packet to send. For example, one could posit that there are only 4 different classes of traffic: e-mail, HTTP, FTP, and real-time. In this case, *c* would range from 0 to 4. If *c=0* there is no packet to send, and the node submits a bid of zero.

A node maintains a probability distribution λ, that tells it the relative likelihood of getting new packets of type *c*. For now, let us assume that λ is exogenous, built up by the node based on historical traffic trends. Just like the consumer in our previous example, a node only gains utility from successful transmissions. The amount of happiness a node receives depends both on the *class* of the packet *c*, and the amount of delay the packet had to endure, *d*. Specifically, successfully sending higher classes of traffic gives the node more utility. At the same time, the delay a packet faces detracts from the nodes ultimate payoff.

Value Iteration

Let us first derive a recursion for the slightly cleaner case of a first-price auction. Consider a node that has a packet of class *c* pending. The transmission has already been delayed for a time *d*. Our node has wealth *w* remaining at its disposal, from which it must make its bid. The node's current *state* thus, is described completely by these three variables.

Our goal is to determine the optimal bid given *w c*, and; we represent this as $B_{w,c,d.}$ In placing a bid *b*, the node knows that if it wins it will receive a utility payoff $u_{c,d}$, and its wealth will go down by the amount bid.



60 The graph was obtained by directly coding the algorithm above in C++. β was set to 0.8

In addition, the node will then receive a new packet of class c (or none at all, $c=0$) according to the distribution λ. Should it lose, however, its wealth will be unchanged, but its packet will remain pending, suffering additional delay, which presumably will reduce eventual payoff.

This process is depicted graphically in the figure, and the recursive equation (5.9) immediately follows:

$$V_{w,c,d} = \max_{bid \le w} \left\{ \begin{array}{l} p_{bid}^{win} \cdot (u_{c,d} + \beta \sum_{\lambda} p_{\lambda} \cdot V_{w\text{-}bid,c=\lambda,0}) \\ + (1 - p_{bid}^{win}) \cdot \beta V_{w,c,d+1} \end{array} \right\}$$

$$B_{w,c,d} = \arg\max_{bid \le w} \left\{ \begin{array}{l} p_{bid}^{win} \cdot (u_{c,d} + \beta \sum_{\lambda} p_{\lambda} \cdot V_{w\text{-}bid,c=\lambda,0}) \\ + (1 - p_{bid}^{win}) \cdot \beta V_{w,c,d+1} \end{array} \right\}$$

β ≡ the discount factor reflecting the myopia of the node
p_{bid}^{win} ≡ the probability of winning the auction, given *bid*
p_{λ} ≡ the probability of receiving a new packet of class λ
$u_{c,d}$ ≡ the payoff for sending a packet of class c after delay d
$V_{w,c,d}$ ≡ the value of having wealth w, while a packet of class c is suffering delay d

Note that in (5.9) above, subscripts represent function *variables*. Thus $V_{w,c,d}$ is actually the value function $V(w,c,d)$, and p_{bid}^{win} is $p^{win}(bid)$, the probability of winning as a function of the amount bid[61].

We discuss how these functions are determined below, but for now let us focus on understanding the equation itself. It may be helpful to refer to the figure.

The second term is straightforward. If the node *loses* its bid, its packet suffers an additional period of delay, but its wealth remains unchanged. In other words, the node transitions to the state $V_{w,c,d+1}$. If the node *wins*, it receives payoff $u_{c,d}$, depending on the class of packet it has managed to send and the delay the packet had suffered. It also receives a new transmission c, transitioning to a new state $V_{w-bid,c,0}$. The summation simply weights this by the relative likelihood of transitioning to each such state[62].

The logic of the iteration in (5.9) should now be clear. As before, we iteratively update V until it converges to its true value[63]. Once V converges, B (w,c,d) contains the optimal bid for a node given its wealth, the kind of packet it has, and the delay that packet has already faced.

Vickrey Auction

Our discussion above was based on a first-price auction. The only change in the recursion for a Vickrey auction is that when the node wins, it expects its wealth to fall by the estimated second-highest price, rather than by the amount it bid. Thus the recursion takes the form in (5.10) below:

$$V_{w,c,d} = \max_{bid \le w} \left\{ \begin{array}{l} p_{bid}^{win} \cdot (u_{c,d} + \beta \sum_{\lambda} p_{\lambda} \cdot V_{w\text{-}\phi_{bid},c=\lambda,0}) \\ + (1 - p_{bid}^{win}) \cdot \beta V_{w,c,d+1} \end{array} \right\}$$

$$B_{w,c,d} = \arg\max_{bid \le w} \left\{ \begin{array}{l} p_{bid}^{win} \cdot (u_{c,d} + \beta \sum_{\lambda} p_{\lambda} \cdot V_{w\text{-}\phi_{bid},c=\lambda,0}) \\ + (1 - p_{bid}^{win}) \cdot \beta V_{w,c,d+1} \end{array} \right\}$$

β ≡ the discount factor reflecting the myopia of the node
p_{bid}^{win} ≡ the probability of winning the auction, given *bid*
p_{λ} ≡ the probability of receiving a new packet of class λ
$u_{c,d}$ ≡ the payoff for sending a packet of class c after delay d
ϕ_{bid} ≡ the expected Vickrey price, **given** that the highest was *bid*
$V_{w,c,d}$ ≡ the value of having wealth w, while a packet of class c is suffering delay d

Building the Distributions

There are some key points to note in (5.9) and its counterpart. In addition to the explicitly exogenous λ, the value function and the optimal bids derived depend on the probability of winning p_{bid}^{win}, and the payoff function $u_{c,d}$. The Vickrey auction also requires knowledge of ϕ_{bid}, the expected second-

[61] Given the dynamic-programming nature of (5.10), we restrict bids to be integers for ease of implementation.

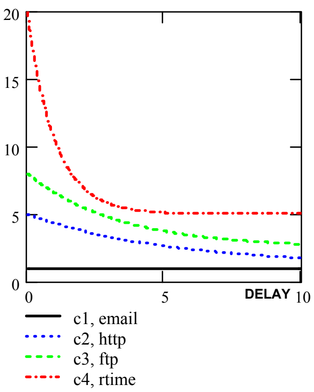
[62] For example, if the node believes, based on its distribution λ, that there is a 75% chance the next state will be quiescence ($c=0$), a 5% chance of directly getting a new real-time packet ($c=3$), and an equal likelihood for an HTTP or an e-mail ($c=1,2$) then the term is: $\frac{75}{100} V_{w-bid,0,0} + \frac{10}{100}\left(V_{w-bid,1,0} + V_{w-bid,2,0}\right) + \frac{5}{100} \cdot V_{w-bid,3,0}$
We will have to more to say on λ later.

[63] Intuitively the first term in (5.9) represents the value from winning the bid, while the second represents the value from losing it. The right-hand side as a whole represents the *expected value* of placing a bid *b*. At every iteration, the node chooses the bid that yields the maximum expected value. Once V has converged, this bid will be optimal, in that it yields maximum expected utility.

highest price. In this section we discuss how these values can be determined.

Payoff Function, $u_{c,d}$

Successfully sending a transmission gives the node *utility*. The utility function itself depends entirely on the user[64], capturing how much he values the transmission, and how much relative delay he is willing to tolerate for each class of transmission.

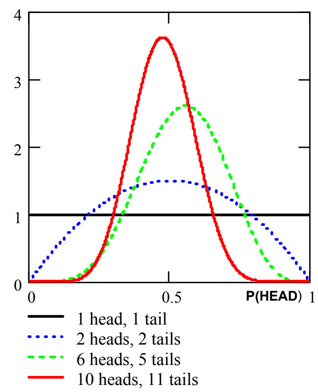


In general, high priority packets, those with high c, will have larger payoffs. Additionally, the payoff should decay with delay d. Thus, for example, real-time traffic should have very high payoffs for a successful packet, but payoff should decay quickly with increasing delay: sending a real-time packet swiftly gives the node a lot of happiness, but since the packet is very sensitive to delay, the payoff falls rapidly[65].

Thus the utility function $u_{c,d}$ is increasing in c and decreasing in d. The specific *shapes* of the tradeoffs depend entirely on the user. The diagram represents the payoffs set by a sample user for our standard example of e-mail, FTP, HTTP, and real-time traffic. Note, for example, that successfully sending a real-time packet with zero delay has a very high payoff, but it falls rapidly with delay. Conversely email has lower pay-off, but is quite insensitive to delay. Finally, note that like all utility functions, the numbers themselves mean very little: units of 'happiness' are rather arbitrary; it is the *relative* importance that matters.

Dirichlet Distribution

The remaining parameters p_{bid}^{win}, and ϕ_{bid} however are entirely *endogenous*, and need to be determined within the system. At this point, we make a very brief digression into some basic properties of the Dirichlet distribution. The Dirichlet is a standard method for building up a Bayesian probability distribution from discrete observations. Readers familiar with the Dirichlet can skip this section; it is not technical and aims only to build intuition for our discussion.



Consider tossing a coin and observing the results. We wish to build up a *distribution* for the probability of getting a head, using our observations. The graph above shows how the PDF for P(head) evolves with additional results. With only one head and one tail, the distribution is flat, though the expected value remains 0.5. With further observations, however, it becomes increasingly concentrated around the mean[66].

The nice thing about this is that given Dirichlet priors, the posterior remains Dirichlet. For example, suppose we have seen two heads and a tail, and thus have a Dir(2,1) prior. Now if we observe four more heads and tails, the posterior can be shown to be Dir(6,5)[67]; thus makes updating our distribution remarkably simple.

Note also that we can initialize our priors arbitrarily to reflect observations we pretend to have seen in the past, giving us a simple way to introduce prior

---

[64] This is entirely analogous to our example of a consumer who gained utility solely from consumption. There, too, the form of the utility function depended entirely on the individual preferences of the consumer.

[65] Note that this is the *total* payoff for a packet, not the *spot* payoff. To make this clearer, an alternative way of expressing our value iteration would be to specify that each time the node loses a bid it suffers a *negative* penalty. However, the *spot* value of sending a delayed packet *increases* with delay. For example a packet sent with 0 delay could have a payoff of 10. With each cycle of delay, it accumulates a negative payoff of -5. After 3 cycles of unsuccessfully trying to send the packet the node's value is at –15, but the spot payoff for the packet has risen to, say, 17, since sending the packet is becoming more urgent. The *total* payoff is only 2. The two methods are equivalent: we choose total payoff simply for expositional simplicity.

[66] Our description of the Dirichlet is based on Parr, R. "Learning Probability Distributions," 2001. Our use of it is inspired by Boutilier & Goldszmidt, 1999.

[67] This is not immediately obvious. Consider an example where X=P(Heads), P(X) is Dir(2,2), and new observations, O are H,T,T. Then $P(X \mid O) = P(O \mid X)\, P(X) / P(O)$, but it is quite complicated to prove that *P(X|O)* is also Dirichlet.

knowledge[68]. Finally, note that we can express the *same* mean with different strengths, simply by changing the number of observations – as the figure shows, Dir(2,2) is quite different from Dir(10,10). For large amounts of new data, of course, our phantom examples will be overwhelmed and not matter. Thus the Dirichlet is ideal for our purpose of learning and building up Bayesian probability distributions based on discrete observations.

We summarize the essential properties of the distribution below.

$$\mathbf{Dir}(\underbrace{x_1 \ldots x_n}_{n\text{ events}} \mid \underbrace{\alpha_1 \ldots \alpha_n}_{\text{observations for each}}) = \frac{\Gamma(\alpha_1 + \cdots + \alpha_n)}{\Gamma(\alpha_1)\cdots\Gamma(\alpha_n)} x_1^{\alpha_1 - 1} \ldots x_n^{\alpha_n - 1}$$

$$\left(\begin{array}{l} \Gamma(\alpha) = \int_0^\infty u^{\alpha-1} e^{-u} du \\ \overline{\boldsymbol{x}}_i = \dfrac{\alpha_i}{\sum_{j=0}^n \alpha_j}, \boldsymbol{\sigma}^2_{x_i} = \dfrac{\alpha_i(\sum_{j=0}^n \alpha_j - \alpha_i)}{\left[\sum_{j=0}^n \alpha_j\right]^3 + \left[\sum_{j=0}^n \alpha_j\right]^2} \end{array}\right)$$

Probability of a bid winning, p_{bid}^{win}

In determining the optimal bid in (5.10), the node needs to have an estimate of how likely it is that the bid will be successful. The node, of course, knows that the probability of success increases with the amount bid, but we need a way to quantify this understanding. How can a node build up this information?

Obviously, the more information a node receives, the better estimate it can build for p_{bid}^{win}. At one extreme we could assume that the entire list of bids is announced, on the other, that the node receives no information other than whether it won or lost[69]. For this discussion, we use the natural assumption that at the end of the auction, only the winner and the winning bid are revealed.

Our node thus maintains a Dirichlet distribution Ω for the winning bids, with parameters $\alpha^1, \alpha^2 \ldots \alpha^{\max}$. Each time the node observes a winning bid of *k* tokens, it increments the Dirichlet parameter $\alpha^k \to \alpha^k + 1$. Thus at any moment the parameter α^k simply contains the number of times the node has observed a winning bid of *k* tokens.

Now consider p_b^{win}. Our node knows that if it does not bid, the winning bid *x* will have the distribution Ω. Consequently, the chance that a bid of *b* tokens will win is simply the expected probability that *x* is **less** than *b*. From the standard properties of the Dirichlet:

$$p_b^{win} = \Omega(x < b) = \frac{\sum_{i=0}^{b-1} \alpha_i}{\sum_i \alpha_i} \qquad (5.11)$$

(5.11) is intuitively appealing. It is merely the fraction of times the node has observed bids smaller than *b* win[70].

Vickrey Price, ϕ_{bid}

Recall from (5.10) that Vickrey auctions require an additional parameter. After a successful bid, the node's wealth falls by ϕ_{bid}, the next highest bid. We can easily estimate this from our vector of observations Ω. If our node wins with a bid of *b*, then had it not bid the winning bid would be *less* than *b*. What would this winning bid have been?

$$\phi_{bid} = \frac{\sum_{i=0}^{b-1} i \cdot \alpha_i}{\sum_{i=0}^{b-1} \alpha_i} \qquad (5.12)$$

---

[68] This can be useful to give nodes some default information, based say, on simulation, that they can use initially. As they observe more and more data, the prior information will become increasingly discounted.

[69] Additional information, such as each player's current wealth, can be utilized to determine better bids. This however makes the game more and more *cooperative*. We focus on the more difficult *non-cooperative* scenario here.

[70] For convenience, we assume that ties are lost. Technically, the equation should read:

$$p_b^{win} = \frac{\sum_{i=0}^{b-1} \alpha_i}{\sum_i \alpha_i} + \frac{\alpha_b}{\sum_i \alpha_i} \cdot \frac{1}{N_b}$$

If the node does not bid, it knows that the winning bid will be *x*. Now consider placing a bid *b*. If *x*<*b*, the bid will be successful. If *x*=*b*, then there is a tie and all tying nodes have an equal chance of winning. The second term in the equation represents the (small) expected probability that *x*=*b*. This is further divided by the number of tying nodes *N,* which must be estimated. While this could easily be done, we opt for clarity and make the reasonable assumption that it is negligible.

This is just the expected winning bid, *given* that it must be less than b[71].

In ending, we note that nodes only need to maintain *one* underlying table: Ω, the vector of observations. The other distributions directly stem from it.

The System

We can now paint a picture of how bidding works. Nodes continually monitor winning bids and update their probability distributions. They also continuously – or periodically, depending on computational availability – recalculate[72] their optimal bids for a given kind of packet.

This allows the system to adapt readily to changing conditions. For example, should network activity fall, lower bids will have a higher chance of winning. The node will observe this and transparently incorporate this information when it recalculates its optimal bids. Similarly, nodes will react appropriately when additional nodes enter or exit the system, since the number of players affects the probability of winning, which in turn determines the optimal bid[73]. Nodes can also transparently account for inflation, or adapt to different budget constraints and funding strategies, which we discuss in the next chapter.

In this manner nodes dynamically determine how much to bid for their packets. This approach automatically adjusts for inflation, variable valuation, uncertainty, and so on. Given the distributions, the bids calculated by the iteration are strictly optimal.

We note in ending, that a token economy system provides nodes with a budget and rationality to use that budget, allowing them to suffer delay when it most suits them, increasing welfare[74]. In the next chapter we simulate a simple token economy in which nodes determine their bids using the methods we have described, and we find that our scheme is viable in practice[75], and improves efficiency.

Observations and Issues

There are some important details that need to be considered in order to gain a more complete understanding of our bidding system. We discuss these next.

Variability of Bids

The bidding system dynamically adapts to changing network conditions by periodically recalculating its optimal bids. This of course, means that the optimal bid for a node varies with network conditions.

Consider an exogenous drop in network activity, which results in the node observing a number of low winning bids. This impacts its vector of observations Ω, and consequently its estimate for p_b^{win}. When the node reruns the value iteration, this new estimate will result in *different* optimal bids. If activity falls and low bids stand a greater chance, a rational agent in a *first-price* auction should clearly reduce his bids correspondingly, since he does not wish to end up paying more than necessary.

It may be less clear why the optimal bid for a Vickrey auction would also shift. We have seen that it is optimal for a node to bid its true, static valuation in such an auction. The amount the node pays has nothing to do with its bid, but rather depends on the second highest price. Since demand has fallen, and lower bids are winning, the Vickrey price ϕ_{bid} it will have to pay is already *automatically* lower. There is in fact, a *deeper* reason for the shift in Vickrey bids.

[71] The expected probability of getting a bid lower than b is just (5.11). The expected bid *given* this, is just the usual expected value, divided by this probability.

$$\frac{\sum_{i=0}^{b-1} i\cdot\alpha_i}{\sum_i \alpha_i}\cdot\left[\frac{\sum_{i=0}^{b-1}\alpha_i}{\sum_i \alpha_i}\right]^{-1} = \frac{\sum_{i=0}^{b-1} i\cdot\alpha_i}{\sum_{i=0}^{b-1}\alpha_i}$$

[72] Alternatively, updates could explicitly *triggered*. Recall that a node continuously updates its vector of observed winning bids in the system Ω. Should Ω begin to drift, it will tell the node that it should recalculate the optimal bids, because the probabilities they were based on have now changed.

[73] Bids thus are variable with time; we discuss the implications of this shortly.

[74] This is in marked contrast to 'penalty-schemes' we discussed in Chapter I, where the device would have to suffer delay after every transmission, regardless of how inconvenient it could be.

[75] Boutilier & Goldszmidt (1999) suggest using dynamic programming to determine bids for combinatorial and sequential auctions. While their purpose is different from ours, they also report excellent practical results. Our earlier discussion draws on their work. We also use their model in detail in Chapter VII.

If lower bids stand a greater chance of success, the *value* of a single token itself has risen: it can buy *more*. Thus a node with a constant true valuation would still find that its Vickrey bid *in terms of tokens* would change[76].

The variability, thus, has no real significance. As the distributions evolve, due to events such as drops in network activity, or the arrivals and departure of competing nodes, the iteration dynamically adjusts the nodes behavior to remain optimal and welfare maximizing.

A Note On λ

In order for a node to perform value iteration, it needs to have an idea of the amount and type of traffic it may have to serve in the future, so that it can budget its wealth appropriately. We addressed this need using λ, our sole exogenous distribution, which provides the node with an estimated likelihood of getting a packet of class c.

λ can be built up directly, using historical observations in a manner identical to p_b^{win}, or it can be based[77] on an arbitrarily complex network traffic model. In practice, it is likely that the arrival rates and parameters of such a model will depend on dynamic factors such as the processes running on the node, the time of the day etc.

No matter what method we choose to estimate λ, it is important to *limit* the horizon of the node. We discuss this explicitly in the next chapter, but for the time being, let us assume that the node is reset to its initial state every T periods. In this case, λ represents the relative likelihood of getting new packets of type c during the next T periods. This is helpful, both because it is easier to make accurate estimates for shorter time periods, and because our system's behavior can now be tailored for conditions specific to the current horizon.

With resets λ can also be made more endogenous, by providing feedback to the user and network applications running on the node. For example, if demand during the current horizon is high, the node can inform the user and running processes that transmissions are currently expensive, and likely to face delay. The user could then reduce activity; this would change the node's estimate for λ for the next horizon, allowing it to distribute higher bids over fewer packets so that they suffer less delay.

Time Weighted Distributions

We may also wish to bias the probability distributions, placing more weight on current observations than on old ones. This can clearly be useful, especially if the level of congestion in the system fluctuates often, even within a single horizon. Fortunately, this is trivial in practice. Instead of directly incrementing the Dirichlet parameter α^k with new observations $\alpha^k \to \alpha^k + \alpha^k_{new}$, we discount old observations: $\alpha^k \to \lceil \delta\alpha^k + \alpha^k_{new} \rceil$, where the discount factor $\delta < 1$

Implementation Independence

In ending, we emphasize that the token system does not depend on the specifics of the simple implementation we have discussed. With time, more sophisticated and intelligent bidding systems will arise as users try to squeeze out the maximum value from a token. Such systems could operate by strategically deferring transmissions, using real-time network traffic models to predict congestion. Alternatively, nodes could identify a set of rivals that they are often in contention with and try to explicitly model their wealth. Similarly, systems could exploit sophisticated AI techniques such as reinforced learning. As we mentioned earlier, revealing additional information (such as individual bids) at the end of an auction can lead to even better estimates for p_b^{win}, and hence better bidding decisions. This could be appropriate in team scenarios or with a common owner, where the nodes are not in direct competition. The important thing is that the token economy creates the correct *incentives* for users to use their endowment to transmit *efficiently*, and this is exactly what we desire.

[76] This is akin to the change in my dollar valuation for an apple, if the dollar itself were to appreciate or depreciate. Note also that the deviation in Vickrey bids is likely to be less than in first-price bids. There are two effects operating in a first-price auction, the fact that it is *rational* to bid less if others are bidding less, and the fact that the value of a token itself has changed. For a Vickrey auction, only the second applies.

[77] A standard approach could be to use a Markov modulated Poisson process. We use such a process in Chapter VI, where the likelihood of getting a new packet of type c, or none at all $c=0$, is based on previous activity. For the zealous reader, we discuss a simple Poisson queue-based model for λ in the appendix.

Summary

We have considered a simple technique that allows nodes to determine their optimal bids in the currency of tokens. Our method allows nodes to automatically respond to exogenous shocks such as changes in activity, nodes entering or leaving the system etc. The bidding system is distributed and non-cooperative, with each node performs its own calculation and optimizations. In combination with the funding strategies from the next chapter, our bidding system allows for the creation of a complete token economy.

VI Token Economy: Funding

At the beginning of Chapter IV, we posed two questions. The first has been answered: we have considered methods for bidding in tokens. We now focus on the second and consider how nodes get tokens themselves.

In a token economy, much like the real world, *funding* sets priority. Richer nodes have greater 'muscle' and can afford to bid more on their transmissions[78]. Determining an appropriate funding strategy thus, is key. Fortunately, our bidding system can automatically account for different funding plans: all we need to do is adjust the value iteration to reflect them.

A Note on Funding and Priority

Why is funding important? We have seen in earlier chapters that the auction mechanism results in the classic 'supply equals demand' equilibrium, and that the resulting allocation is strictly optimal. The Second Theorem of Welfare Economics states that *every* efficient allocation can be achieved using this technique: if there are *n* efficient ways to distribute a set of goods, one can use the auction mechanism to reach any of these efficient outcomes, *merely by changing the endowments of the bidders*.

Prices have two roles in a market mechanism: first, they are indicators of scarcity, and second, they determine how much different agents can buy. The theorem essentially states that these two roles can be *separated*. We can set wealth endowments to control what each agent can buy to satisfy administrative or social criteria, while still reaping the benefits of pricing as a signal of scarcity. As we have seen, the way to achieve an efficient allocation is for each agent to face the true social costs of his actions, and to make decisions that reflect those costs. Thus, the decision of whether to consume more or less goods will depend on the *price* of those goods. The price provides a single measure of how much everyone else values them, and how much of a social cost the agent's consumption will impose. The pricing mechanism has nothing to say about *allocation* issues, such as: What can node *A* afford? Should it consume more than *B,* or less? Such issues can be handled simply by changing *A* and *B*'s relative funding.

In fact, setting endowments for different nodes is essentially equivalent to setting priorities. Intuitively, wealthier nodes get more priority in an economic system than others; *Spawn*, for example, reports this in practice:

> The auctions employed by *Spawn* are second price auctions… ***Spawn's* market mechanisms can also give some tasks high priority, where higher funding corresponds to higher priority**. To demonstrate this, we introduced a high-priority task into a system in which low-priority tasks were running… the high-priority task captured approximately 60% of the total system resources. This is in close agreement with its 'fair' value of 70%, since the three competing tasks were funded in a 7:2:1 ratio…
> [Waldspurger etal. *Spawn: A Distributed* Computation *Economy,* IEEE Trans. Soft. Eng. 18:2 Feb. 1992, emphasis added]

This provides a simple solution for a common administrative issue: setting priorities. For example, a professor's wireless device may deserve more priority than a student's – and this can be accomplished easily by giving the preferred device more funds to bid with.

Open Funding

Perhaps the most straightforward method for funding nodes is to grant them a fixed amount of income, μ every period. Such 'open' funding strategies are likely to be combined with a wealth-cap, to prevent nodes that are not spending enough from gaining wealth *ad infinitum*.

With funding every period, the value iteration in (5.9) becomes (6.1) below.

$$V_{w,c,d} = \max_{bid \le w} \left\{ \begin{array}{l} p_{bid}^{win} \cdot (u_{c,d} + \beta \sum_{\lambda} p_{\lambda} \cdot V_{w+\mu-bid,c=\lambda,0}) \\ +(1-p_{bid}^{win}) \cdot \beta V_{w+\mu,c,d+1} \end{array} \right\}$$

[78] In addition, nodes that expect to have fewer transmissions to send can afford to bid more per transmission, and gain priority. This, too, is essentially a funding effect: these nodes have been funded more per transmission than others.

In (6.1) above, our node knows that regardless of whether it wins or loses it will receive an additional endowment of μ. Since the node *knows* this, it will take this into account in determining the value of tokens and its optimal bid. Simulations performed with this funding method show that nodes do adjust their spending appropriately: their wealth does not rise dramatically with time, hovering instead comfortably below the wealth-cap.

Adding a Time Horizon: Resets

As we discussed in the previous chapter, it is often desirable to limit the node's horizon. For example, it is likely that λ can be modeled with greater accuracy over shorter periods. It is easier for a node to expect, for example, that there will be 20 email requests in the next hour, instead of trying to predict the distribution over a much longer period. Resets are also important for correcting errors that may creep into the system over a period of time, and for allowing nodes to respond more aggressively to local events.

Let us assume then, that every *T* periods, all nodes are reset to their starting wealth. Obviously, this would result in a creeping inflation. To see this, consider a node in the last period *T-1*. Our node will bid everything it has, without trying to save anything for the future, simply because there is no future. Fortunately, our bidding strategy can automatically take this into account. With an income of μ and a reset every *T* periods the value iteration becomes:

$$V_{w,t,c} = \max_{bid \le w} \left\{ \begin{array}{l} p_{bid}^{win} \cdot (u_{c,t-t_0} + \beta \sum_{\lambda} p_{\lambda} \cdot V_{w+\mu\text{-}bid,t+1,\lambda}) \\ +(1 - p_{bid}^{win}) \cdot \beta V_{w+\mu,t+1,c} \end{array} \right\}$$

$$V_{w,T,c} = V_{w_0,0,c}$$

So long as the node is aware of when the reset is going to occur, it will adjust its bidding behavior accordingly. The frequency of the reset is up to the system administrator, but semi-frequent resets are likely to be useful.

An alternative approach is to periodically issue unexpected stochastic resets into the system. This will tend to reduce a nodes value for saving, since every period there is now a non-zero chance that its 'savings' will evaporate, but it will avoid systematic inflation. As we have seen, however, inflation can automatically be handled by the value iteration. The primary purpose of such a stochastic scheme would be the simplicity of implementation.

Open Funding In Practice

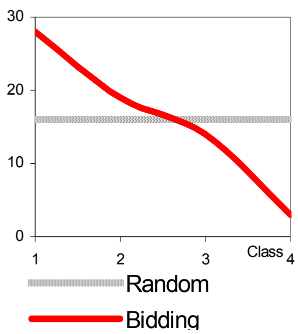
To verify that the token economy can effectively utilize open funding in practice, we performed Monte Carlo simulations for a population of twenty identical nodes using such a scheme. Starting wealth was arbitrarily set to 20 tokens and an income of 2 tokens was awarded every period. There was a wealth cap of 1000 tokens. As usual, we assume that nodes received packets according to the exogenous distribution λ, which was modeled as a simple Markov modulated Poisson process for four classes of traffic, e-mail, HTTP, FTP, and real-time[79].

At every period an auction was held to determine who would transmit, and the winner was charged the second-highest price. The winner would then receive a new packet - or none at all - according to λ, while existing packets of other nodes would remain pending. The utility payoffs used for the value iteration were identical to $u_{c,d}$ described in the pervious chapter.

Nodes observed winning bids, and continuously updated Ω and p_b^{win}, as detailed in the previous

[79] Specifically, λ was modeled as a 5x5 matrix of functions representing transition probabilities from one class to another. For example, the first row of the matrix has five cells representing the probabilities that a node which currently has *no* packet will either: (0,0) remain quiescent; (0,1) get a packet of class e-mail; (0,2) get a packet of class HTTP; (0,3) get a packet of class FTP; and (0,4) get a real-time packet.
In general, we assumed that:
(a) Transmissions are infrequent, thus the probability of remaining silent, (0,0) is high.
(b) The likelihood of getting new packets increases with *d*, the amount of time the current packet has been pending.
(c) New high-priority real-time or FTP packets are much less common than low-priority e-mail or HTTP packets, e.g. (0,4) < (0,1)
(d) Current FTP and real-time packets are more likely to be followed by additional FTP and real-time packets, while other types are likely to be followed by a return to quiescence
λ, of course, can be made arbitrarily complex. This simple approach is sufficient to verify that the system operates as intended. More complex network traffic models can be simulated, and we intend to study these in the near future.

chapter. After every thousand periods, they would re-run the value iteration in (6.1) and update their optimal bids based on the current p_b^{win}. Since there were no exogenous shocks, after several such update-cycles the system converged to a steady state. The figure depicts the *average delay* suffered by packets with, and without the bidding system in steady state.



The horizontal line indicates the average delay that each class suffered using random-access schemes. Obviously, the delay is the same, regardless of class or priority, since the winner is random. The curve represents the average delay for packets using the bidding system. Note that high-priority, delay-sensitive packets go through far more quickly. Of course, this gain comes at the expense of the low-value packets, such as e-mail, which now have to wait longer.

Social welfare is clearly higher using the bidding system, since high-priority packets have to wait *less.* These simple results are encouraging, and indicate that our faith in an economic solution is well founded.

Closed Funding

We have seen that open funding schemes are simple, efficient, and workable. An essential characteristic of these systems is the continuous creation and destruction of wealth. The income of nodes adds to the total tokens in the system, while successful bids *reduce* the amount of tokens in the system. Tokens, thus, are continually created and destroyed. Our valuation function ensures that these two forces are in balance for active nodes. Indeed, simulation results show that the nodes' average wealth tends to hover around an average, instead of increasing to the wealth cap or falling to zero[80].

Though we have an efficient, workable system, the continuous creation and destruction of money can be inelegant. In the rest of this chapter we discuss alternatives in which the total amount of money is *fixed*, shared amongst the nodes. We stress that the following discussion is exploratory.

One way to do this conceptually is to consider each node as owning part of the airwaves. His individual share, s^i is not enough to support a transmission; thus when, he transmits he must use other's shares as well. These nodes must be compensated for his usage of their rights. In this way, the money paid by a transmitting node is redistributed to others; keeping net money in the system constant.

Period	0	1	2	3	N
Net Resources	*1*	*1*	*1*	*1*	…
A's share	1/3	1/3	1/3	1/3	…
B's share	1/3	1/3	1/3	1/3	…
C's share	1/3	1/3	1/3	1/3	…

Consider the simple example above. Three nodes share resources, in a network where only one node can access the airwaves at a time. We have arbitrarily granted all three an equal share, for simplicity. Of course, this is not necessary. Some devices may have more priority than others, and they could be granted more property rights.

How does the allocation work? Each node starts with a fixed amount of wealth w_0, and has a fixed, perpetual share s^i of the resources. As usual, nodes wishing to transmit gain access to the airwaves through a Vickrey auction. Assume that the clearing price for the auction at time t is p_t.

Each node's wealth evolves as:

$$w_{t+1}^i = w_t^i + p_t(s^i - X_t^i) \tag{6.2}$$

Where X_t^i are the network resources consumed by node i in this period. Assume that node A wins, and for example, uses up all network resources. Thus $X_1^A = 1$ and the second period wealth evolves as:

$$\begin{aligned} w_1^A &= w_0^A - \frac{2}{3}p \\ w_1^B &= w_0^B + \frac{1}{3}p \\ w_1^C &= w_0^C + \frac{1}{3}p \end{aligned} \tag{6.3}$$

[80] Nevertheless wealth caps and taxes are useful to prevent possible overflow in inactive nodes that receive money, but do not use it.

And the net wealth remains constant. In general, summing over (6.2) for all nodes:

$$\sum_i w_{t+1}^i = \sum_i w_t^i + p_t \sum_i (s^i - X_t^i) \qquad (6.4)$$

The last term in this equation is always **zero.** If consumption is less than the total supply of the resource, the auction-clearing price will be *zero*. Alternatively, if there is more demand than supply than the price will rise until consumption *equals* the supply of resources. In this case, $\sum_i (s^i - X_t^i) = \sum_i s^i - \sum_i X_t^i = 0$.

Thus (6.4) can be rewritten as:

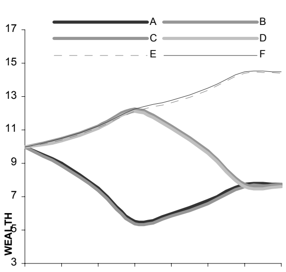
$$\sum_i w_{t+1}^i = \sum_i w_t^i \qquad (6.5)$$

and the sum of wealth in the system is always constant.

The table below shows the consumptions X_t^i, and evolution of the wealth w_t^i, for 6 nodes over multiple periods. Each node has an equal share of total network resources: thus $s_i = K/6$, and the network has the capacity for $K=2$ simultaneous transmissions.

	x_0^i	x_1^i, w_1^i	x_2^i, w_2^i	x_3^i, w_3^i	x_4^i, w_4^i	x_5^i, w_5^i	x_6^i, w_6^i	x_7^i, w_7^i
A	10	*1,9*	*1,7.5*	*1,5.5*	*0,6*	*0,6.75*	*0,7.75*	*0,7.75*
B	10	*1,9*	*1,7.5*	*1,5.5*	*0,6*	*0.6.75*	*0,7.75*	*0,7.75*
C	10	*0,10.5*	*0,11.25*	*0,12.25*	*1,11.25*	*1,9.75*	*1,7.75*	*0,7.75*
D	10	*0,10.5*	*0,11.25*	*0,12.25*	*1,11.25*	*1,9.75*	*1,7.75*	*0,7.75*
E	10	*0,10.5*	*0,11.25*	*0,12.25*	*0,12.75*	*0,13.5*	*0,14.5*	*1,14.5*
F	10	0,10.5	0,11.25	0,12.25	0,12.75	0,13.5	0,14.5	0,14.5
Σ	60	*2,60*	*2,60*	*2,60*	*2,60*	*2,60*	*2,60*	*1,60*
p_t		1.5	2.25	3	1.5	2.25	3	0

Note especially period 7, where only *E* placed a bid to transmit. Since total demand was less than network resources, no one was inconvenienced and the Vickrey auction yielded a nil price. Note also that nodes *A* and *B* made three transmissions, as did *C* and *D*. Their ending wealth is the *same* – generally these will be distinct as the Vickrey auction may result in slightly different prices at each round. The principle, however, is clear: by remaining quiescent, a node quickly



regains its wealth, and the total amount of wealth in the system is constant.

Conceptually the system is appealing since each node reimburses others for the use of their 'shares' of the spectrum. *K* is also generalizable to an arbitrary amount of network 'resources', allotted by auction or other market mechanism[81].

How does this closed funding strategy affect the value iteration for bidding? Let us denote the fraction of spectrum the node owns as η. In the case of equal shares, this is merely 1/N, where N is the number of nodes in the system. Consider the slightly cleaner case of a first-price auction[82]. The node knows that if it wins a bid *b*, its wealth will fall by *b* but it will also *receive b*/N. Thus, should the node win, its wealth falls by *b*(1-N)/N. Should the node lose however, it knows that it will receive 1/N$^{\text{th}}$ of the winning-bid.

Thus the value iteration is just:

$$V_{w,c,d} = \max_{bid \le w} \left\{ \begin{array}{l} p_{bid}^{win} \cdot (u_{c,d} + \beta \sum_{\lambda} p_{\lambda} \cdot V_{w + \frac{bid}{N} - bid, \lambda, 0}) \\ + (1 - p_{bid}^{win}) \cdot \beta V_{w + \frac{1}{N}\Phi, c, d+1} \end{array} \right\}$$

Here $\Phi = E(\Omega > bid)$ the expected winning bid, given that it is larger than *bid*[83]. Note that this iteration requires that each node know *N*, the total number of nodes in the system, which leads to decentralization issues that we discuss later.

Wealth As 'Property Rights'

In the simple closed economy system described above, each node owns a fixed share of the spectrum, and the proceeds from winning bids are distributed accordingly. Let us explore an alternative closed form that makes the concept of wealth a little more meaningful[84]. Let us begin by recalling that the entire point of value iteration was to allow nodes to gain an understanding of the purely nominal wealth they possessed.

[81] See, for example, the Vickrey auction for bandwidth in the Appendix.
[82] The logic for the Vickrey auction is identical.
[83] This is analogous to the Vickrey price ϕ_{bid} (the expected winning bid, given that it is **less** than *bid*) and can be calculated from Ω in an entirely analogous manner.
[84] This idea is partially based on personal discussions with Prof. Paul Milgrom, 2002.

Consider a closed form system in which the node's wealth is the *share* of the spectrum, s^i it owns. This of course means that $w^i = s^i$ is no longer constant, fixed in perpetuity, but changes with time. With this change in mind, the new allocation rule becomes:

$$w_{t+1}^i = w_t^i + p_t(w_t^i - X_t^i) \qquad (6.6)$$

This means that the node's share of the winning bid is now determined by the current fraction of the spectrum it owns, which is the same as its wealth. The economy is still closed; the total wealth is constant and (6.5) still holds. Note further that the total wealth in the system must add up to unity[85]. It is instructive to see the evolution of the simple three-node example we considered before. *A* wins the auction, and is required to pay p_1. Subsequently, it wins again, and is required to pay p_2. The wealth/shares thus evolve as:

Period	0	1	2	*N*
Net Shares	*1*	*1*	*1*	…
A's share/wealth	$\frac{1}{3}$	$\frac{1}{3}-\frac{2p_1}{3}$	$\frac{1}{3}-\frac{2(p_1+p_1p_2+p}{3}$	…
B's share/wealth	$\frac{1}{3}$	$\frac{1}{3}+\frac{p_1}{3}$	$\frac{1}{3}+\frac{p_1+p_1p_2+p_2}{3}$	…
C's share/wealth	$\frac{1}{3}$	$\frac{1}{3}+\frac{p_1}{3}$	$\frac{1}{3}+\frac{p_1+p_1p_2+p_2}{3}$	…

Now when *A* examines its wealth it has a real meaning: it is the fraction of the channel that *A* owns. *A* knows that should it bid this fraction, its total wealth will remain *unchanged*. Bidding *more* than this will cause its share to drop. A node understands that if it owns a fourth of the channel, it will receive income from it that will allow it to transmit a ¼ of the time. Thus in considering placing high bids that reduce its share to, say, a fifth of the channel, the node must decide if this transmission is worth delaying subsequent packets by an additional period. This is something that *real* value can be assigned to, independent of the nominal token currency. Thus, this scheme opens up the possibility of giving nodes something concrete to determine their bids with, even without running a more formal value iteration.

[85] Of course, we could scale this to a conveniently large constant.

Observations and Issues

We end our exploration of closed economies with some general observations on issues that plague them. We also make a few concluding observations on token economies in general.

Closed Economies: Decentralization and Unity

It is important to ensure that funding policies and their corresponding iterations can be implemented in a decentralized manner. It is also desirable for accounts to be maintained by nodes themselves, since centralized accounting would create considerable burden on the central auctioneer, and generate substantial communication overhead. We discuss distributed implementations in the last chapter, but for now let us consider a challenge for decentralization unique to closed economies: the problem of maintaining *unity*.

While open economy funding schemes were entirely distributed, the closed economy presents a special problem: ensuring that the total wealth in the system remains constant. Recall the first closed economy, in which each node owned a fixed share of the proceeds, s^i. The problem, simply, is ensuring that s^i adds to unity. In a system where nodes can be inactive, join or leave the system, this task can be difficult to do in a distributed fashion without transitory *error*. The problem, of course, is that *any* error in estimating s^i will result in the economy becoming 'un-closed'. For example, nodes may not realize that a peer has entered the system and may continue to award themselves more than the correct fraction of the winning bid, increasing the total wealth in the system[86]. Thus there is a chance for inflation – or deflation – to slowly creep into the system, undermining our 'closed' economy.

The problem is worse in the more sophisticated closed form we discussed next. In that system, the share of spectrum owned s^i, was the same as the node's wealth, and thus constantly shifting. Ensuring that the ever-changing individual wealths in the system always add up to unity leaves an even larger margin for error. And error will lead inevitably to the economy expanding, or contracting.

[86] As we have seen, value iteration automatically accounts for inflation, but this defeats the point of having a closed economy.

We are not concerned with devising a decentralized update protocol: there are a number of excellent existing ones that could be adapted. Our concern is how a closed economy should deal with the inevitable probability of error. One approach is to fall back on our strategy of periodic resets – which we have already argued are important for other reasons. Thus while economies may expand or contract, they will periodically be reset to a starting point. Resets, however, are infrequent, and it is desirable to have some compensatory mechanism functioning in between, that pulls the economy back towards unity. One solution is to impose a *tax/subsidy* that helps compensate for the inflation. Like resets, the tax is multi-purpose. Not only can it help mitigate the problem of 'leaky' closed economies, it can also address *distributional* issues such as hoarding. We discuss taxes next.

Hoarding

Token economies, both open and closed, are susceptible to nodes that hoard tokens. A node that has been quiescent for a long period of time can amass a large wealth. It could then use this wealth to outbid everyone else and monopolize the channel for quite a while. Of course in doing so, its wealth will fall while others will get increasingly richer until they will be able to outbid it again. Economically this is completely fair, and even very efficient. Practically, however, this can be undesirable.

If there are periodic resets[87], the problem will naturally be mitigated. Even with resets, however, we may wish to limit the amount of wealth that a node can amass. To do this, we can bring any of the vast array of economic instruments at our disposal to bear. For open economies, perhaps the best solution is the simple wealth-cap: no node can amass more than an arbitrarily set amount. As we have seen, this approach held us in good stead in simulation as well.

While wealth-caps would work just as well in closed economies, an alternative approach is to use a *tax*. Consider charging each node a *wealth* tax. There are known to be efficient, *non-distortionary* taxes: unlike income tax, they do not detract from total welfare[88].

The new allocation rule is:

$$\begin{aligned} w_{t+1}^i &= p_t(w_t^i - X_t^i) + (1-\tau)w_t^i + \tau \cdot w_0 \\ &= p_t(w_t^i - X_t^i) + w_t^i - \tau(w_t^i - w_0) \end{aligned} \qquad (6.7)$$

This is identical to (6.6), except that every period the node is taxed on the wealth it has gained over its initial endowment. Nodes who have less than they started with *receive* wealth, since the last term becomes positive. Thus the 'tax' continually pushes nodes back towards their starting points. Note that (6.7) achieves redistribution: the total amount taxed is the total amount received, as $\sum_i (w_t^i - w_0) = 0$.

Note further that this approach is well suited for distributed implementation. Provided that each node is aware of the tax-rate, it can tax or subsidize itself every period. Finally, note that if left unused, the system lapses back to its starting state: essentially acting as an automatic reset, which is very convenient.

Finally note that (6.7) helps mitigate the unity problem. Since the tax/subsidy constantly pushes nodes back towards their initial endowments – which add up to unity – it can help counter inflationary or deflationary trends in the system. As nodes enter and leave the system, temporary errors in estimating initial endowments will be made. During this time the economy made inflate or contract. But once estimates have stabilized the tax will start pushing the economy back towards its closed form.

Closed Economies: Redistribution and Incentives

Note that both closed systems we have described redistribute the winning bid amongst *all* the nodes in the system, each of which have 'own' a share of the spectrum. Since the price the winning node(s) pay is determined by a Vickrey auction, it will correctly reflect the inconvenience caused to others, and this will result in a socially optimal equilibrium.

It may be tempting to consider distributing winning bids amongst nodes based on their individual *bids*; after all, one might reason, their bids indicate their desire to transmit. Shouldn't we compensate each node based on how much it is inconvenienced? Thus, for example, we may wish to award each losing node an amount proportional to: $b_i / \sum_i b_i \cdot [b_{win}]$

Essentially, a share of the proceeds weighted by the amount bid.

[87] We discussed the case for having periodic resets in the system in the section on open funding.

[88] Refer, for example, to Varian. H. *Microeconomic Analysis*, Norton Ed. 3rd edition

Such distributions however, affect the truth-telling incentives of the Vickrey auction. Intuitively, a bid in a Vickrey auction is merely a signal to enter into the market, the good a node receives – and the price it pays for it – depends on the bids of *others*. The allocation above however, creates strategic incentives for bidding, since the nodes pay-off is now directly linked to its bid. For example, no node would want to bid zero in such a scheme. It could always do better by placing a low bid and thus sharing in the income. Of course, the value iteration scheme will still mathematically converge to the new, strategically optimal bids – though it may take longer – but clearly, it will no longer be optimal to bid one's true valuation if tokens are worth real money, as in the case of secondary markets. Since there is no gain in welfare from introducing such forms of redistribution, and the issues they may cause are complex, it is best to avoid them altogether.

Credit Market & Interest Rates

The economies we have developed have no borrowing or lending. In determining their optimal bids, nodes perform a *constrained* optimization: they only consider placing bids less than their wealth. A functioning credit market removes this constraint, and allows nodes to bid more than they have, with the understanding that nodes will have to pay back these loans with interest. This allows for increased social welfare and truly optimal bids.

In practice however, the overhead of implementation may outweigh the benefits. Interest rates cannot just be set arbitrarily. In a true credit market the interest rate is endogenous, determined by the *demand* for credit, which is difficult to determine in a distributed manner. Nevertheless, credit markets remain an interesting possibility, and one that we plan to explore in the future.

Security

Token economies are susceptible to malicious nodes who place deliberately destructive (and self-damaging) bids. The fact that they are wasting their money would usually be a deterrent, but in a token economy they can wreak havoc without having to pay in 'real' dollars. We envision the token accounts implemented in NIC hardware, but should a malicious user manage to hack the hardware, granting himself unlimited quota, he could cause trouble in the system. Such problems, however, are not unique to token economies: a malicious user who managed to physically modify his 802.11 card would wreak equal havoc. Security techniques, nevertheless, remain an interesting area of study. We briefly consider security issues in the context of decentralization in the final chapter.

Summary

We have described a number of different forms for our token economy. The simplest is that of the open economy, where nodes receive fixed income every period. Simulation results show us that such systems are viable in practice and improve welfare. Each *individual* node determines which transmissions it considers more urgent, and the system ensures that they get priority.

We also discussed the possibility of developing 'closed' economies, in which the total wealth remains constant. These have some appealing economic properties, including the potential to allow nodes to determine bids without formal iteration. Such systems, however, also pose some unique challenges and we considered methods for addressing them.

We end on a positive note. In the last two chapters, we have seen that token economy systems are workable: nodes successfully distribute delay over packets that are less sensitive to it. Where the token-dollar exchange rate is clearly established, as it is in secondary markets – the bidding strategy is simple: nodes bid their true valuation for each packet. Where tokens are purely nominal, the strategy of value iteration helps determine optimal bids.

In the next chapter we address issues unique to multi-packet transmissions, and then conclude with a discussion on decentralized implementations for our system.

VII Multi-Packet Transmissions & Derivatives

Often, a node's transmission does not consist of a single packet. So far, we have focused on single-packet transmissions – and for these, our system behaves as desired. Multi-packet transmissions create two new issues:

First, recall that prices in our system are determined by auction, on a packet-to-packet basis. A node making a multi-packet transmission faces the risk that the price for sending a packet may rise in the middle of its transmission. Can nodes wishing to avoid this risk purchase some form of insurance against it?

Second, consider a node making a multi-packet transmission. The node probably has an assessment of the value of the entire transmission, but how should this value be distributed[89] over the constituent packets for the purposes of bidding? We explore these two questions here.

Derivatives & Uncertainty in the Spot Market

In the system we have developed, nodes bid for a single transmission slot and the current market-clearing price is determined by an auction. The auction results in a *spot* price that changes with network conditions and demand.

P_3	Prob	Payoff
20	10%	0, option not exercised
30	25%	30-25 = 5
35	50%	40-25 = 10
40	15%	50-25 = 25

E[Payoff] = 0.15·25 + ¼·5 + ½·10 = $10

Consider a node wishing to send a file. Since a packet-by-packet auction is held, the node may successfully transmit a few initial packets of the file, only to find that demand has risen, and it is losing its bids for the remaining packets. The file transfer is thus delayed in the middle. Of course, the file will eventually go through, but such 'intra-transmission' delays can be troublesome for applications. The node could respond by increasing its bids for the remaining packets[90]. In this case however, the file will end up costing the node more than it originally expected.

There is nothing *economically* wrong with this scenario. After all, the system was designed to ensure that nodes face the true social cost of their transmissions. If demand rises, transmissions inconvenience more nodes and the spot price *should* rise correspondingly. Nodes must either be willing to tolerate delay, or pay up.

There is however a third alternative: futures, or derivatives. Nodes unwilling to face the variability of the spot market may purchase the right to transmit at a flat price. Specifically, a node may wish to purchase a *call option*: the right to transmit at a fixed price, independent of the prevailing spot price. This would allow it to avoid the risk and uncertainty of spot markets.

Of course, such derivative contracts must themselves be sold for a 'fair' price. In financial markets, this fair price is determined by the venerable Black-Scholes formula, which uses the fact that at equilibrium there should be no *arbitrage* opportunities between an equity and its future contracts. In other words, nodes should not be able to strategically buy futures when the spot price is very low, in order to avoid purchasing transmissions when the spot price is higher[91]. We can utilize the same concept here.

The Traditional Call Option

In financial markets, a call option is a contract that gives one the right – but not the obligation – to purchase a stock at a pre-specified 'strike' price at a given time *T*. For example, an investor may purchase the right to buy a share of IBM for the price of $25, precisely 3 months from now[92].

[89] Traditionally multi-item markets result in combinatorial auctions. Each node reports its valuations for *all* possible combinations of the items it could find useful to a single central auctioneer, which then solves an NP complete allocation problem and informs each node of what it gets. This traditional centralized approach is completely unsuitable for network purposes. Fortunately, we avoid it altogether.

[90] Since the node dynamically recalculates its optimal bids, this will happen automatically.

[91] We draw upon Lazar & Semret's (1999) discussion on bandwidth futures in the following discussion.

[92] This is the standard 'European call', where one can only exercise the option when it matures at time *T*.

What is the value of such a call option? Let us consider a simple example. Assume, that after 3 months a share costs P_3, where P_3 takes only the values in the table with the probabilities indicated.

Three months from now, the node's contract has matured and it is considering its alternatives. If the market spot price P_3 is below \$25, it is better for the node to buy at the market price, and thus holding the option has no value. If the market price is \$30, however, the node will exercise its right to buy the share at the strike-price of \$25. In this case, the value of holding the contract is \$5.

In this manner, we can determine that the *expected* value of holding the contract. From the values in the table, this is just 0.10×0 + 0.25×5 + 0.50×10 + 0.15×25 = \$10. Thus the fair price ψ for the option to buy a share for *p, T* periods from time *t*, when the spot market price is an unknown P_T, is just the expectation[93]

$$\psi \triangleq E\left[P_T - p\right]^+$$

The standard Black-Scholes formula in (7.1) below, gives an exact closed form for ψ, based on the strike price p_{call}, the current stock price p_0, the volatility of the stock σ, and the time to maturity *T*.

$$\psi(p_0, p_{call}, T, \sigma) = p_0 \mathbb{N}(\mathrm{d}) - p_{call} e^{-rT} \mathbb{N}(\mathrm{d} - \sigma\sqrt{T})$$
$$d \equiv \left(\ln(\tfrac{p_0}{p_{call}}) + \left(r + \tfrac{\sigma^2}{2}\right)T\right) \Big/ \sigma\sqrt{T}$$
$$\mathbb{N} \equiv \text{standard normal distribution}$$

The Node's Problem

To address the node's particular problem, the concept of a traditional call option must be extended. Fortunately, this is simple to do. Recall that a call option is the right to buy a share at the strike-price. The option can only be exercised when it matures, *T* periods from the present.

What our node wishes to do is *reserve* the channel for *T* periods, ensuring that it will never pay more than its bid *b* for each period. In other words, it wants the option to buy transmissions *repeatedly* at *b* every period, up to time *T*[94]. The 'fair price' for such a reservation is simply:

$$\Re(T) \triangleq \int_0^T \psi_\tau d\tau \qquad (7.2)$$

Where ψ immediately follows from the Black-Scholes formula above[95].

Reservation Fee

Thus a node wishing to avoid the uncertainty in the spot market may simply purchase such a reservation option. It can place a bid of *b*, and should it win, also purchase a reservation for an amount $\Re$ in (7.2). In buying a reservation contract, the node is ensuring that for the next *T* periods it will never pay more than *b* for a packet.

Plugging in the Black-Scholes formula for ψ into (7.2), and simplifying yields[96]:

$$\begin{aligned} \Re(p_t, b, T, \sigma) &= \int_t^{t+T} \psi_\tau d\tau \\ &= \int_t^{t+T} p_t \mathbb{N}(d) - b\mathbb{N}(d - \sigma_t\sqrt{\tau}) d\tau \end{aligned} \qquad (7.3)$$
$$d \equiv \left(\ln(\tfrac{p_t}{b}) + \tau\tfrac{\sigma_t^2}{2}\right) \Big/ \sigma_t\sqrt{\tau}$$
$$\mathbb{N} \equiv \text{standard normal distribution}$$

This then, is the fair price for our reservation contract. The price depends on four distinct factors, all of which are known to the nodes. The first is the current *market-clearing price*, p_t. This is just the current spot price for making a transmission, determined by the most recent auction. The second factor is the desired *strike price*, *b*. This is the price at which the node wishes to reserve bandwidth.

For example, consider a typical situation in which a node has just won an auction, paying a price $5 - p_t$ is thus 5. Our node now wishes to purchase a reservation allowing it to transmit until time *T*,

93 J. C. Hull. *Options, Futures and Other Derivatives*. Prentice-Hall, 1997.

94 We draw on the formal concept of 'reservations' as described by Lazar & Semret (1999).

95 Intuitively, the node wishes to have the option to buy at p_{bid} in period 1 and in period 2 and… T. This is the same as purchasing separate call options with increasing maturities.

96 Since there is no borrowing in our simple system, the interest rate can be taken as zero.

paying no more than 5 per packet. The strike-price, thus, is also 5.

The third factor is the *time T* for which the node wishes to be able to hold the channel, and the final factor is *volatility* of the spot price for transmissions, σ. This is a measure of how much the cost of a transmission fluctuates. Recall from Chapter V that nodes already track market-clearing prices to update Ω. With this information, the volatility is trivially calculated[97].

τ	d	ψ_τ
9	2	0.36
8	1.7	0.57
7	1.4	0.78
6	1.1	0.96
5	0.8	1.07
4	0.6	1.08
3	0.4	0.99
2	0.2	0.84
1	0.1	0.60
	$\mathfrak{R}$ =	**7.25**

For σ =0.15, p_t=b=10

To make this concrete, consider a simple example. A node wishes to send ten packets in its quest to transmit a small file, but is unwilling to face the risk of fluctuating spot prices. Let us further say that it has just managed to send the first packet of the file, for a cost of 10, and would like to reserve the remaining nine slots at the same price. From its recent historical observations, the node knows that the market price for a transmission has a volatility of 0.15. The table shows the calculation for the integral in (7.3); the reservation fee $\mathfrak{R}$ is 7.25.

Observations

Non-Consecutive Options

In our discussion on reservations, we assumed that the node desired the option to purchase transmissions at *b*, for a *consecutive T* periods. We envisage this to be the most common use of such 'future' options: nodes occasionally wishing to transmit streams of data, and unwilling to take the risks of the spot market during such critical transmissions.

Conceivably, however, a node may wish to purchase non-consecutive reservations. For example, at time *t* a node may know that it will need to send four packets at a set of future times, $F = \{t+1, t+5, t+13, t+15\}$. If our node wishes to avoid the risks of the spot market, it should simply purchase four individual calls, maturing at the times it plans to send. Thus, the reservation fee for such a node would be:

$$\begin{aligned}\mathfrak{R}(p_t, b, F, \sigma) &= \sum_{\tau\in F}\psi_\tau \\ &= \sum_{\tau\in F} p_t\mathbb{N}(d) - b\mathbb{N}(d-\sigma_t\sqrt{\tau})\end{aligned} \qquad (7.4)$$

Where d and $\mathbb{N}$ are the same as before. Non-consecutive reservations thus present no particular challenges, though there are less likely to be used in practice.

Volatility

Of the four factors that determine the cost for a reservation, the volatility is the sole unknown, and must be estimated. We have mentioned that nodes can trivially estimate the volatility of the spot price from historical observations. Recall that nodes continually monitor the clearing-price for each auction to update the Dirichlet parameters Ω. Thus calculating the volatility presents no additional challenges. An important thing to note, however, is that volatility can also be easily reverse-estimated from the price of options[98]. If a lot of options are being sold, nodes can also utilize this 'implied volatility' to estimate the price for a reservation.

Centralization & Implementation

A concern with futures is that they have to be bought *from* someone, and they have to be *honored*. In general, the issuing party must ensure that conflicting futures are not issued. For example, in a network where only one node can transmit at a time, we must ensure that *two* nodes do not acquire a future for the next 10 periods[99]. Such issues lead inevitably to centralization, which is something we wish to avoid. There is no obvious way to offer derivative contracts in a decentralized market, though we briefly consider an approach in the last chapter. For such reasons, it may be better to put the burden of facing risk on the nodes themselves.

[97] For example, if the winning bid at *t* was 12, and the winning bid at *t+1* is 15, the change in price Δ=3. The volatility σ is just the standard deviation of Δ.

[98] See, for example, Bodie, Kane, and Marcus *Investments*. Irwin Press, 1996 or Bodie & Merton, *Finance*. Prentice Hall, 2000

[99] Though technically very desirable, this is not economically necessary. Rather than granting the right to *transmit* at the strike-price, an option actually grants the node the right that its chosen strike price will be taken as equivalent to the *maximum* bid for the auction. The node still faces the usual risk that if another node places the exact same maximum bid (using futures or not), the winner will be chosen randomly.

Black-Scholes Variants
Recall from our earlier discussion that the price of a call-option with a strike price of p, maturing at time T is the expectation:

$$\psi \triangleq E\left[P_T - p\right]^+$$

Calculating this expectation requires a stochastic model for the future market price P_T. The Black-Scholes formula uses a standard geometric Brownian diffusion model and lognormal distributions to estimate the evolution of the spot price. This is a standard stochastic approximation, and one that is widely used.

We may be able to do better by replacing the standard diffusion model with an explicit model for the underlying network traffic. Lazar and Semret, for example, use Poisson arrivals and a heavy-traffic diffusion model to re-derive a variant of the Black-Scholes formula. Their tailored estimate for ψ could be utilized in (7.3). Other even more sophisticated statistical variants may be available in the literature.

Distributing Value in Multi-Packet Transmissions

For some applications, nodes may not have a clear value for *single* packets. A node may value getting a file, retrieving a web page or sending an e-mail, and such tasks typically involve *multiple packets*. Thus, while a user could assign value to successfully sending a file, successfully sending a single packet of the file has a more uncertain value[100].

There are two important things to note about multi-packet transmissions. The first is that packets now exhibit *complementarities*: successfully sending one packet of a transmission without sending the rest has very little value. In addition, the transmission slots in which packets can be sent are often *substitutable*, packets can often be sent at time t, or at time $t+1$, or $t+3$, without significant loss of value.

Our usage-based system is essentially a *sequential* auction, where 'resources' (transmission-slots) are auctioned in sequence. Nodes bid for packets in this order, and can base their bids for packets based on earlier losing or winning bids or on closing prices. In the case of multi-packet transmissions, nodes have a valuation for a *bundle* of packets B, but no independent value for the individual packets comprising it. Thus the question we seek to answer is, that given a bundle B with value v, how should the node 'distribute' this value amongst the packets in B in order to compute individual bids?

There is little work on sequential auctions in the literature. What has been done is focused on the seller, rather than the buyer. There is however, one recent exception. Boutilier and Goldszmidt suggest using dynamic programming to distribute value over individual economic resources that exhibit complementarities. Since our system already uses a dynamic programming algorithm to determine bids, such an approach is a natural extension, and one that we explore here. It may be helpful for the reader to review Chapter V before continuing.

General Sequential Model

Recall that in the case of secondary markets, a node can assign a dollar valuation to its transmissions. When these transmissions involved only single packets – as in the previous chapters – it was optimal for the node to simply bid its true valuation, *\$v*. In the case of multi-packet transmissions, the node must decide how to apportion *\$v* over multiple packets.

We start by presenting a general approach to this question, and then tailor it for our specific needs. Let us denote the right to transmit at time t by s_t. A node wishing to make a multi-packet transmission requires a single *bundle*[101] of such transmission-slots, from a *set* of acceptable bundles.

For example, a node wishing to urgently send a two-packet transmission at time t, may consider any of the three transmission-slot bundles:

$$b^1 = \{s_t, s_{t+1}\},\ b^2 = \{s_t, s_{t+2}\},\ b^3 = \{s_{t+1}, s_{t+2}\}$$

acceptable. Let us denote the set of such *acceptable* bundles for a node by $B = \{b^1, \cdots b^k\}$. The node has a different positive value $v(b^i)$ for each bundle $b^i \in B$.

[100] File-transfers are likely to be the most common multi-packet transmission. In other traffic types, such as real-time streams, each packet carries distinct, useful information, and could conceivably be assigned a value directly.

[101] We draw **extensively** on the terminology, discussion and underlying model in Boutilier & Goldszmidt (1999). We restate and summarize their model and tailor it for our specific purposes.

In our preceding example, the node may value $b^1 = \{t, t+1\}$ more than the other bundles, since value is likely to decrease with delay.

Finally, let us denote by S the current slots the node has used: this is the *status* of its transmission. For instance, $S = \{s_{t+1}\}$ indicates that, so far, the node has managed to send only one packet[102], at time $t+1$.

Let us now consider what computing optimal bids for *individual* packets entails. In our simple example above, the bundle $b^1 = \{s_t, s_{t+1}\}$ has value $v(b^1)$, and the node wishes to apportion this value over the two slots. Naturally, if it thinks that there is likely to be more contention for s_t, a larger portion of v should be allotted to bidding for the first slot. If the node gets s_t it should place a substantial bid[103] for s_{t+1}. If it gets s_{t+1} it is done, but should it fail it can no longer attain b^1, so it focuses its attention on s_{t+2} bidding up to $v(b^2)$ for it. Similarly, if the agent fails to get s_t, it should give up on b^1 entirely and focus its energy on b^3, apportioning $v(b^3)$ over s_{t+1} and s_{t+2}. The agent should simultaneously reason about the *relative* chance of getting a particular bundle for a reasonable price, and focus on the better seeming bundles.

All this suggests using a technique in which the bid for a current resource is *conditional* on the result of earlier bids. Just such a technique has held us in good stead before: value iteration.

Iteration[104]

The state of a node at any point, based on previous outcomes, is determined entirely by three variables: the time t, the status of its transmission S, and its current wealth w. Being in such an intermediate state has an expected value $V_{w,t,S}$, while the value for a terminal state, where a bundle has been acquired, is:

$$V_{w,t,S=b^i \in B} = v(b^i) + w \qquad (7.5)$$

This states that the value of completing a transmission using the slots in bundle b^i is the utility to the node for sending the transmission in this particular manner, plus any remaining funds[105]. Now consider the iteration (7.6) below:

$$V_{w,t,S=(b^i \in B)} = v(b^i) + w$$

$$V_{w,t,S} = \max_{bid \le w} \left\{ p_{bid}^{win} \cdot V_{w-bid,t+1,S \cup s_t} + (1 - p_{bid}^{win}) \cdot V_{w,t+1,S} \right\}$$

$$\pi_{w,c,d} = \arg\max_{bid \le w} \left\{ p_{bid}^{win} \cdot V_{w-bid,t+1,S \cup s_t} + (1 - p_{bid}^{win}) \cdot V_{w,t+1,S} \right\}$$

Imagine a node that has a wealth of w, and by time t has sent some packets S. In other words, consider a node in state $V_{w,t,S}$. In placing a bid, our node knows that should it win, it will send an additional packet in slot s_t, and be charged accordingly[106]. If it loses, its wealth will remain unchanged, and the remaining packets will stay pending. This is precisely what is expressed in (7.6). The utility of each *terminal* state – in which the node successfully manages to complete the transmission – is determined according to (7.5). There are no intermediate rewards, and the only way for the node to attain utility is for it to complete its transmission[107] and reach a terminal state[108].

As usual, at the end of the iteration, π contains the optimal bids for each individual packet s_t in the transmission. These bids depend on the nodes current wealth, and the current *status* of its transmission. Thus our node knows exactly how much it should bid for a single packet – our problem is solved. This system automatically accounts for the complementarities and substitutability of individual packets. It also takes care of issues such as sunk costs and uncertainty. Given correct distributions, the

[102] In practice, the value of a transmission bundle is likely to depend solely on when the *last* packet was sent. Thus we could simplify S by keeping track of only the *number* of packets sent, rather than keeping track of each successful transmission. We make this simplification later, but for now let us consider the more general case.

[103] This can approach $v(b^1)$ since what it paid for s_t is essentially a sunk cost. Obviously, if the node expected that it would end up paying more than $v(b^1)$, it wouldn't have bid on s_t to begin with.

[104] In our exposition we assume that that nodes observe winning bids and maintain the probability distribution p_{bid}^{win} exactly as in Chapter V.

[105] To make this concrete, recall our node urgently wishing to send a two-packet file. Assume that the node's value for $v(b^1)$=\$1.5, while the slower $v(b^2)=v(b^3)$=\$0.5. If the node were to acquire b^1 with \$5 remaining, $V_{w,t,S=b^1} = 6.5$. If it were to get b^2 with \$5 in hand, $V_{w,t,S=b^2} = 5.5$

[106] Upon winning, the node would update $S \rightarrow S \cup s_t$

[107] We assume that the iteration considers states up to $t=N$, and if the transmission is still not complete by then, the exiting state $V_{w,N,S \notin B}$ receives zero or nominal value.

[108] This iteration is conceptually equivalent to those in Chapter V, with the exception that the reward is attained only at the end of the entire transmission, and depends on the *manner* in which it was sent.

computed bidding function is optimal.

Boutilier and Goldszmidt report excellent simulation results in using such a dynamic programming approach to sequential auctions. We are thus confident that the system will work well in practice.

Specific Sequential Model

Our problem with multi-packet transmissions is that they are largely *atomic*. A node has value for the entire transmission, but each packet is worth little or nothing without the others. It is unclear how to valuate each packet given the complementarities with all its counterparts.

We can use the atomicity of transmissions to dramatically reduce the complexity in our solution. In the general scheme above, a node had to enumerate, and iterate over *all* possible bundles in the set *B*. Now consider a node transferring a file. In practice, the node wishes to send, say ten packets, and cares only about the time the entire transfer will take. The exact *slots* in which individual packets end up being sent is irrelevant – the node only cares about *when* the transfer is completed: the sooner the better.

S, the status of the transmission, is now simply an *integer* representing the number of successfully sent packets, or equivalently, the number of remaining packets. There is only one terminal state, *i.e.* when the file is **complete** and the number of packets remaining, *S=0.* The value for being in this state is:

$$V_{w,t,0} = v(t - t_0) + w$$

This asserts that the value of completing this class of transmission depends solely on how long it took to complete it, and not on the particular times its constituents were sent. Given this setup, (7.6) becomes (7.7) below:

$$\begin{aligned} V_{w,t,S=0} &= v(t-t_0) + w \\ V_{w,t,S} &= \max_{bid \le w} \left\{ p_{bid}^{win} \cdot V_{w-bid,t+1,S-1} + (1 - p_{bid}^{win}) \cdot V_{w,t+1,S} \right\} \\ \pi_{w,t,S} &= \arg\max_{bid \le w} \left\{ p_{bid}^{win} \cdot V_{w-bid,t+1,S-1} + (1 - p_{bid}^{win}) \cdot V_{w,t+1,S} \right\} \end{aligned}$$

The number of states in (7.7) is dramatically less than before. The reason is simple: we avoid needless distinctions between $S = \{s_t, s_{t+1}, s_{t+4}\}$ and $S = \{s_t, s_{t+2}, s_{t+4}\}$. All the node cares about in practice is that seven packets of the file are unsent by time *t+4*. Thousands[109] of spurious states $V_{w,t,S}$ can thus be collapsed into one[110]. As usual, π contains the optimal bids for individual packets, and as before, the iteration system automatically compensates for complementarities and the like[111]. This approach is best suited for our particular needs, and as we mentioned earlier, is based on a techniques that have been simulated and verified in practice[112].

Sequential Valuation in Tokens

In usage-based systems, or in a token economy with secondary markets, a user can easily specify the value of a transmission in terms of dollars, and the sequential iteration system distributes this value optimally over the packets in the manner described.

Recall that in a pure token economy, however, tokens have nominal value. Before it can bid a node must have a way to assess the 'value' of a transmission *in terms of tokens*. We discussed this issue in chapters IV and V, and saw how value iteration could help nodes assess this value.
The basic issue is the presence of wealth, *w* in the sequential iteration derived above. The value for successfully completing a transmission, according to

[109] Consider a node wishing to send a three-packet file. Let us say that it wishes the file to be complete by period 10. The number of acceptable bundles of slots in which the file could be sent is 120, thus there are 120 distinct possibilities for *S*. With the specific system, there are only *two*!
[110] If a node actually did care about the exact pattern of its transmissions, the more general approach would have to be used. Of course, if the pattern was *regular* – for example, if the node desired a certain maximum inter-packet delay, the number of bundles, *B* in the general approach could still be dramatically reduced.
[111] We mention one subtle enhancement. In running the iterations in (7.7) and its more general counterpart (7.6), the node uses the p_b^{win} distribution – this is the node's estimate of the probability of winning, given its bid. We have seen how this is built up in Chapter V. From our discussion on derivatives, we have also seen that node can easily determine the *volatility* of p_b^{win}, and use this measure to estimate $p_{t,b}^{win}$ - the expected probability of winning with a bid of *$b*, in period *t*. Since this a sequential auction, p_b^{win} may evolve with time, and so in updating the state $V_{w,t,S}$ in the iteration, it is slightly better to use an estimated $p_{t,b}^{win}$ for the appropriate period *t*, rather than p_b^{win} directly.
[112] Boutilier & Goldszmidt, ibid.

(7.5), depends on the manner it was sent, and the *amount of wealth remaining.* This is perfectly fine if this were real wealth, because the node knows how much value to assign to it. But if this wealth is in the currency of tokens, the node must first have an assessment of the value of having such wealth. We could this in two separate stages, first running an iteration to determine the value of a token, and then distributing this value over the individual packets using the sequential iteration described above. Fortunately, both stages can be combined with a simple change. The reader may wish to refer to (7.7) and (5.9):

$$V_{w,t,S=0} = v_{c,t-t_o} + \beta \sum_{\lambda=c_0}^{c_n} p_\lambda \cdot V_{w,0,\lambda,S=\overline{n}_c}$$

$$V_{w,c,t,S} = \max_{bid \le w} \left\{ p_{bid}^{win} \cdot V_{w-bid,c,t+1,S-1} + (1 - p_{bid}^{win}) \cdot V_{w,c,t+1,S} \right\}$$

$$\pi_{w,c,t,S} = \arg\max_{bid \le w} \left\{ p_{bid}^{win} \cdot V_{w-bid,c,t+1,S-1} + (1 - p_{bid}^{win}) \cdot V_{w,c,t+1,S} \right\}$$

Intuitively, the sole *value* of wealth w is its use in future transmissions. Thus we replace w with the summation, which represents as in (5.9), the value from expected transmissions of various classes that could arise in the future. $\overline{n}_c$ is the expected length of a transmission of class c.

Summary

We have explored two topics relating to multi-packet transmissions. First, we described how nodes unwilling to face the risks of bidding in the spot market might purchase futures that grant them immunity from fluctuating prices. Second, we discussed how nodes might determine their optimal bids for underlying packets, given a value for the entire transmission. Next, we consider distributed implementations for our token economy.

VIII Decentralized Coexistence

In order for our token economy to be practical, we must reduce the overhead of holding centralized auctions every period. In Chapter III, we discussed using a tâtonnement to accomplish just this. We described how an auctioneer could hold auctions *intermittently*; dynamically recognizing that the asking price was becoming too inefficient, and a new auction was needed. Proceeding in this manner significantly improved efficiency, making centralized auctions feasible in practice.

While such an approach may be well suited for networks that naturally have central points, such as a hub, it is equally important to design solutions for completely decentralized systems. Consider for example the most difficult case – an arbitrary, peer-to-peer wireless network. Implementing a token economy in such a distributed system is extremely challenging – but vital to consider. In this final chapter, we explore just such decentralized systems.

Reserving Reception

What does interference really mean? Radio waves, after all, just pass through each other without harm. The problem is distinguishing competing radio signals *at* the receiver. Traditionally, "in order for a sender's transmission to be intelligible, the signal sent by it must be 'louder' than the combination of all other signals received by the receiver"[113]. This is the simple concept behind standard wireless technology: "transmission power is 'focused' into a narrow frequency band, thereby drowning out interference in that channel. The receiver tunes into the channel and comprehends the intended signal, simply because it is much louder than all other competing signals and noise in that narrow channel combined. Naturally, if more than one transmitter uses this strategy for the *same* narrow frequency; neither can be heard by the receiver"[114]

Such technology has traditionally defined economic scarcity in wireless. Each channel takes up not only the bandwidth required to transmit a signal, but also some minimal bandwidth to separate adjacent channels. Thus the total number of channels available in the spectrum is limited.[115].

The problem of coexistence thus is primarily one of *reception*. Ideally, a transmitter would like to *reserve* the area around the receiver, and ensure that no competing *transmissions* take place there. It is well known that attempts at centralized optimal solutions for wireless coexistence typically result in NP-complete *k*-coloring graph problems. They also require complete knowledge of network topology to function. This is clearly intractable in practice: a decentralized solution must be found. We develop precisely such a solution here.

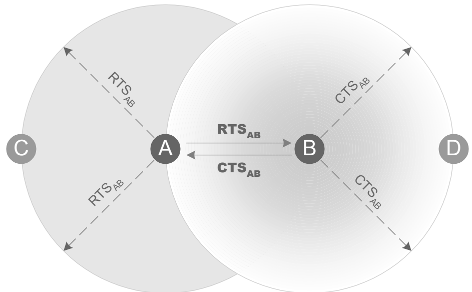
The RTS/CTS Protocol

Imagine a decentralized, peer-to-peer wireless network. In this setting, a naïve CSMA "Listen Before Talk" scheme is ineffective for preventing interference. Before starting transmission, a node must know whether or not there is activity around the *receiver*. CSMA alone will just tell it whether there is activity around it, the *sender*. In practice, wireless systems traditionally use a combination of CSMA and RTS/CTS messages[116] to avoid interference. Let us briefly examine how this works.

A node wishing to transmit sends an **RTS,** a request-to-send message to the receiver. All nodes within range of the transmitter hear this message; but only the intended recipient heeds it. Upon receiving an **RTS** the receiver responds with a **CTS**, a clear-to-send message. All nodes within the footprint of the receiver hear the **CTS** message and realize that a node nearby is going to receive a transmission. They

[113] Benkler, "Overcoming Agoraphobia" 1998, 28.
[114] Benkler, *supra*, 33 edited.

[115] Benkler, *supra.* Newer technologies, such as spread-spectrum/CDMA focus transmission power over the entire available bandwidth, encoding each transmission with a special code. The receiver scans for this code and listens to the transmission. Because of the spread-out low power, the competing transmissions appear as normally distributed noise. This approach is more efficient, allowing for additional transmissions to coexist. It also simplifies the economic allocation problem: there are no longer *specific* spectrum-portions or frequencies to be assigned: merely *K* identical, indivisible items – codes.
[116] For details see Bhargavan et al. 1994, and Karn, 1990



therefore defer their own transmissions and remain silent. In this way collisions are avoided.

Consider a node *A* wishing to transmit to *B*. In other words, it wishes to ensure that no transmissions occur in the vicinity of *B*. It begins by sending *B* an **RTS** frame, and *B* responds to this with a **CTS** frame. Upon receiving the **CTS**, *A* initiates its data transmission.

Let us consider how this avoids interference. The circles in the figure represent the *range* of the signals from *A* and *B*. For example, *D* is within range of *B*. It hears the **CTS** from *B*, and thus knows that it is close to a station that is about to receive a transmission – so it defers from sending anything until the transmission is complete[117]. In contrast, *C* is within range of *A*, but not of *B*. It thus does not hear a **CTS**, and is free to transmit whenever it wishes – its activity will not interfere with reception at *B*. Intuitively, by sending a **CTS**, *B reserves* an area around itself to facilitate *A*'s transmission.

This approach avoids interference, and is a standard method for sending wireless transmissions – the vast majority of peer-to-peer wireless networks use the RTS/CTS protocol. However since the allocation that results is completely *random*, the scheme suffers from the issues discussed in Chapter I: it is economically inefficient, and susceptible to tragedy.

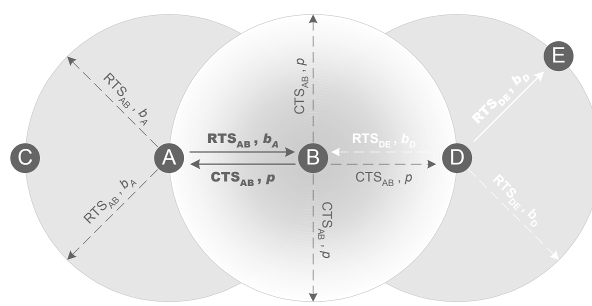
Decentralized Economy with RTS/CTS

Recall that a transmitter wishes to *reserve* the area around the **receiver**, and this reservation means that other nodes in **that** region cannot **transmit**. This causes them inconvenience, and imposes social costs.

As we have discussed repeatedly throughout this work, in (2.7) etc, these social costs should be internalized by charging a *price* for transmissions that cause this inconvenience.

A centralized approach is clearly intractable. The set of nodes affected by reserving the receiver's footprint is unknown to the transmitter. For example, *A* may be blissfully unaware of *D*'s existence. Even if *A* were aware of *D*, it has no way of knowing *how* much *D* currently values transmitting, and thus has no means of estimating the cost imposed on it should it have to forego its transmission. To make matters worse, the set of nodes may change from time to time, as wireless nodes tend to be semi-mobile. In such a space, how is *A* to determine the social cost its transmissions impose?

Consider piggybacking 'bids' onto **RTS** messages. A transmitter places a bid, indicating its value for a successful transmission, inside an **RTS** message and sends it to the receiver as usual. All nodes habitually monitor **RTS** messages that they overhear, keeping track of bids in their area[118]. Thus every node knows full well the *social cost* of reserving the region around it. Should it receive a transmission, its neighbors will be forced to forego transmitting – and since it has been hearing their bids, it knows exactly how much this will inconvenience them.



Let us make this concrete. Consider our transmitter *A* wishing to send to receiver *B*. In other words, it wishes to reserve the area around *B*, and prevent other outgoing transmissions from occurring therein.

[117] *D* is free to receive a transmission – the CTS merely obliges it not to *send*.

[118] *B* only hears *RTS* frames originating from all nodes *within* its footprint, the central circle. By monitoring the bids in these frames, *B* knows the value nodes within its footprint place on their transmissions. Thus it knows the true social cost imposed by reserving the footprint and forcing nodes within it to forego their transmissions.

Accordingly, A sends B an $\mathbf{RTS_{AB}}$ message containing its bid for the transmission, b_A.

B, like all other nodes, has been monitoring the bids in **RTS** messages originating in its region, and thus knows how much transmitters around it value their transmissions. In the figure, B has recently observed an **RTS** from D to E, carrying a bid of b_D. In a similar manner, B has observed the recent bids of other nodes within its region, and thus knows the social costs that sending a **CTS** to A will create.

When B receives $\mathbf{RTS_{AB}}$ it compares the attached bid b_A with the social cost for reserving its region. In essence, B runs a Vickrey auction: if K concurrent transmissions can occur, it compares A's bid with the $K+1^{th}$ highest recent **RTS** that it has seen[119]. If *A's* bid is higher than the current price in the region, B responds with a **CTS** marked with the Vickrey price, which A debits from its account. Nodes hearing the **CTS**, such as D, realize that the area has been reserved and refrain from transmitting[120].

In this simple, distributed manner, transmitting nodes pay for the inconvenience they cause, ensuring optimal usage of the network.

Observations

The grace of this mechanism lies in the fact that it is entirely distributed: nodes rely purely on local information. Recall that the optimal price depends solely on the set of nodes inconvenienced, and determining this set in a centralized manner is virtually impossible. Our scheme, however, puts the burden of determining who is inconvenienced, and by how much, on the *receiver* – who has access to precisely this information. Moreover, we accomplish all this without overhead. There are no additional transmissions, no formal auctions to be held, and no delay engendered in the system. The latency in our approach is directly comparable to that in standard wireless RTS/CTS systems – with the crucial difference that the resulting allocation prevents greed and is economically efficient and welfare maximizing.

With this system we can envision a completely distributed, peer-to-peer wireless token economy. Using the sundry techniques we have described in previous chapters, nodes *locally* determine their own optimal bids, win or lose auctions using the system above, and debit their own accounts. Nodes face prices that reflect social costs, greed and tragedy are avoided, and network resources are always utilized optimally. And all this is accomplished in an entirely decentralized fashion.

Probability Distributions

An interesting subtlety with our RTS/CTS system occurs in the case where there are *no* secondary markets. Recall that in this case, nodes must determine their optimal bids in purely nominal tokens, using value iteration. As we discussed in Chapter V, this requires nodes to have an estimate of p_b^{win}, the probability of winning given a bid b. Intuitively, there could be slightly different levels of demand in regions around different receivers. We could address this by requiring nodes to maintain a distinct $p_{bid,R}^{win}$ for each receiver, R that they talk to. This of course, would complicate the value iteration. In calculating the expected value of tokens, the node would now also need to estimate the relative frequency with which it makes transmissions to different receivers. In practice, it may be best to assume reasonably consistent activity in the network. Where secondary markets *do* exist, of course, there is no need[121] for the node to maintain p_b^{win}: it simply bids its true dollar valuation directly, making the issue moot.

Decentralized Economy with Random Access

Let us step back for a moment and re-think the problem of distributed access control. Since we determine the right of access to the medium using auctions, we naturally think of a central auctioneer. Even in the entirely distributed RTS/CTS system

[119] In practice, this should be based on a statistical function, such as the *average* of recent $K+1^{th}$ highest bids, rather than a single sample.

[120] Note that D can *receive* data while A is transmitting to B. A **CTS** only obliges nodes hearing it to refrain from *transmitting*. Note also that C can both transmit and receive.

[121] Even with secondary markets, the node may still need to maintain p_b^{win} for multi-packet transmissions where it is uncertain of the value of individual packets. It would then use it to run the iterations described in the previous chapter.

above, the receiver *B* was essentially acting like an auctioneer.

In practice, distributed wireless systems use random access schemes to allocate the right to transmit. For example, in the standard FCC UPCS etiquette, a device must find the channel unused for a *monitoring-time* before it can initiate a transmission. Nodes wait out their monitoring periods, and the first node to grab the channel gets to transmit: if there is a collision, randomized backoff picks a winner. Such random access is essentially an auction for control of the medium – where the winner is chosen randomly.

In a series of papers Peha et al. (1997) have proposed varying device monitoring-times to provide nodes with a disincentive to consume network resources. After successfully using the channel, the node is forced to monitor the channel for an additional 'penalty time' before it can initiate another transmission. We discussed such 'penalty schemes' and their shortcomings in Chapter I, but we can exploit the concept of modifying monitoring times to implement yet another decentralized version of the token economy.

Consider making the monitoring time inversely proportional to the *bid* of the node[122]: the higher the bid, the shorter the monitoring time. A node with a high bid will thus have a statistically greater chance of grabbing the channel in a random access scheme[123]. Consider a channel that is currently held by a node. Two other nodes *A* and *B*, wish to make a transmission, and have individually determined their optimal bids. Let us assume, arbitrarily, that $b_A > b_B$. Once the transmission ends, both *A* and *B* will begin to monitor the channel. Since the monitoring time is inversely proportional to the bid, *A* will grab it first and make its transmission. By the time *B's* monitoring time runs out the channel will be busy. Thus the highest bidder will win.

Finally, we assume that once *A* has grabbed the channel, it initiates transmission by announcing its winning bid. In this manner other nodes can observe the winning bids, updating their probability distributions as described in Chapter V. Best of all, there are no significant overheads and no increase in latency. We have the best of both worlds: the simplicity of random access-control schemes, and the economic efficiency of the token economy.

[122] It must be ensured that there is sufficient monitoring time to avoid collisions. In addition the proportionality to *monitoring-time* does not need to be linear: bids should be *linearly* proportional to the increasing probability of successfully grabbing the channel.

[123] Additionally, the exponential *backoff time* could also be reduced for nodes with high bids.

Issues with Decentralization

Strategic Behavior & First-Price Auctions

Note that our simple random-access technique amounts to a first-price sealed bid auction. The winner announces and pays its winning bid[124], but *nothing* is known about the bids of nodes that failed to grab the channel. This means that optimal bids can be determined exactly as described in Chapters V and VI. In our discussion in Chapter II, we saw that first-price auctions can be susceptible to strategic behavior, which can reduce efficiency. As we mentioned there, however, the gain from such strategic behavior falls rapidly with the number of players, and strategizing is only viable for games of three or four. Secondly, since our value iteration process already calculates individually optimal behavior for each node, based on equations (5.9), and there is no additional gain from strategizing. Finally, the first-price auction is actually simpler to implement. Recall from Chapter V that it requires less computation and estimation on the part of the node.

Monitoring Time & Bids

One problem with having a *fixed* relationship between monitoring time and bids is that in periods of low activity, when bids are likely to be low or zero, nodes will have to wait out the full monitoring time. It may be better to have a sliding scale for the bid vs. monitoring time relationship, centered at the previous winning bid, which is known to all.

Decentralized Accounting & Security

In both distributed systems described above nodes determine their own bids, place them, and should they win, debit their own accounts. They also grant themselves funding every period, as in Chapter VI.

[124] This is roughly verifiable by the other nodes based on how long it was between the channel becoming idle and being grabbed again.

The system thus proceeds in an entirely decentralized manner. Of course, the decentralization hinges on the fact that nodes will indeed debit and credit their accounts appropriately.

We envision the token economy implemented in NIC firmware, and thus consider the security risks to be minimal. Should a malicious user modify the hardware on his card to give himself, say, an infinite token account, he could cause problems in the system. This scenario is hardly unique to the token economy, a malicious user with a modified 802.11 card for example, would wreak equal havoc. Nevertheless, it may be useful to have some form of verification. Let us say that a node has been transmitting excessively over a long period of time. Such 'suspicious' nodes may be *audited* by a peer, or a trusted authority. During such an audit, the authority could ask the node to head its transmissions with its current wealth, suitably encrypted. The authority could then observe if it were being maintained correctly. We do not explicitly address the issue of distributed security in this work, but we do note that if the token economy is implemented in firmware, such scenarios are unlikely.

Decentralized Future Contracts

One final concern with decentralization is with *futures*. In a decentralized economy, a node can individually determine the optimal price for an options contract directly from (7.3). It can conceivably also 'purchase' such a future and debit its account accordingly. But since such 'reservations' are being sold in a decentralized manner, they can conflict. Consider two nodes that have purchased a reservation for the channel for the next 5 periods. If the capacity of the network $K=1$, then only one of them will actually be able to transmit.

The simplest approach to this would be to simply let nodes with overlapping options vie for the channel. From an economic viewpoint this is strictly efficient: since both have the right to have their bids considered the 'maximum' there is a tie, and normal ties in our system are broken randomly. There is, however, a more practical approach. Consider a node that has decided to purchase a reservation. It calculates the price, debits its account, and *announces* it to the channel. Other nodes then know that the channel is reserved from time t_0 to t_N, and thus do not attempt to purchase conflicting reservations.

The Vision

Ferguson (1989) describes three characteristics of a typical economy. The first is competition: agents selfishly compete for resources and do not work together, yet the outcome is welfare maximizing. The second is decentralization: an economy is populated with independent agents, each making his own decisions, based on his own individual goals and agendas, and using his own endowment of wealth to do so. Finally, an economy uses the price-mechanism, which provides both a metric for the value of a resource, and ensures that the resource goes to whoever values it most. The wealth endowment itself defines the importance and priority of each agent.

We have created an artificial economy that ensures efficient coexistence – and our system shares all three of these characteristics. The price mechanism curbs greed and prevents tragedy whilst ensuring that nodes that most desire access are the ones to receive it. Moreover, this can occur without centralization. Using the techniques we have developed in previous chapters, nodes observe channel activity, grant themselves income, maintain probability distributions, calculate and place optimal bids and make competing transmissions – all in a distributed manner. In the case of secondary markets, things are even simpler; nodes buy tokens offline and simply bid their true valuation.

More specifically, we have seen that simple token economies can improve welfare over other methods we discussed in Chapter I. For example, recall the 'penalty schemes' which curbed greed at the expense of throughput by penalizing every transmission with a *fixed* delay. In contrast, the token economy allows nodes to use their endowments to dynamically distribute delay over transmissions, allowing them to optimize their own *individual* agendas. In doing so, it ensures economic efficiency, something other mechanisms do not attempt to address. Moreover, we have also seen how to easily adapt *existing* wireless systems to create distributed token economies.

Economic solutions to coexistence issues are thus clearly viable. This has been a first step, and there is much yet to explore. We envision the next as a test-bed of devices forming a token economy, and we intend to take it in the near future.

IX References

Given its exploratory nature and breadth, this work draws very heavily upon numerous sources, especially when providing summarized, contextual background. Credit is formally granted to all these respective authors:

1. Abramson, N. “The ALOHA System - Another Alternative for Computer Communication.” In *Proceedings AFIPS Fall Joint Computer Conference*, pages 281-285, AFIPS Press, NJ, 1970.
2. Adda, J. and Cooper, R. *Dynamic Programming: Theory and Applications*, forthcoming, MIT Press
3. Angeletos, G. Personal interviews, Apr–Jul, Massachusetts Institute of Technology, 2002.
4. Arrow, K.J. and Intrilligator, M.D. *Handbook of Mathematical Economics*. North-Holland Publishing Co., Amsterdam, 1981.
5. Benkler, Yochai. "Overcoming Agoraphobia: Building the Commons of the Digitally Networked Environment." *Harvard Journal of Law and Technology*, 11 # 2 (Winter 1998): 287-400.
6. Bertsekas, D. and Gallager, R. *Data Networks*. Prentice-Hall, Englewood Cliffs, NJ, 1987.
7. Bharghavan, V. et al. "MACAW: Media Access Protocol for Wireless LANs." *ACM SIGCOMM*, 1994.
8. Boutilier, C. and Goldszmidt, M. “Sequential Auctions for the Allocation of Resources with Complementarities.” In *Proc. 16th International Joint Conference on Artificial Intelligence*, 1999.
9. Brain, M. “How the Radio Spectrum Works.” In *How Stuff Works*, Online <www.howstuffworks.com /radio-spectrum.htm>
10. Cassady, R. *Auctions and Auctioneering*. Berkeley, University of California Press, 1967.
11. Cheng J.Q. and Wellman M.P. “The WALRAS algorithm: A Convergent Distributed Implementation of General Equilibrium Outcomes.” *Computational Economics*, 12:1--24, 1998.
12. Doepke, M. Lehnert, A. and Sellgren, A. *Macroeconomics*, University of Chicago 1999
13. Drexler, E. and Miller, M. “Incentive Engineering for Computational Resource Management.” In *The Ecology of Computation*, pages 231-266. North-Holland, Amsterdam, 1988.
14. Drexler, K. and Miller, M. “Markets and Computation: Agoric Open Systems.” In *The Ecology of Computation*, 133-176. North-Holland, 1988
15. Ellison, G. Personal interviews, Apr–Jul, Massachusetts Institute of Technology, 2002.
16. Ferguson, D. “The Application of Microeconomics to the Design of Resource Allocation and Control Algorithms.” *Ph.D. dissertation*, Columbia University, 1989.
17. Ferguson, D. Yemini, Y. and Nikolaou, C. “Microeconomic Algorithms for Load Balancing in Distributed Computer Systems.” *8th Int. Conf. on Distributed Computing Systems*. 491-499, June 1988.
18. Gumalla, C.V. and Limb, J.O “Wireless Medium Access Control Protocols." in *IEEE Communications Surveys & Tutorials*, 2000.
19. Hardin G. “The Tragedy of the Commons”, *Science*, Vol.162, 1968, pp. 1243-1248.
20. Harkavy, M. Tygar, J. and Kikuchi, H. “Electronic auctions with private bids.” In *Proceedings of the 3rd USENIX Workshop on Electronic Commerce*, pages 61--74, 1998.
21. Hazlett, T. “Spectrum Flash Dance: Eli Noam's Proposal for Open Access to Radio Waves.” *Journal of Law & Economics* XLI (2), Oct 1998
22. Hazlett, T. “The Wireless Craze and the Unlimited Bandwidth Myth.” *Harvard Journal of Law & Technology*, Spring 2001, 335-567
23. Holtkamp, R. et al. “A Distributed Media Access Control for Wireless ATM Networks.” *5th International Conf. On Telecommunication Systems*, Nashville, Mar 1997.
24. Jordon J.S. “The Competitive Allocation Process is Informationally Efficient Uniquely.” *Journal of Economic Theory*, 28:1-18, 1982.
25. Karn, P. “MACA: A New Channel Access Method for Packet Radio.” In *ARRL/CRRL Amateur Radio 9th Computer Networking Conference*, pages 134-140, 1990.
26. Kearns, J. “Dynamic Optimization.” In *Notes for Macroeconomic Theory III*, Massachusetts Institute of Technology, 2001.

27. Krumke, S.O. Marathe, M.V. and Ravi, S.S. "Approximation algorithms for channel assignment in radio networks." In *2nd International Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications*, Dallas, Texas, 1998.

28. Laibson, D. "The Eat-The-Pie Problem." In *Solutions for Economic Theory,* Harvard University, Fall 2001. Online < http://icg.harvard.edu/~ec2010c/>

29. Lazar, A. and Semret, N. "Design and Analysis of the Progressive Second Price Auction." In *Telecommunication Systems*, special issue on Network Economics, 1999.

30. MacKie-Mason, J. and Varian, H. "Economic FAQs on the Internet." *Journal of Economic Perspectives*, 8(3):75-96, Summer 1994.

31. MacKie-Mason, J. and Varian, H. "Generalized Vickrey Auctions." *Technical report*, Dept. of Economics, Univ. of Michigan, July 1994.

32. MacKie-Mason, J. and Varian, H. "Pricing Congestible Network Resources." *IEEE Journal on Selected Areas in Communications*, 1995.

33. MacKie-Mason, J. and Varian, H. "Pricing the Internet." In *Public Access to the Internet*, Prentice Hall, 1995.

34. MacKie-Mason, J. and Varian, H. "Some Economics of the Internet." In *Network infrastructure, and new tasks for regulation*, Ann Arbor: The University of Michigan Press, 107-136. 1996.

35. McAfee, P. and McMillan, J. "Auctions and bidding." *Journal of Economic Literature*, 25(2):699-738, June 1987

36. McPhillips, L. "Internet Resources By The Slice." *Network Magazine.* CMP Inc. Aug, 2002.

37. Milgrom P. and Weber J. R. "A Theory of Auctions and Competitive Bidding." *Econometrica*, 50:1089-122, 1982.

38. Milgrom, P. "Auction Theory." In *Advances in Economic Theory: Fifth World Congress*, Cambridge University Press, 1987

39. Milgrom, P. "Auctions and Bidding: A primer." *Journal of Economic Perspectives*, 3(3):3 22, 1989.

40. Milgrom, P. Personal interviews, Apr–Jun, Massachusetts Institute of Technology, 2002.

41. Noam, E.M. "Taking the Next Step Beyond Spectrum Auctions: Open Spectrum Access." *IEEE Communications* vol. 33, no. 12, Dec. 1995, pp. 66–73.

42. Parr, R. *Learning Probability Distributions*, in Numeric Artificial Intelligence, Duke University, 2001.

43. Peha, J.M. "Spectrum Management Policy Options." *IEEE Communication Surveys*, Mar 1998.

44. Peha, J.M. "The Path Towards Efficient Coexistence in Unlicensed Spectrum." *IEEE 802.16 Wireless High-Speed Unlicensed Metropolitan Area Network Standards*, April 2000.

45. Peha, J.M. "Wireless Communications and Coexistence for Smart Environments." *IEEE Personal Communications*, 2000.

46. Reynolds, K. *Auction History*. Agorics, Inc. 1996

47. Salgado, Sirbu & Peha. "A Narrow Band Approach to Efficient PCS Spectrum Sharing." *IEEE Personal Communications*, Vol 4, #1, Feb 1997

48. Sandholm, T. "Distributed Rational Decision Making." In *Multiagent Systems A Modern Approach to Distributed Artificial Intelligence*. MIT press 1999, pp. 201

49. Satapathy, D.P. and Peha, J.M. "Etiquette Modifications For Unlicensed Spectrum: Approach and Impact." In *Proceedings of the 48th Annual International IEEE Vehicular Technology Conference*, May 1998, vol.1, pp. 272-276.

50. Satapathy, D.P. and Peha, J.M. "Performance of Unlicensed Devices With a Spectrum Etiquette." In *Proceedings of IEEE Globecom*, Nov. 1997, pp. 414-418.

51. Satapathy, D.P. and Peha, J.M. "Spectrum Sharing Without Licenses: Opportunities and Dangers," *"Interconnection and the Internet"* In *Papers From The 1996 Telecommunications Policy Research Conference*, Lawrence Erlbaum Associates, 1997, pp. 49-75.

52. Semret, N. and Lazar. A. "Spot and Derivative Markets in Control." *16th International Teletraffic Congress*, Edinburgh, UK, June 1999.

53. Sobol, I.M. *The Monte Carlo Method*. Moscow: Mir, 1975.

54. Steer, D.G. "Coexistence and Access Etiquette in the United States Unlicensed PCS Band.'' *IEEE Personal Communications,* vol. 1, no. 4, Fourth Quarter 1994, pp. 36–43.

55. Steer, D.G. “Wireless Operation in the Unlicensed Band.” *Proc. Intl. Conf. on Universal Personal Communications*, Sept. 1994, pp. 586-90.

56. Steiglitz, K. Honig, H. and Cohen, L.M. “A Computational Market Model Based on Individual Action.” In *Market-based Control, A Paradigm for Distributed Resource Allocation*, World Scientific, Palo Alto, 1996.

57. Stokey, N.L. Lucas R.E. and Prescott E.C. *Recursive Methods in Economic Dynamics*, Harvard University Press, 1989.

58. Tanenbaum, A.S. *Computer Networks*, third edition, Prentice Hall 1996.

59. Temin, P. Personal interview, Mar 26th, Massachusetts Institute of Technology, 2002.

60. Varian, H. “Economic Mechanism Design for Computerized Agents.” In *USENIX Workshop on Electronic Commerce*, New York, NY, July 1995.

61. Varian, H. *Intermediate Microeconomics*. W. W. Norton and Company, 1987.

62. Vickrey, W. “Counterspeculation, Auctions, and Competitive Sealed Tenders.” *Journal of Finance* Vol. 16, March, 1961 pp. 8-37.

63. Waldspurger, C.A. et al. “Spawn: A distributed computational economy.” *IEEE Transactions On Software Engineering*, 18(2):103-117, February 1992.

64. Waldspurger, C.A. “Lottery and Stride Scheduling”, *LCS Technical Report 667*, MIT 1995

65. Wang, Q. Peha, J.M. and Sirbu M. “Optimal Pricing for Integrated-Services Networks with Guaranteed Quality of Service.’’ In *Internet Economics*, MIT Press, 1997, pp. 353–76.

66. Wellman, M. “A market-oriented programming environment and its application to distributed multi-commodity flow problems.” *Journal of Artificial Intelligence Research*, 1:1-23, 1993.

67. Wellman, M. et al. “Auction protocols for decentralized scheduling” *Technical report*, University of Michigan, July 1998.

Appendix A An Example on λ

We describe a simple Poisson model for λ, the probability distribution for incoming packets used in the value iterations. Our aim is entirely expository: the model is simple, and serves only as an example. As such, it is essentially a technicality, and only the zealous reader should proceed.

Poisson Arrivals and λ

λ is our sole exogenous probability distribution that determines the likelihood that a node gets a new packet of type *c*. We asserted in Chapter V that in addition to standard historical techniques, λ could be determined using an arbitrarily complex network traffic model of our choosing. To make this conceptually clear, we briefly describe a simple Poisson model with a queue here.

Let us assume that there are only two classes of service $c = 1$ and $c = 2$; $c = 0$, as usual represents having no packet. Packet arrivals are Poisson, with arrival rates α and β respectively. Finally, let us make the usual assumption that the inter-arrival time is large, compared to each period. Then the probabilities of a packet of class 1 or 2 arriving within a single period are α and β respectively.

Assume that the packet at the head of the queue arrived in period 0, and had to wait in line for *d* periods before being sent. All this while, more packets were building up behind it. Given *d*, consider the probability that the *next* packet in the queue is type 1. Either this packet arrived in period 1, or *no* packet arrived in period 1 and it arrived in period 2, or no packets arrived in period 1 and 2, and this one arrived in period 3… The probability, then, of a packet of type 1 being directly behind the current packet, given that the current packet has waited *d* periods in queue is:

$$\sum_{i=0}^{d}(1-\alpha-\beta)^d\alpha .$$

The probability of there being no packet waiting is just $(1-\alpha-\beta)^{d+1}$

Now consider the open economy value iteration:

$$V_{w,d,c} = \max_{bid \le w}\left\{\begin{array}{l} p_b^{win}\left(\begin{array}{c} \alpha V_{w+\mu-b,d,1} \\ u_{c,d} + \beta V_{w+\mu-b,d,2} \\ (1-\alpha-\beta)V_{w+\mu-b,d,\emptyset} \end{array}\right) \\ +(1-p_b^{win})V_{w+\mu-b,d+1,c} \end{array}\right\}$$

$$V_{w,d,\emptyset} = \left(\begin{array}{l} \alpha V_{w,d-1,1} \\ +\ \beta V_{w,d-1,2} \\ +\ (1-\alpha-\beta)V_{w,d-1,\emptyset} \end{array}\right)$$

The state of the node is as usual defined by *w,c,d,* the wealth of the node, the class of the packet it has and the *total* delay that packet has faced in the queue. If the node loses, it gets an extra period of income, while the packet suffers additional delay.

If the nodes wins, it has sent a packet of type *c* after a period of delay *d*, and thus receives a total reward of $u_{c,d}$. It also receives income, and in addition, gets a new packet from the buffer that has already suffered some delay *d* from waiting in the queue. What is the chance that this new packet is of type 1, and has suffered a delay of exactly *d* so far? This would mean that the new packet had arrived *directly* after the one just sent. The probability of this happening is just α. What is the chance that this new packet is of type 1, but has only suffered a delay of *d*-1 so far? This means that *no* packet arrived directly after the one just sent, and a type 1 packet arrived the period after. The probability of this is just $(1-\alpha-\beta)\alpha$. Similarly the chance that it has suffered a delay of *d-2* is $(1-\alpha-\beta)^2\alpha$ and so on.

Note how the iteration takes care of this: with probability α we directly transition to a new packet of type 1 with current delay *d*. With probability $(1-\alpha-\beta)$ we transition to a null state (representing that no packet came immediately after the one just sent), which just immediately redirects us with probability α to the state representing a packet of type 1 with delay *d-1*. The total probability of reaching this state is thus $(1-\alpha-\beta)\alpha$, exactly as we saw above.

This simple, highly stylized model should make it clear how λ can be modeled in practice. Running the value iteration above will result in the optimal bids for a network where traffic exhibits such arrival probabilities. In practice of course, as we have discussed, λ can be made arbitrarily complex.