Pirated for profit

JOSHUA SLIVE University of British Columbia
DAN BERNHARDT University of Illinois at Urbana Champaign and Queen's University

Abstract. This paper explains why a software manufacturer may permit limited piracy of its software. Piracy can be viewed as a form of price discrimination in which the manufacturer sells some of the software at a price of zero. In the presence of significant network externalities for the software, it may be profit maximizing for the software manufacturer to tolerate piracy by home consumers, most of whom have a low willingness to pay. This can increase the demand for the software by business users. JEL Classification: L2

Profits en se laissant pirater. Ce mémoire explique pourquoi un manufacturier de logiciel peut permettre un degré limité de piraterie de son logiciel. La piraterie peut être considérée comme une forme de discrimination par les prix dans laquelle le manufacturier vend une quantité de logiciels au prix zéro. S'il y a des externalités importantes de réseau pour le logiciel, il peut s'agir d'une stratégie de maximisation de profit pour le manufacturier que de tolérer la piraterie par des utilisateurs dans leur foyer, puisque la plupart ont une propension à payer assez faible. Voilà qui peut augmenter la demande pour le logiciel par les utilisateurs du monde des affaires.

1. Introduction

News stories lamenting the large amount of software piracy in Canada and worldwide have become a regular feature of North American media. Software manufacturers, through their trade organizations, release statistics asserting that huge damage is inflicted on their businesses by the illegal use of software. The Business Software Alliance claims that the industry loses '$13 billion per year, $35 million per day, and $407 per second from software piracy' (*Newsbytes*, 3 Jan. 1995). The Canadian Alliance Against Software Theft argues that 59 per cent of all software

This paper is derived from Joshua Slive's undergraduate honours thesis. We thank John Duggan, Lorenzo Garlappi, Burton Hollifield, Frank Lewis, John Spicer, Dan Usher, and participants at the UBC MIS workshop for their comments. Financial support from SSHRC is gratefully acknowledged. The authors take sole responsibility for any errors.

Canadian Journal of Economics Revue canadienne d'Economique, Vol. 31, No. 4
October / octobre 1998. Printed in Canada Imprimé au Canada

0008-4085 / 98 / 886–99 $1.50

used in Canada is illegal, costing the industry $316 million (Canadian) per year (*Financial Post*, 28 May 1994). According to Computer Reseller News, 35 per cent of business software in the United States is pirated (7 Nov. 1994, 148(1)).

These numbers, which are enormous relative to the size of the industry, suggest a large gap in the enforcement of copyright law. It would seem that the amount that the software industry is spending on enforcement and prevention is far below the optimal level. The scope of the problem appears to call for large investments in anti-copying technology and enforcement of copyright laws. However, software manufacturers continue to spend only a small portion of their total production costs on anti-piracy measures. Indeed, evidence can be found to assert that software manufacturers lend their tacit approval to piracy. For example, some companies allow fully functional *evaluation versions* of commercial products to be freely downloaded from the World Wide Web. In other words, the tough talk of the software manufacturers is not being backed up by action.

Given this seemingly illogical allocation of resources by software manufacturers, a fresh perspective on the software market seems necessary. Why do software manufactures find this low level of anti-piracy spending, with its accompanying high level of piracy, to be optimal? In this paper we examine the possibility that, under certain circumstances, manufacturers may be better off with piracy than without.[1]

This paper identifies two features of the software market (other than costly enforcement) that, in combination, can lead manufacturers to wink at limited piracy:

1. There are significant network externalities in software consumption (Gandal 1994, 1995; Shurmer 1993; Church and Gandal 1992).
2. Different consumer groups vary substantially in their willingness to pay for software. In addition, they have different likelihoods of being caught pirating and different penalties if they are caught.

In the remainder of the paper, we discuss how network externalities and price discrimination can be combined to explain the optimizing behaviour of software manufacturers. We create a simple model that explains the seemingly low expenditures of manufacturers on anti-piracy measures. Two widely known software products are then discussed in the context of the model, and the evidence supporting the basic premises of the theory is examined.

2. Network externalities

A network externality arises whenever the value of a product to consumers is positively related to the number of people using the product (David 1985; Katz and Shapiro 1985). These externalities can generally be divided into two types: direct and indirect. Direct network externalities, as the name suggests, occur when the

1 A similar issue is addressed from a purely empirical perspective in Givon, Mahajan, and Muller (1995).

positive benefit of the externality is generated directly by the increased number of users of the product. The most obvious examples of this are communication networks such as e-mail systems. The more people on the Internet, the more valuable e-mail is to its users, because there are more people who can be contacted.

Indirect network externalities occur when some second-round or after-market effect of an increasing user base increases the value of the product, for example, automobiles and automobile service technicians. The greater the number of people who own a specific automobile, the more expertise will be available when it comes time to repair it. The increased value is not generated directly by the increase in the number of car owners, but rather is caused indirectly when more car owners increase the availability of after-market service.

Network externalities are an important feature of the computer software market. Because people often wish to exchange data files, software exhibits direct network externalities in the form of a communications network. For example, Wordperfect was more valuable when it was the de facto standard format for interchanging word-processing documents in the business environment.[2] Word processors that could not read and produce documents in this format were less valuable, regardless of other features.

An important indirect network externality in the computer software market is the availability of after-market products and services. For some types of software a large market has developed to supply computer manuals, training seminars, and product support to the user base of a particular brand of software. In addition, for many market segments, the availability and quality of add-on products has become a crucial determinant of the value of software. Products like *plug-ins* for Netscape Navigator and macro packages for Lotus 1-2-3 add extensive functionality to the base software and are often designed so that they work only with specific brands of software. All these products enhance the value to consumers of the software. The more people who use a specific type of software, the larger will be the supply of these after-market products, which in turn will increase the value of the software.

Perhaps the most important indirect network externality in the computer software market is generated by knowledge of how to use software. Because of the complexity of many types of software, a significant investment in time and resources is often necessary in order to master a product. The more popular a specific brand is, the more expertise there will be available on how to use it. More important for our model, this knowledge is tied to the individual user and can be costlessly transferred between the home and the work environment. A person who uses Wordperfect to do word processing at home will need less training and be more efficient if he or she also uses Wordperfect in the office.

The importance of product knowledge for the valuation of computer software is evident from the database and spreadsheet markets of the early 1990s. Dbase and Lotus 1-2-3, respectively, were the dominant players in these market segments through much of the first decade of the personal computer. By 1990 these products

2 Just joking; this paper was written in $\TeX$.

were dominated both in features and in price by new market entrants. Yet despite the superiority of their rivals, the veterans maintained a large share of the market for several years. Efforts by the new entrants to capture network externalities by making their products file compatible with that of the industry leaders proved insufficient. The decisive factor for many purchasers had little to do with price, features, or even file compatibility. From the perspective of a large organization, 'The cost of a transition, retraining an entire organization, and rewriting applications, very likely outweigh whether a package can turn a spreadsheet into an outline' (Parker 1991). Of course, the superior new entrants did win out in the end. The fact that the older companies could continue for years to generate huge revenues based on past successes, however, shows the importance of knowledge network externalities in the computer software industry.[3]

3. Price discrimination

Distinct groups of customers differ dramatically in their willingness to pay for software. To make this notion concrete, we identify two distinct groups of consumers: business consumers and home consumers. For many software applications most home consumers have a lower willingness to pay than business consumers. Network externalities are also more significant for business consumers in many applications. A business gains, for instance, when a newly hired employee is already familiar with the software the business uses. Finally, it is far more difficult to catch and punish home consumers who pirate than delinquent firms: firms are easier to monitor and greater penalties can be exacted from them if they are caught pirating.

If software manufacturers could price discriminate between business and home consumers and enforcement was not prohibitively costly, then it would be optimal to invest significant resources in order to prevent piracy. Price discrimination in the software market is difficult, however, because it is difficult for the most part to segregate pools of potential buyers. Tolerated, limited piracy can be seen as an alternative method of discriminating between pools of buyers.

One can view tolerated piracy as a type of price discrimination in which the manufacturer sells some of the software at a price of zero. It may be optimal for a software manufacturer to encourage limited piracy, because network externalities are created by both legal and illegal consumption of software. Software piracy acts to increase the size of the network, as well as increasing the availability of after-market products and services. Perhaps most important, a pirate's knowledge of how to use a software product can later be applied to the legal software market. These externalities can be interpreted as allowing manufacturers to earn a positive marginal revenue from a product being sold at a price of zero.

3 There is clear evidence that the software manufacturers understand the importance of these knowledge network externalities. In particular, Lotus engaged in a protracted and unsuccessful legal battle in an attempt to prevent Borland from copying the Lotus 1-2-3 user interface (see, e.g., Thomson 1996). It would be difficult to argue that this user interface has any value to either company other than through the large number of people who have invested time in learning it.

There are two countervailing forces acting on the profitability of software when piracy expands or contracts. When piracy increases, the number of paying customers is reduced, decreasing the profitability of the software manufacturer. Owing to the positive network externality, however, an increase in piracy also increases the value of the product to others. If it is possible to increase the number of people who are pirating by a large margin while decreasing the number of paying customers only slightly, an increase in piracy may increase the software manufacturer's profits. Hence, if network externalities are sufficiently great, it might be profit maximizing for the software manufacturer to tolerate limited piracy by home consumers in order to increase the demand for software by business consumers.

4. The model

A single monopolist software manufacturer has two distinct markets for its product: home consumers, whom we index by δ, and business consumers, whom we index by α. The software manufacturer's marginal cost of production is assumed to be zero. The total measure of potential consumers is normalized to one, and it is assumed that fraction μ are potential business consumers, and fraction $(1-\mu)$ are potential home consumers.

Home and business consumers differ in two respects. Both the network externalities and the costs of piracy for business consumers are greater than they are for home consumers. This fact is modelled simply by assuming that network externalities exist only for business consumers and that software manufacturers can costlessly impose piracy costs on business consumers but must expend resources to impose piracy costs on home consumers.

It is assumed that the value business consumer α derives from the software when there are $N = N_H + N_B$ total users of the software is given by

$$V_B(N, \alpha) = \alpha + \theta N, \tag{1}$$

where $\theta \in (0, 1)$ represents the network externality. In turn, the value derived from the software by home consumer δ is given by

$$V_H(N, \delta) = \delta. \tag{2}$$

For simplicity, both home and business direct valuations are assumed to be distributed uniformly on the interval $[0, 1]$. This implies that market demand functions for the two groups are linear.

The software manufacturer can impose an expected cost of piracy $Z_H \geqq 0$ on home consumers who choose to use the software without purchasing it. The cost to the software manufacturer of imposing a piracy cost Z_H on the home consumer is assumed to be independent of the number of units sold and described by the linear enforcement cost function $C(Z_H) = cZ_H$. Further, it is assumed that the software manufacturer can costlessly impose an expected piracy cost of Z_B on business consumers. Given the valuation and cost structure, the software manufacturer will maximize profits in all cases by setting the cost of piracy to business consumers above the equilibrium price. This implies that businesses will always choose to purchase software rather than to pirate.

The key features of the parameterization are that both network externalities and piracy costs are greater for business consumers. The qualitative result now detailed – if network externalities are sufficiently great, then software manufacturers will wink at piracy by certain groups of consumers – holds in less parameterized formulations of the economy. The network externalities make it profitable for piracy to be used as a form of price discrimination in which the manufacturer receives a price of zero from consumers in one market. Specific parametric assumptions are made only to obtain closed-form solutions and to ease comparative static analyses.

To simplify exposition, we divide the firm's optimization problem into two cases based on the relationship in equilibrium between the cost of piracy and the price. In the first case, the software manufacturer finds it worthwhile to set $Z_H^* < P^*$, so that consumers find it less expensive to pirate the software than to purchase it. In the second case, the software manufacturer prevents piracy entirely by setting $Z_H^* \geqq P^*$.

4.1. Piracy

Given that $Z_H^* < P^*$, a home consumer δ will pirate if and only if its valuation exceeds the expected cost of piracy. The marginal buccaneer is given by

$$\delta^* = Z_H \equiv Z, \tag{3}$$

so that the total measure of pirates is

$$N_H = \int_{\delta^*}^{1} (1-\mu)dH = (1-\mu)(1-Z). \tag{4}$$

Since the expected cost of piracy for business consumers is prohibitive, any business consumer wanting the software will purchase. A business consumer α purchases the software if and only if $V_B(N, \alpha)$ exceeds the price, P^*. The marginal business purchaser, α^*, is given by

$$\alpha^*(P) = \max\{0, P - \theta(N_H + N_B)\}, \tag{5}$$

so that the measure of legal consumers is

$$N_B = \int_{\alpha^*}^{1} \mu dB = \mu(1 - \alpha^*(P)). \tag{6}$$

Substituting for N_H and N_B and solving for $\alpha^*(P)$ yields

$$\alpha^*(P) = \max\left\{0, \frac{P - \theta(1 - Z + \mu Z)}{1 - \theta\mu}\right\}. \tag{7}$$

The software manufacturer's profit in the piracy case is then given by

$$\int_{\alpha^*}^{1} P\mu dB - C(Z) = \begin{cases} P\mu\left(1 - \dfrac{P - \theta(1 - Z + \mu Z)}{1 - \theta\mu}\right) - cZ & \\ & \text{if } P > \theta(1 - Z + \mu Z) \\ P\mu - cZ & \text{otherwise.} \end{cases} \tag{8}$$

It is intuitively clear that, since we have ruled out any home consumers paying for the software, the software manufacturer optimally sets $Z^* = 0$ in this case, both to minimize enforcement costs and to increase the effects of the network externality. This is confirmed by the Kuhn-Tucker conditions for the profit-maximization problem with respect to Z. Further, we can see that it will never be optimal to set a price such that $P < \theta(1 - Z + \mu Z)$, since in this case the manufacturer can increase the price without losing any customers and thereby guarantee a larger profit. Using these results and the first-order condition with respect to price, we solve for the profit-maximizing price when piracy is feasible:

$$P_P^* = \begin{cases} \dfrac{1}{2}(1 - \theta\mu + \theta) & \text{if } \dfrac{1}{2}(1 - \theta\mu + \theta) > \theta \\ \theta & \text{otherwise} \end{cases}, \tag{9}$$

which simplifies to $P_P^* = \max\{\theta, (1 + \theta(1 - \mu))/2\}$.

Substituting the profit-maximizing price and piracy cost into the profit function yields maximized profits:

$$\pi_P^* = \begin{cases} \dfrac{\mu(1 + \theta(1 - \mu))^2}{4(1 - \theta\mu)} & \text{if } \theta < \dfrac{1}{1 + \mu} \\ \theta\mu & \text{otherwise} \end{cases}. \tag{10}$$

The second case corresponds to the situation where the manufacturer sets prices to a level that induces all potential business consumers to purchase the software. Inherently, this case is less interesting than the case where some business consumers choose to purchase and some do not. We suppose for the remainder of the paper that network externalities are not so large, that is, we assume that $\theta < 1/(1 + \mu)$. The qualitative results of the paper extend to the case where network externalities are larger, so that allowing piracy is even more profitable.[4]

4 To see why the main result holds even when all business consumers choose to purchase, consider a value of $\theta \geqq 1/(1 + \mu)$. At this level, increases in the network externality do not increase the quantity demanded by paying customers, but they increase the price that the manufacturer can obtain. Conversely, since home consumers are not affected by the externality, their valuations are unchanged. Revenues are maximized by charging a price equal to θ earning revenues $\theta\mu$ by tolerating piracy, rather than attempting to prevent piracy and charging a price less than one to induce a few home consumers to purchase.

4.2. No piracy

Now examine the case where the expected cost of piracy for home consumers exceeds the equilibrium price. The marginal home consumer, who now purchases software, is given by $\delta^*(P) = P$, so that the measure of home consumers who purchase the software equals

$$N_H = \int_{\delta^*}^{1} (1-\mu)dH = (1-\mu)(1-P). \tag{11}$$

The marginal business purchaser is determined just as it is in the no piracy case:

$$\alpha^*(P) = \frac{P - \theta(1 - P + \mu P)}{1 - \theta\mu}. \tag{12}$$

Total profits now equal

$$\int_{\alpha^*}^{1} P\mu dB + \int_{\delta^*}^{1} P(1-\mu)dH - C(Z) = \frac{P(1-P)}{1-\theta\mu} - cZ. \tag{13}$$

The profit-maximizing price is obtained by maximizing total profits with respect to P and Z, subject to the constraint that $Z \geqq P$. It is easy to show that the constraint will bind – the software manufacturer will avoid extra costs by setting Z at the lowest level that will prevent piracy. The profit-maximizing price, P^*_{NP}, is given by

$$P^*_{NP} = \frac{1}{2}(1 - c + \theta\mu c). \tag{14}$$

In turn, maximized profits equal

$$\pi^*_{NP} = \frac{(1 - c + \theta\mu c)^2}{4(1-\theta\mu)}. \tag{15}$$

5. Analysis

PROPOSITION 1. *There exists a critical level of network externalities, θ^*, such that the firm earns greater profits from permitting home consumers to pirate than from preventing piracy if and only if $\theta > \theta^*$. As a function of the marginal enforcement cost and the fraction of business consumers, θ^* is given by*

$$\theta^*(c,\mu) = \frac{1 - c - \sqrt{\mu}}{\sqrt{\mu}(1 - \mu - \sqrt{\mu}c)}. \tag{16}$$

Proof. Solve

$$\pi^*_P - \pi^*_{NP} = \frac{\mu(1 + \theta(1-\mu))^2}{4(1-\theta\mu)} - \frac{(1 - c + \theta\mu c)^2}{4(1-\theta\mu)} = 0. \tag{17}$$

TABLE 1
Comparative statics

	Value	Partial derivatives θ	c	μ
$\pi_P^* - \pi_{NP}^*$		+	+	+
$P_P^* - P_{NP}^*$	+	+	+	−
$Q_P^* - Q_{NP}^*$	−	−	+	+

NOTES:
θ represents the intensity of the network externality, c is the parameter of the linear enforcement cost function, and μ is the proportion of potential business consumers in the economy. The difference in profits can be calculated using equations (10) and (15). The difference in price can be calculated using equations (9) and (14). The quantity $Q_P = N_B$ is the amount of software used by paying customers in the economy with piracy and $Q_{NP} = N_H + N_B$ is the amount used in the case where piracy is prevented. All calculations reflect those restrictions on parameters discussed in section 4.

Since the derivative of this difference with respect to θ is positive for $\pi_P^* - \pi_{NP}^* \gtreqless 0$, the result follows.[5] □

Hence, if and only if the network externality is strong enough ($\theta > \theta^*$), the manufacturer optimally permits home consumers to pirate the software. Effectively, the manufacturer price discriminates by charging a price of zero to the subset of home consumers with valuations $\delta > Z$.

Comparative statics results are summarized in table 1. The table shows that the attraction to the manufacturer of allowing piracy is increasing in the strength of the network externality, the proportion of the market that is business oriented, and the marginal enforcement cost. As the strength of the network externality, θ, rises, the manufacturer becomes more concerned with the total number of users and is therefore more willing to sacrifice some paying customers in order to obtain a larger user base. Similarly, as the proportion of paying users increases, the profit lost by forgoing the revenue in the pirating group becomes relatively less important than increasing the valuations of non-pirates. Note that when piracy occurs, the equilibrium price is higher and sales (Q) are lower than it is when piracy is unprofitable. The lower sales to home consumers are compensated for by the ability to extract a higher price from business consumers. Observe that increasing the network externality, θ, raises the profitability of piracy compared with that of the economy without piracy because it raises, relatively, how much business consumers are willing to pay. In turn, this relatively increased willingness to pay by

5 To see that this critical value can be reached under the assumption that $\theta < 1/(1+\mu)$, note that with $c < 1$, the root to $\theta^* = 0$ will occur at $\mu < 1$. Then, since θ^* is increasing from zero and $1/(1+\mu) > 1/2$, there are values of the parameters that satisfy the restriction.

business consumers when there is piracy raises the price differential in the two cases, indicating that with piracy the firm's optimal price reflects the fact that it sells solely in the market affected by the externality, whereas if the firm does not tolerate piracy, only a fraction of the buyer valuations are raised by the increase in the externality. Hence, since the increase in θ raises prices relatively more in the piracy economy, non-piracy sales are raised relative to piracy sales. Increasing the proportion of business consumers, μ, also increases the attractiveness of piracy, but it has the opposite effect on the difference in price and sales. For this parameter, the extra advantage to piracy is realized through the higher proportion of available paying customers.

Summarizing the results, the software manufacturer wants to make piracy costless if there are sufficient network externalities. If, instead, network externalities are smaller, the manufacturer wants to increase, if possible, the expected piracy costs, so that home consumers do not pirate. If the software manufacturer cannot raise piracy costs sufficiently high to induce many home consumers to purchase, however, then software manufacturers should focus little effort on reducing piracy in this group.

Further, the greater the proportion of the market that is composed of business consumers, the relatively more profitable it is to permit piracy. That is, allowing piracy tends to be more profitable when the software market consists primarily of business consumers (e.g., the market for Lotus 1-2-3). Conversely, products targeted at home consumers are less likely to find the piracy case more beneficial. These would include home entertainment software. Because the business market is so small in this case, the network externalities generated by piracy in the home market can have little positive benefit to the software manufacturer. Therefore, unless the degree of the externality is very high, software manufacturers would prefer to prevent piracy. In fact, counter to this prediction, the evidence seems to be that there is extensive piracy in home entertainment software. This likely relates to the difficulty, and consequently high cost, of preventing piracy in this market. Software manufacturers in this market *do* expend considerable resources in trying to prevent piracy, through the use of special codes and key disks for example. The model predicts only that software manufacturers would like to prevent piracy if the cost is sufficiently low, not that they will prevent it.

6. Discussion

To take this model as a serious explanation for real world practices in the computer software industry, it is crucial that the assumptions on which it is based be examined. The key assumption that the software manufacturer can impose different costs of piracy on different groups of consumers seems to be borne out in practice. Almost all anti-piracy and enforcement measures are focused on business consumers. For example, many manufacturers insist that larger businesses use licence-counting software in order to ensure that all software in use on a network is licensed. In addition, the costs of being caught in an act of piracy can be very high for businesses with fines of up to one million dollars and up to five years in jail for officers of the company (*Financial Post*, 28 May 1994, S18). According to the Software Publishers Association, threats of raids in combination with increased penalties for piracy have been enough to end illegal use of software in larger corporations.

In contrast, almost no effort is directed at catching individual illegal users of software. Indeed, industry organizations appear largely ambivalent towards very small-scale piracy. This attitude is exemplified by statements from software manufacturers such as Bill Leethan, chief financial officer of Berkeley Systems, who has been quoted as saying that his company does not care about very small-scale illegal use of software and prefers to focus its attention on 'people profiting on our stuff' (*Howe Office Computing*, Sept. 1994, 26(2)). Even if an illegal home consumer were to be caught, the penalty is generally little more than a slap on the wrist. Some recent licensing agreements have even allowed employees of companies that own a piece of software to copy it and freely use it at home. This difference in enforcement expenditures can be partially attributed to better direct returns to enforcement expenditure on the business side of the market. However, this does not negate the fact that the difference does exist and can be exploited by the software manufacturers.

A second key assumption is that the network externality crosses between the different groups of consumers: additional consumption by home consumers increases the value of the product for business consumers. It may seem strange to suggest that the consumption choice of home consumers will influence the choice of a major corporation. A major group of home consumers and, consequently, potential pirates, however, are the employees of purchasing corporations. These employees will be more productive if they can apply knowledge obtained from home use of the software to the business environment. Furthermore, employees who bring work home will need to use software which can read the same file format. These facts provide incentives for business users to purchase software that is widely used in the home market.

In discussing the utility of the model, it is useful for us to look at specific cases where network externalities and piracy have jointly played an important role in the software market. Two interesting cases centre around very successful software products, each of which helped to create new uses of computers. Lotus 1-2-3 has often been credited as the application that sparked the initial growth of the personal computer. This spreadsheet was crucial in bringing the computer to the business desktop, since it allowed users without programming experience to perform financial and scientific modelling. Tasks ranging from the basic home budget to complex scientific analyses could be performed in a simple spreadsheet format.

Gandal (1994) shows that the network externalities are an important factor in the market for computer spreadsheets. In particular, network externalities played a large role in making Lotus 1-2-3 the dominant player in its software market. Its file format quickly became a standard format for interchanging numeric data, not only for spreadsheets but also for databases, accounting, and statistical programs. The ability to read and write data in this format gave 1-2-3 an important role as the centre of a communications network. As the program became more popular, a large library of add-on products was developed by other companies, further enhancing the spreadsheet's value. Most important, learning how to make the best use of 1-2-3 required significant time investments by consumers. Not only was time needed to learn the user interface, but this program required people to develop a new way of thinking about data analysis. The more people who learned and passed on these skills, the more valuable was 1-2-3.

Lotus 1-2-3 was one of the most pirated software products of its day. In the context of our analysis, it is easy to see how this piracy may have raised Lotus's profits: having a large base of legal or illegal home users greatly increased the value of 1-2-3 to businesses. Rather than spending large amounts of time training employees in the use of proprietary analysis tools, a company could exploit the knowledge its employees had gained on their own time by using Lotus 1-2-3.

Additional insights can be gained from alternative characterizations of the two groups. Another possible grouping of consumers of software is *early adopters* and *late adopters*. Early adopters, largely home consumers, are less affected by network externalities than are late adopters, who correspond to business consumers. For a ground-breaking program like Lotus 1-2-3, it was important to build a base of users who could be informal teachers and advocates for the program. Lotus could have price discriminated intertemporally, initially charging less for its product. However, it could also simply tolerate piracy by early adopters in order to build value into its product for late adopters. If this interpretation is accurate, then Lotus should have increased its anti-piracy spending as the product became more successful. In fact, Lotus became one of the strongest proponents of copy-protection technology and expended significant resources trying to prevent software piracy in the later part of the 1980s. In practice, almost all anti-piracy spending is done by well-established companies, suggesting that new entrants have more to gain, or at least less to lose, from piracy.

A second path-breaking software product is the Netscape Navigator web-browsing client. This software has a dominant market share among programs used to access the World Wide Web. There are many ways in which network externalities affect the value of this product. Although the general file format used for web documents is based on a common standard, the hyper-text mark-up language (HTML), Netscape has added many extensions to its browsers that do not strictly conform to the standard. Authors who use these extensions can have added features on their web pages, but these web pages work properly only if the Netscape software is used to view them. The more users of Netscape software there are, the more likely it is that authors will use Netscape-specific extensions, which in turn increases the value of Netscape software. Web browsers can also gain many extended features through the use of *plug-in* programs, which allow users to view file formats other than HTML. These plug-ins often work for only one brand of software, and since Netscape is the most popular web browser, it has the biggest selection of plug-ins. Finally, like the other software discussed, web browsers are very complex and require an investment of time to master all their features. This investment is not easily transferred to browsers made by different companies.

The owners of Netscape, in effect, have staked their company on the ability of these network externalities to increase the value of their product. To maximize these externalities, the company is overtly supporting piracy of their software by making fully functional evaluation copies available free of charge on their website. In this way, Netscape seeks to build a large Internet user base that will translate into a valuable product to sell in the more lucrative *intranet* market. The *intranet* market consists of internal corporate networks that use the same network protocols and software as the Internet. Because Netscape dominates the Internet (home) market, the extensive supply of plug-ins and the wide user knowledge about the product make it an obvious choice for intranet (business) customers. In this way, piracy in the Internet market can increase overall profits.

A related phenomenon that our model can explain is the practice of bundling low-priced computer software with new computer hardware. It has become a common practice for certain software manufacturers to distribute some of their software through original equipment manufacturers (OEMs) at prices substantially below those charged in regular retail channels. The price difference between the direct and the OEM market can sometimes be so great as to preclude simple price discrimination; some software distributed in the OEM market is free. Our model can reconcile this practice. There are two well-defined, and easily separated groups: new computer users who buy hardware and software through OEMs; and after-market software consumers who already own hardware and wish to buy only software.

Substantial differences in willingness to pay between the two groups are easily identifiable. Purchasers of bundled systems are most often new and inexperienced computer users. As such, their willingness to pay for any particular type and brand of software could be much lower than an after-market consumer who has a much better understanding of the capabilities and limitations of software products.

Interpreting bundled-system consumers as home consumers and the after-market consumers as business consumers, our piracy analysis suggests that if the network externality is sufficiently strong, a software manufacturing monopolist may find it optimal to give away software in the bundled market in order to benefit from the network externalities in the after market.[6]

References

Church, J., and N. Gandal (1992) 'Network effects, software provision and standardization,' *Journal of Industrial Economics* 40, 85–104

David, P. (1985) 'Clio and the economics of QWERTY,' *American Economic Review Papers and Proceedings* 75, 1145–51

6 A similar explanation can be provided for Apple's practice of offering low-priced or free computers to public schools.

Gandal, Neil (1994) 'Hedonic price indexes for spreadsheets and an empirical test of the network externalities hypothesis,' *Rand Journal of Economics* 25, 160–70
— (1995) 'Competing compatibility standards and network externalities in the P.C. software market,' *Review of Economics and Statistics* 77, 599–608
Givon, Moshe, Vijay Mahajan, and Eitan Muller (1995) 'Software piracy: estimation of lost sales and the impact on software diffusion,' *Journal of Marketing* 59, 29–37
Katz, Michael, and Carl Shapiro (1985) 'Network externalities, competition and compatibility,' *American Economic Review* 75, 424–40
Parker, Rachel (1991) 'Why should anyone buy Lotus 1-2-3 or Dbase anymore?' *InfoWorld*, 18 February
Shurmer, Mark (1993) 'An investigation into sources of network externalities in the packaged P.C. software market,' *Information Economics and Policy* 5, 231–51
Thomson, William E. (1996) 'Searching for a solution to the user interface dilemma,' *Managing Intellectual Property*, April