Proprietary Software, Free and Open-Source Software, and Piracy: An Economic Analysis

A theoretical approach to competition between free and non-free software in the presence of unauthorised copying and network externalities

Arne Rogde Gramstad

Thesis for the degree
Master of Economic Theory and Econometrics

Department of Economics
University of Oslo

May 2012

Proprietary Software, Free and Open-Source Software, and Piracy: An Economic Analysis

2012

Proprietary Software, Free and Open-Source Software, and Piracy: An Economic Analysis

Arne Rogde Gramstad

http://www.duo.uio.no/

Trykk: Reprosentralen, Universitetet i Oslo

Summary

This thesis aims to analyse the impact of software piracy on competition between a non-free proprietary type of software and a free/open-source type of software.

In pursuing this, I use a model inspired by Besley et al. (2010) originally applied to describe voting behaviour in political elections. In the benchmark model with no piracy there are two types of software: one free (e.g. open-source) and one non-free type (i.e. proprietary). I show that under certain conditions the proprietary software type may strategically take advantage of network externalities by reducing the price in order to prevent users from choosing the free type of software. In this way the proprietary software developer may avoid that the free software type generates sufficient network externalities in order to create high demand for the free software type. However, such a strategy may involve a large price reduction. Therefore, the profit maximising strategy may rather be to set the price higher so that both types of software generate sufficient network externalities to exist side by side.

When users have the option of obtaining an unauthorised copy of the proprietary type of software (i.e. piracy), the optimal pricing scheme may change relative to the no-piracy benchmark. I find that when piracy is present, it is more often optimal to keep the free type of software out of competition by strategically taking advantage of network externalities. This is because the threat from piracy may force a price reduction which also affects the demand of free software. In addition, piracy takes market share directly from the free type of software. Hence, market dominance of the proprietary type of software arise more easily when piracy is present.

Furthermore, I provide empirical evidence that suggests that Linux (i.e. free and open-source software) usage is negatively affected by the extent of software piracy.

The main conclusions of this thesis are that piracy affects demand for free/open-source software negatively, and that piracy may contribute to market dominance by the non-free proprietary software type when network externalities are present. This is because piracy mitigates the competitive advantage of free software (the price) in competition with non-free software. In addition, the pricing strategy towards competition from free software may change when piracy is present. Despite the market dominance that may occur from piracy, the model gives no implications that piracy may increase profits.

Preface

This thesis marks the ending of the five year program in economic theory and econometrics at the University of Oslo. The writing and research process has been fun, difficult and exhausting, but all in all a valuable experience.

My supervisor, Tore Nilssen, deserves special thanks for his enthusiasm, useful comments and suggestions.

I am also grateful to my parents for support and for their effort of proof-reading the thesis.

In the spirit of the topic of this thesis, it is worth mentioning that free and open-source software has been widely used working with this thesis: The document and some figures are made using LibreOffice, scatter plots and regressions are calculated using Gretl, and some minor image editing was done using GIMP.

Pirated software has not been used writing this thesis.

Table of contents

List of tables

List of figures

Abbreviations

BSA – Business Software Alliance

BSD – Berkeley Software Distribution

EULA – End-user license agreement

GDP (PPP) – Gross domestic product (GDP) at purchasing power parity (PPP) exchange rates

GPL – GNU General Public License[*]

OLS – Ordinary least squares

OS – Operating system

OSS – Open-source software

* GNU – GNU's Not Unix (recursive acronym)

1. Introduction

Free software has grown to hold a considerable market share in several markets for software. However, a large share of this freely available software is unauthorised copies of proprietary software and was never intended to be free. With the rise of broadband internet, effective file sharing technologies and lenient enforcement on copyright infringement, piracy represents a major influence in various software markets. At the same time, open-source software and other types of free software have become an influential force, but has received mixed success in various segments of the software market.

Certain types of free and open-source software have become dominant players in among others the markets for web servers (Apache) and mobile phones (Android). For other types of software, such as office suits software (LibreOffice/OpenOffice.org) and software for academic purposes (e.g. the econometrics package Gretl), free and open-source software is influential, but is by no means dominant. Finally, there are some fairly well-known types of free software, but at the same time with rather limited success in sense of market share, such as various types of Linux operating systems for desktop computers, like Ubuntu.

In this thesis I study the interaction between non-free proprietary software, free software and piracy. Specifically I examine how competition between free and non-free proprietary software is affected by unauthorised copying of proprietary software in the presence of network externalities. Furthermore, I look at piracy's impact on prices and firm profits when a proprietary software vendor faces competition from both unauthorised copying and free/open-source software.

In pursuing this, I use a model from political economics inspired by Besley et al. (2010), originally applied to explain voting behaviour in elections. User preferences depend among other things on product quality and size of installed user base as a measure of network externalities. This work shows that when network externalities are strong and the non-free proprietary type of software has a competitive advantage from a relatively larger constant installed user base and higher product quality, it may be a feasible option for the developer of the non-free software type to maintain low prices in order to keep out competition from the free/open-source type of software. However, as such a strategy may involve a large price reduction, this may not necessarily be the optimal pricing scheme in terms of profits.

Because piracy takes user shares from both user segments of the software market, and that price reduction may be an effective method of fighting piracy, the free/open-source type of software may be driven out of competition more easily when piracy is present. This follows from that in the presence of network externalities, it is more likely that strategic pricing in order to avoid individuals using free/open-source software is the optimal pricing scheme with the existence of piracy relative to the no-piracy case.

My findings suggest that, in the presence of strong network externalities, software piracy can be destructive as it reduces competition and thus leaves consumers with little variety of products to choose from. Also, profits of the proprietary software vendor is likely to be negatively affected by piracy, even if piracy leads to market dominance.

In order to avoid confusion it should be stressed that free software and open-source software are not synonymous. Proprietary (closed-source) software may be free of charge, and open-source software can in theory be sold. Moreover, open-source software is often described as being *free* both in the sense of “free of charge” and “having freedom” due to the few legal restrictions in open-source software licensing. In this thesis the “free of charge” interpretation of the word is used. Hence, “free software” may refer to both open-source software and free proprietary software, although open-source software may be more applicable to the competitive environment described in this thesis.

The rest of the thesis is organised as follows: Section 2 presents some facts and characteristics of the software market in general and free software and piracy specifically. Section 3 reviews relevant literature regarding the topic of this thesis. The model is presented in section 4. Section 5 discusses the model's findings and reflects on possible limitations and extensions. Empirical evidence is provided in section 6. Section 7 concludes.

2. Markets for non-free proprietary software, free software and piracy

I this section I briefly present some characteristics of the software market, as well as present some facts on the extent of software piracy and free and open-source software. I follow up with a brief discussion in light of the facts presented.

2.1. Supply, demand and other characteristics

Software is only one of many categories of goods defined as information goods. By using the definition of Shapiro and Varian (1999), information is anything that can be digitised, that is encoded as a stream of bits. Thus books, web pages, music, pictures, films, and of course software, are all examples of information goods. Moreover, regarding both the demand and supply side of markets, information goods may differ substantially from physical goods.

On the supply side, the cost structure is the most obvious difference. The production of an information good is characterised by a constant, and, for most practical purposes, zero marginal cost. Also, the production of an information good usually requires a very small amount of physical capital. In many cases only a computer is needed, meaning that anyone with the necessary skills and a computer can produce an information good. Elementary microeconomic theory predicts that in a competitive environment prices will converge to its marginal cost in production for a given quantity. When the marginal cost is zero, the price will in many cases in fact be zero. On the other hand, information goods are usually differentiated to some degree. An mp3 file with a Justin Bieber song is distinctly different from an mp3 file with Mozart, and Windows is not at all the same as a Linux operating system (OS). Property right holders of differentiated goods may thus have some market power. A constant marginal cost and a positive fixed cost characterises another well-known phenomenon from microeconomic theory called economies of scale which usually leads to natural monopolies. As we know, monopolies seldom give away their products for free, but rather sell at a painfully high price in the eyes of the consumer. We thus have two strong forces pushing in different directions: the marginal cost that pushes prices to zero and the monopolies that want all of the economic surplus for themselves.

On the demand side, other interesting features occur. There seems to be an abundance of possibly free goods available (including pirated goods, which I will come to) which leads us down to a place on the demand curve seldom observed for goods of economic interest. When the price is zero, a budget constraint will not explain a lot when it comes to demand, and as the price of a given hard disk space is cut in half every 18 months or so ("Moore's law"),[1] constraints in form of gigabytes are diminishing. This could lead us to believe that individuals will consume almost an infinite amount of free information, and that competition would push the price of all information goods to its marginal cost. Obviously, this is not the case. Especially for software, which is the main focus in this thesis, there may be properties on the demand side making the markets not so competitive after all.

First of all, when it comes to software, many goods are substitutes. That is that your consumption of good 1 decreases your demand for good 2 since most of the services provided by good 2 already are covered from your consumption of good 1. For example if you already use Microsoft Office, your demand for OpenOffice.org/Libreoffice or Google Docs will certainly be lower (or zero).

Second, many types of software are experience goods: consumers do not know their valuation of a product before they have experienced it. This may lead to substantial informational asymmetries between consumers and software producers. Moreover, consumers may be biased towards the software they already know or have experienced. On the other hand, fast internet connection and effective search engines contribute to easier access to independent reviews and even targeted advertising that help remove the informational barriers between consumers and producers.

Third, switching from one type of software to another may be costly. Someone who uses software 1 and considers changing to software 2, must take into account the learning costs of figuring out how to use the operating environment of software 2. Furthermore, there may be issues regarding changing file formats and possibly even costs of changing complementary applications. Also, as mentioned above, there might be informational barriers. If these "switching costs" are sufficiently high, it may be unthinkable to switch software at almost any price, and we end up in what is called *consumer lock-in*. When consumers are locked in, software vendors are in a strong position as they can price their product higher. On the other hand, the opportunity of potentially locked-in customers in the future may lead firms to lower the price of the software in order to attract more buyers, knowing that a new customer is likely to be a life-long customer. Moreover, the trade-off between charging high prices in order to extract profits from locked in consumers and charging low prices in order to attract new consumers easily leads to various price discrimination strategies such as student discounts, etc. (Varian et al., 2004).

1 "Moore's Law to roll on for another decade" CNET News, 10 February, 2003, http://news.cnet.com/2100-1001-984051.html , retrieved 3 February, 2012.

Finally, many types of software provide *network externalities*. Network externalities occur when demand for a given good depends positively on how many others are using that good. For that reason, network externalities generate what often is called “demand side economies of scale”. As explained by Economides (1996), network externalities appear from complementarity between components of a network. Therefore, network industries share many economic features with non-network industries that are characterised by strong complementary relations. Typically a distinction is made between *direct* and *indirect* network externalities. *Direct network externalities* occur when consumers are directly identified with the components of the network. However, when an increase in network size yields increased demand for complementary goods and thus potentially increase the variety of complementary goods to the network good in question, we have *indirect network externalities*.

Network externalities may occur for a number of reasons: Communication technology such as the telephone or e-mail are prime examples of goods generating direct network externalities as these technologies are useless unless others use this technology as well. Compatibility issues caused by e.g. industry standards are another factor leading to network externalities. In the markets for software, operating systems are goods generating strong indirect network externalities as third party application developers build their applications in order to ensure compatibility with the most popular operating systems (i.e. the industry standards). Compatible applications to a given OS are thus complementary goods which increase the value of the OS. The lack of compatible applications on the other hand will likely decrease the demand of the OS, which in turn will give weaker incentives for other developers in producing compatible applications. Another factor contributing to network externalities may be word-of-mouth effects as it is e.g. easier to find solutions to problems that may occur by using a piece of software in online fora if many others use this specific piece of software.[2] Hence, demand for a piece of software where network externalities are strong is easily found in a virtuous or vicious cycle, depending on whether the installed base of users exceeds a critical mass necessary for other users to demand a piece of software (Shapiro and Varian, 1999). Network externalities tend to result in one dominant player within each market. However, there is often room for more than one platform. For instance Windows have since the early 1990s been the dominant operating system for desktop computers after they defeated Apple in becoming the standard OS, and for almost 20 years Apple was only an alternative for very few enthusiasts as popular applications commonly used on Windows did not work on an Apple computer. In recent years, Apple

2 I.e. when the software you are using is used by many others, the likelihood that someone has encountered the same problem as you increases.

computers with their own operating system, OS X, have caught on, and OS X now holds a significant share of the OS market for desktop computers (approximately 6% worldwide and as much as 14% in North America).[3] As a result, most popular applications (web browsers, office software, games, etc.) are compatible with both Windows and OS X since both operating systems hold a large share of the market. At the same time, various free of charge open-source Linux distributions such as Ubuntu, Fedora, Debian and Linux Mint have improved much the recent years both regarding quality and usability.[4] Especially in the last 5-10 years, developments in open-source desktop environments like GNOME and KDE have made Linux easier to use for non-technical users. Although these Linux distributions are completely free of charge, the global desktop OS market share for Linux remains slightly above 1%.[5] As a result, only the most important applications commonly used in Windows, such as web browsers, are compatible with Linux OS,[6] while most computer games and applications developed by Microsoft and Apple are generally not compatible with Linux.[7]

A market where network externalities have become much more important in recent years is that of operating systems for mobile phones. As third party application compatibility is an important determinant for usability of a phone with a given OS, demand for smart phones depends a lot on the number of applications that can run on the phone. Before the introduction of smart phones, there was a swarm of different mobile phone operating systems, as each mobile phone manufacturer bundled their own OS with their phones. Now the smart phone OS market is dominated by IOS by Apple, the Linux-based Android, developed by Google and the Android open-source project, and to some extent Windows Phone by Microsoft, while e.g. the market share of Symbian by Nokia never seemed to exceed a critical mass in order for application developers to make Symbian compatible applications. Nokia's recent switch to Windows Phone can thus be seen as a response to the lack of network externalities generated by the Symbian OS.

Moreover, it may seem that network externalities and consumer lock-in are intertwined in certain segments in the markets for software. The switching cost of going from Windows to OS X is probably lower than switching from Windows to Linux. Most of the popular Windows applications, such as MS Office, also work on an Apple computer. Hence the largest cost (excluding the rather high price of a Mac), relates to learning how to operate in the OS X

3 Net Market Share: http://marketshare.hitslink.com/, retrieved 30 January, 2012.
4 A distribution is a bundle of software already compiled and configured. A Linux distribution is an operating system built on a Linux kernel including a collection of a software applications, desktop environment, etc.
5 Net Market Share: http://marketshare.hitslink.com/, desktop OS user share of Linux 1.4%, retrieved 21 January, 2012.
6 Those are Google Chrome, Firefox and Opera. I.e. there is no Linux version of Internet Explorer.
7 Wine, an application for Linux, runs Windows applications, however Windows applications run in Wine seldom work perfectly.

environment. Going from Windows to Linux on the other hand means giving up MS Office, your favourite computer games and possibly facing various hardware issues such as connecting an Ipod, as well as learning how to use a number of new applications. Switching from Linux to Windows on the other hand, is less costly. After all, most relevant Linux applications also exist in Windows compatible versions.

2.2. Proprietary software, free software and open-source software

Proprietary software is software licensed under exclusive legal right of the copyright holder, and the user of a proprietary piece of software is granted use under certain conditions. Typically, the user must accept an end-user license agreement (EULA), a contract between user and publisher, in order for an application to be installed on a hard-drive. By accepting the EULA, the user agrees not to e.g. modify the software, derive the source code, by-pass protection mechanisms, redistribute the software, in addition to various other restrictions.

Proprietary software includes *free* proprietary software and *non-free* proprietary software.[8] *Free proprietary software* comes in various shapes. Some types of software are given away for free as a strategy of generating revenue around the product, usually by selling complementary goods in some form (giving away razors in order to sell razor blades). For instance Apple gives away their media player ITunes for free as a strategy of selling more media content on ITunes Store. Other types of free software may work as a promotional strategy for other products, or may contribute in building company loyalty. For instance Google gives away various products that yield no direct revenue, but as these products increase in popularity, chances are that people might use the Google search engine more (i.e. Google's main source of revenue).[9] A third business model involving giving away something for free is based on versioning of products by giving away a basic product for free with the option of a premium version for a fee, popularised under the term *freemium* by Anderson (2009). Included in freemium models are "try & buy" with a limited time trial of a product, advertising on the basic version where the premium version removes advertising, and a premium version with more or upgraded features relative to the free basic version. The freemium model attempts to mitigate information problems as well as possibly taking advantage of consumer lock-in by giving away the basic version for free. On the other hand, if competition is fierce, the basic version cannot be

8 Unless otherwise stated, proprietary software is equivalent to non-free proprietary software in this thesis.
9 With the new Google privacy policy, implemented on March 1st, 2012, information gathered from an individual's use of Google's free applications may also be used to improve the accuracy of Google's targeted advertising services, which in turn may increase Google's advertising revenues.

too basic, or else consumers will choose the competitor's free basic product with less limitations. Also, if the basic version is too feature limited, people might not try the piece of software at all. Hence, the software developers may be forced to give away a fairly high quality product. Because of these mechanisms, in many cases the basic version is used by the vast majority in markets with many competitors using the freemium model, as the basic version often differs little in quality from the premium version. The basic version can be regarded as a promotional sample for the premium version, but as giving away these "free samples" is costless, the developer can generate revenues even if only a small share choose to upgrade to premium as long as total user share is large. Anderson (2009) refers to this as the "5 percent rule", that the 5% of premium users subsidise the 95% using the basic version.[10] Examples of popular products using the freemium model are the Winamp media player in which the premium version includes among other thing the possibility of CD burning, CD ripping as well as mp3 encoding, and various anti-virus programs such as Avira and AVG where the premium versions include more security features in addition to the virus protection already included in the basic versions. Common for these examples are that most of the additional features in the premium versions in fact are available for free in other types of software. The fact that some users still "go pro" suggests either that they prefer an "all-in-one" solution or that they are sufficiently locked-in so that the price of the premium version is lower than the searching costs and learning costs of the free alternatives.

Open-source software (OSS) is software that usually is given away for free under a license, most commonly under "copyleft" licensing such as the GNU General Public License (GPL), which gives "legal permission to copy, distribute and/or modify the software."[11] Moreover, copyleft licensing requires that the copyright within the license is maintained in extended and modified versions, ensuring that the modified work will e.g. not be converted into proprietary software. Another type of license used in open-source software is the Berkeley Software Distribution (BSD) license which have even less restrictions.[12] For software under the BSD license, extended and modified versions can be converted into proprietary software, such as Google's web browser Chrome which is built on the BSD licensed web browser, Chromium.

The development of OSS happens through a collaboration of programmers working for free following a project plan, and in some cases also in collaboration with professional

10 Anderson (2009), pp. 26-27.
11 "GNU General Public License" GNU Operating system: http://www.gnu.org/licenses/gpl-2.0.html, retrieved 20 April, 2012.
12 "BSD License information", The Linux Information Project: http://www.linfo.org/bsdlicense.html, retrieved 20 April, 2012.

programmers.[13] Typically, bugs and suggestions for improvements of the software are posted by users, and members of the community are free to contribute by fixing bugs or working on improvements by accessing the source code. OSS has grown to be an important influence, and, is in some market segments the dominant type of software. Sourceforge, the largest online host for OSS, hosts nearly 300,000 software projects and reported more than 500 million downloads in the second quarter of 2011.[14] The perhaps most successful open-source projects measured in total users are the web browsers Mozilla Firefox and Google Chrome,[15] which combined are estimated to have more users than Microsoft's Internet Explorer.[16]

Apart from the formal differences between open-source and proprietary software such as price, development and legal aspects, there might be significant qualitative differences between the two types of software as well. Focusing on software primarily relevant for academic purposes such as statistics, econometrics and numerical analysis, Yalta and Yalta (2010) survey literature regarding the accuracy of econometrics software as well as applying various accuracy tests on relevant closed and open-source software packages. They argue that commercial software vendors may introduce various difficulties in the research process by not correcting known errors as well as by not giving access to the details of algorithms. Thus it becomes difficult, if not impossible to study and verify the programming code. In relation to this, complications may occur in replicating and verifying previous research. However, their study finds several defects in the accuracy of the open-source Gretl econometrics package. On the other hand, as the scholars were able to access the Gretl source code, they were able to find the exact cause of the defects, and the errors were corrected within a week of reporting. Likewise, they tested four widely used proprietary econometrics packages for flaws reported more than five years earlier, and found that only two of the software companies had corrected the errors.

It has also been claimed that open-source software may be more secure than closed source software. Since the code is open, individuals are able to discover and report security breaches, so that these flaws can be fixed rapidly. On the other hand, for the same reason individuals with bad intentions may exploit security breaches discovered by inspecting the code.

From conventional economic theory, the success of OSS development is difficult to explain. Why are so many people willing to sacrifice so much time and effort in developing a product with no direct monetary pay-off? As in other markets, altruism cannot explain major

13 For instance are companies such as Google, Oracle and Canonical involved in various open source projects.
14 http://sourceforge.net/blog/sourceforge-working-to-make-our-site-better-every-day/ retrieved 23 January 2012
15 Strictly speaking Google Chrome is not open-source. However, Google Chrome is built on the source code of the open-source browser Chromium.
16 Statcounter: Global stats. Measured user share: IE 35.8%, Chrome 29.8%, Firefox 24.9%: http://gs.statcounter.com/, retrieved 1 March 2012.

contributions such as those observed in open-source projects. The discussion by Lerner and Tirole (2002) suggests that reputational benefits from successful contributions such as peer recognition and attention may be important factors in explaining the degree of contributions. Non-reputational personal benefits may also apply in the case of fixing or reporting bugs which in turn may lead to higher quality for all users. In addition to this, source code contributions may work as training or investment in human capital for potential and actual professional programmers which may increase their value as labour. Moreover, it is not unthinkable that some degree of contribution may be regarded as a pleasurable activity. Based on a web survey administered to 684 software developers, Lakhani and Wolf (2005) report that enjoyment-based motivation is the strongest and most persuasive driver of source code contribution.

2.3. Piracy

Unauthorised copying of software and other information goods is almost as old as the personal computer itself, but with the introduction of broadband internet and cheap sources of storage such as CD-R and DVD-R enabling cheap and easy reproduction, piracy has become a serious threat to the software industry as well as to the music, television and film industry.

In the early days of personal computing, piracy could involve a group purchasing and sharing a piece of software by themselves.[17] However, copy protection of varying degrees of sophistication was soon introduced by the software developers in order to mitigate the degree of illegal copying. Soon various cracker groups emerged with the intention to "crack" the copy protection and release unauthorised copies of copyrighted works, so-called *warez* ("leetspeak" plural form of *ware*, short for software), which allowed the continuation of illegal copying, even under fairly advanced forms of copy protection.

With the rise of widespread use of internet and the increasing capacity of hard disk space, pirated products have been increasingly more easily available to less advanced computer users. In addition, this has allowed the warez community to grow as well as to have increased cooperation and efficiency in order to crack the increasingly more sophisticated copy protections made by the software developers. With the inventions of peer-to-peer (p2p) networks and broadband internet, the warez scene is no longer dependent on large computer servers in order to spread their pirated products, and thus it has been increasingly difficult for governments, software developers and other anti-piracy agents to crack down on piracy. In addition, these innovations have made it possible for less advanced computer users to obtain pirated products at

17 Varian (2000) provides a model with this type of information good sharing.

a low transaction cost.

The Business Software Alliance (BSA) estimates in the 2010 Global Piracy Study the global software piracy rate to be 42%, which ammounts to a commercial value of 58,754 millions USD if sold instead.[18] Obviously, the demand for those products is not infinitely inelastic, so the losses to the software industry from piracy is believed to be much lower, but still substantial. Furthermore, the BSA report estimates piracy rates to be much larger in emerging markets than in developed economies. For instance, the piracy rates in Georgia and China, are estimated to be 93% and 78%, respectively. United States, Japan, and Luxembourg are on the other hand estimated to have the lowest piracy rates in the world with 20%, which still accounts for a large share of the proprietary software market.[19] Figure 2.1 illustrates the rather strong negative correlation between piracy rates and GDP per capita.

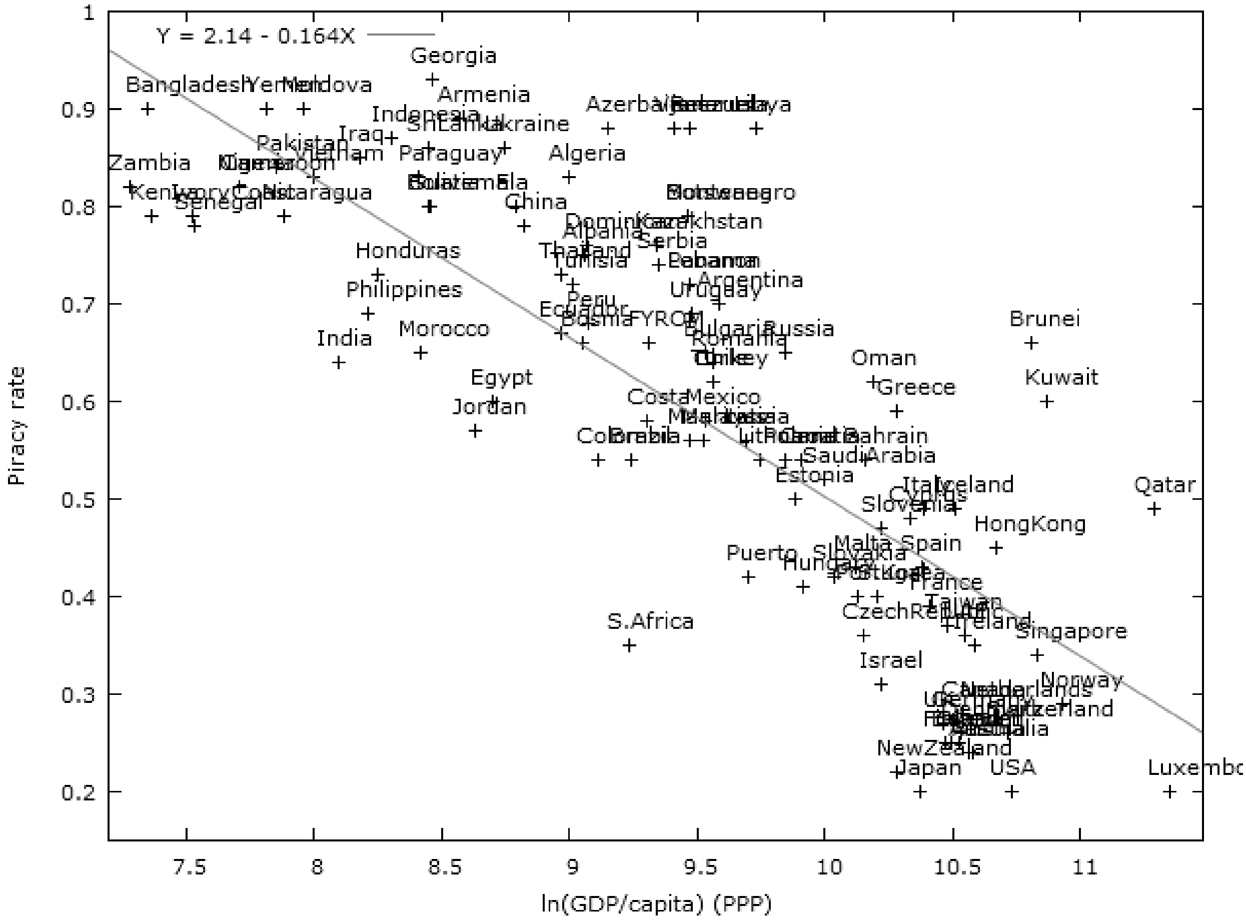

Figure 2.1: Scatter plot with OLS regression line between piracy rates and ln(GDP/capita) (PPP) in USD for 2010. Sources: Own calculations based on data from 2010 BSA Piracy Study and The World Bank.

However, the BSA piracy study has been criticised for its methodology. Among other things, Png

18 Business Software Alliance (2011), p. 9.

19 Business Software Alliance (2011), p. 3.

(2010) argues that the BSA statistics should be used with caution in government policy and academic studies, and questions among other things the estimates from many developing countries since national income has been used as a measure of piracy rates in the cases where software usage was not directly measured.

2.4. Non-free proprietary software vs. free software and piracy

The fact that piracy is more widespread in developing than developed countries should come as no surprise. First of all, the price of a piece of software as a fraction of income is much larger for relatively poor individuals, so the cost of purchasing a legitimate product relative to the cost of obtaining an unauthorised copy of a piece of software is probably much larger for the median Chinese than a Norwegian. Also, developing countries on average have less advanced institutions and are generally more lenient on enforcing copyright infringement. As a result, pirated products of all sorts can be purchased at very low prices from street vendors in many developing countries, and are in many cases more easily obtainable than the legitimate products.

On the other hand, with all the legitimate *free* software available one would expect these to be relatively more popular in developing countries, but generally they are not. Webmasterpro.de, a German site offering web programming services, managed to estimate user share of office suits between 28 countries with its online statistics tool Flash Counter by checking which fonts were installed on the systems registered by visitors on sites using their statistics service.[20] Their estimates suggest rather large differences in the free OpenOffice.org office suits between countries,[21] and, as figure 2.2 shows, there is in fact a rather strong positive correlation when comparing the numbers to GDP of the countries in the sample.

As the publisher of the statistics points out, there is a fairly large margin of error in the estimates. Also, the Flash Counter tool is primarily used on German sites, suggesting that the estimates may suffer somewhat from self-selection bias of “germanophile” netizens.

20 Statistics available on http://www.webmasterpro.de/portal/news/2010/02/05/international-openoffice-market-shares.html, retrieved 24 January, 2012.

21 The informed reader might wonder why LibreOffice is not included in the statistics. This is because the data is from 2010, and the initial release of LibreOffice came in 2011.

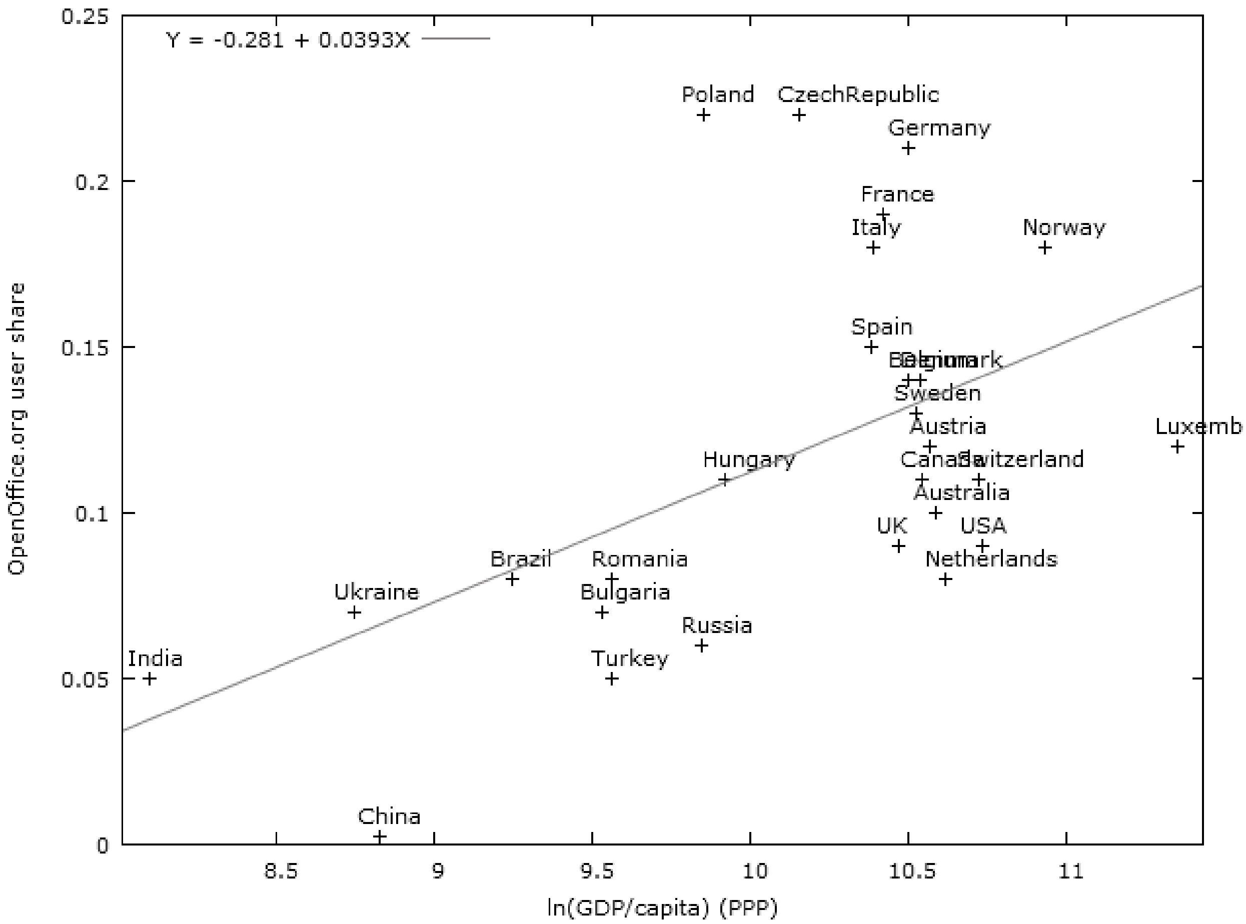

Figure 2.2: Scatter plot with OLS regression line between OpenOffice.org user share and ln(GDP/capita) (PPP) in USD for 2010. Sources: Own calculations based on data from http://www.webmasterpro.de/portal/news/2010/02/05/international-openoffice-market-shares.html and The World Bank.

Despite the possible large measurement errors in both OpenOffice.org user share and piracy rates between countries, and that OpenOffice.org is only one of many types of free software, there seems to be established some relationship between usage of OpenOffice.org and piracy (see figure 2.3). The relatively larger use of piracy in developing countries is not unexpected, but that individuals living in rich countries are more likely to choose a free office suit than those in poor countries seems counter-intuitive. This observation suggests that demand for free software in fact may be affected by the availability of piracy. Taking network externalities into account, this negative effect on demand for free software may even be amplified.

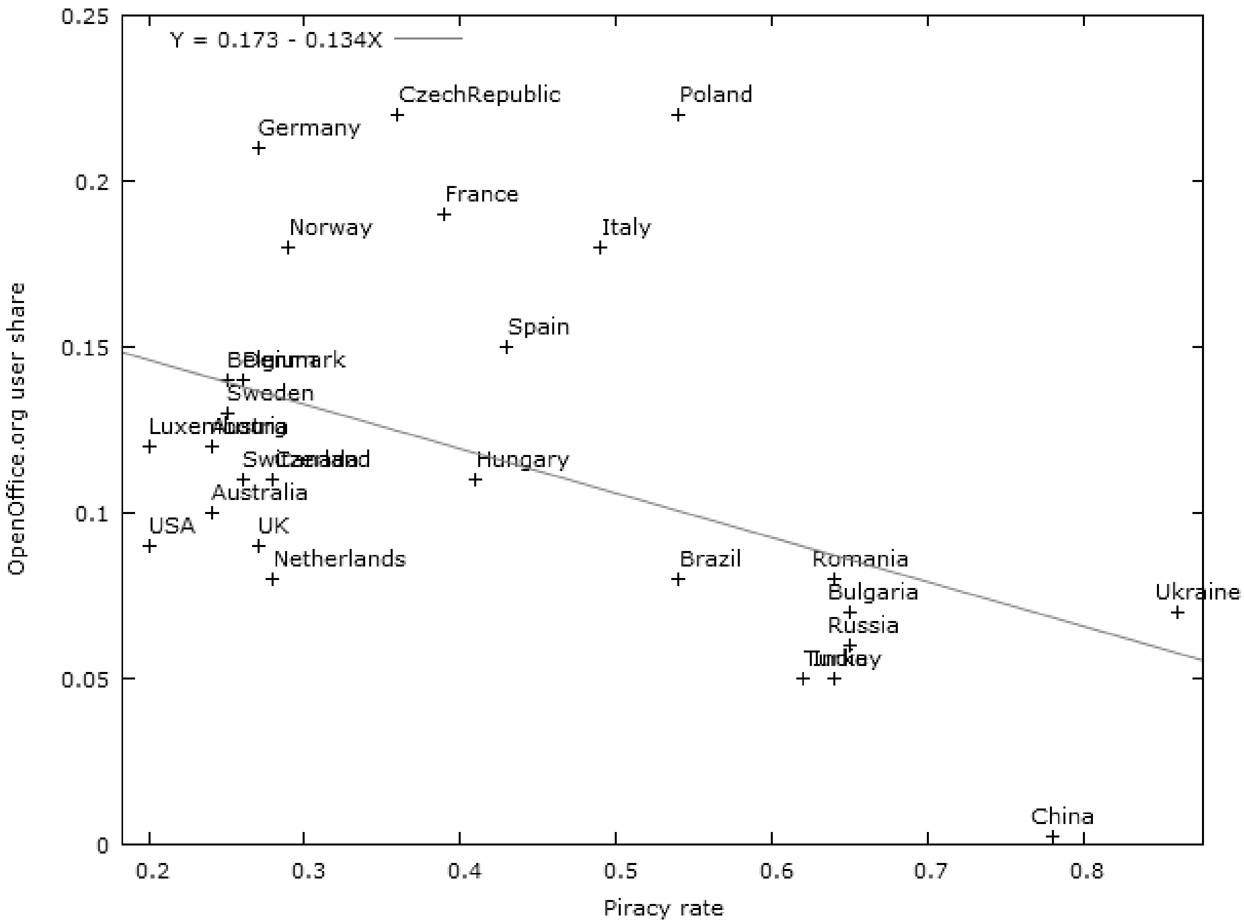

Figure 2.3: Scatter plot with OLS regression line between OpenOffice.org user share and piracy rate for 2010. Sources: Own calculations based on data from http://www.webmasterpro.de/portal/news/2010/02/05/international-openoffice-market-shares.html and 2010 BSA Piracy Study.

In some cases, free/open-source software is the main competitor of market leading non-free proprietary software. For office suits, OpenOffice.org and LibreOffice hold a significant user share, although still far behind MS Office. Among users of advanced graphics editing, Adobe Creative Suite (CS) (which includes Photoshop) is believed to be the undisputed market leader while for instance the free open-source alternative GIMP, although having registered more than 117 millions downloads on Sourceforge (which does not imply that 117 million people actually use GIMP), lies far behind Adobe CS in market share.[22] Considering the estimates by the 2010 BSA Piracy study of a global software piracy rate of 42% and that Adobe CS5 retails for 699 USD (199 USD for upgrade), it is likely that the pirates exceed more than half of the total Adobe CS users. The fact that Adobe CS is the most shared and downloaded pirated application on The

22 Adobe does not reveal sales numbers of their product. However, a quick Google search reveals that “Adobe Photoshop” has almost four times as many hits as “GIMP”. A vast number of the “GIMP” hits are non-software related, suggesting that Adobe Photoshop by far beats GIMP in popularity.

Pirate Bay supports this.[23]

Although a significant share of the pirates are potential buyers, one has to ask what the software developers of these products would prefer: that a large market share goes to pirated versions of their software or that they choose a (legal) free alternative? Officially the developers of proprietary software take a hard line against piracy. On Microsoft's web pages it is stated that "software piracy equals lost wages, lost jobs, and unfair competition. Struggling to fight against piracy, some companies must devote resources to anti-piracy technology, ultimately slowing down the development of better products and services."[24] Microsoft has a valid point, but effects may go in more than one direction. To a group of students at Washington University in 1998 former Microsoft CEO Bill Gates was quoted as saying: "Although 3 million computers get sold each year in China, people don't pay for our software. Someday they will, though, and as long as they're going to steal it, we want them to steal ours. They'll get sort of addicted, and then we'll somehow figure out how to collect sometime in the next decade."[25] A similar argument was given by Microsoft executive Jeff Raikes to the business technology magazine Information Week in 2007 where he pointed out the importance of anti-piracy efforts, but that the approach must be balanced by the recognition that users of pirated software one day might become legitimate users.[26] It seems that Microsoft has recognised how piracy, at least in part, may contribute to maintaining their dominant market position, as well as to avoiding use the competitors' software (including free software). In addition, piracy may contribute to network externalities of their products which indirectly cause *lack of* network externalities to the competitors.

The open-source community recognises how piracy may hurt demand for free software as well. To Computer World in 2008, community manager for the OpenOffice.org project at former Sun Microsystems (later acquired by Oracle), Luis Suarez-Potts, argues how piracy is hurting the open-source community as unauthorised copying of proprietary software "represents a lost opportunity for open-source software makers to get their own software onto the computer hard drives of new users." He further argues that this lost opportunity, in addition to the reduced influx of new users, hurts the development of open-source software as additional users also are potentially active participants in the open-source community.[27] In a comment in (the now

23 Top 100 Applications, The Pirate Bay: http://thepiratebay.se/top/300, retrieved 22 March, 2012.
24 "Protect yourself from piracy", Microsoft Corporation: http://www.microsoft.com/piracy/reporting/default.aspx retrieved January 25, 2012.
25 Anderson (2009), p. 102.
26 "If You're Going To Steal Software, Steal From Us: Microsoft Exec", Information Week, 12 March, 2007: http://www.informationweek.com/news/198000211 retrieved 25 January, 2012.
27 "Software piracy hurts the open-source community, too", Computer World, 23 July, 2008: http://www.computerworld.com/s/article/9110560/Software_piracy_hurts_the_open_source_community_too_ retrieved 25 January, 2012.

defunct) Tux Magazine in 2007, a magazine aimed at Linux users, Jon Knight argues similarly that easy access to *warez* creates a bias towards the established major software suits as people tend to use the piece of software used by "everyone else". As piracy removes the cost factor of proprietary software, piracy also mitigates the biggest advantage of free and open-source software, namely the price.[28]

2.5. Wrapping up

In the preceding sections, I have mentioned some typical characteristics in the markets for information goods and software in particular. Zero marginal cost, consumer lock-in, and network externalities may have great influence on prices and market power. Furthermore, many segments in the software market are affected by two types of free software: unauthorised copies of proprietary software (piracy), and free and open-source software.

In the rest of the thesis I will try to explain how competition between a free and non-free type of software is affected by piracy in the presence of network externalities in a static environment. Specifically I aim to find out how prices, profits and market shares between a free and non-free software vendor are affected by piracy, and whether network externalities are important for the results.

What this thesis will *not* explicitly answer is how such an environment may be affected by consumer lock-in and the possibility of pirates turning into legitimate users of proprietary software. Although these are important questions, the effects of users of pirated software getting "sort of addicted"[29] and end up purchasing require dynamic modelling which is not covered by the model presented in this thesis. However, some light is shed on these issues in section 5.

28 "Why Piracy Hurts Open Source", Tux Magazine, Issue 4, 2007: http://www.tuxmagazine.com/node/1000266 retrieved 25 January, 2012.
29 In the words of Bill Gates. Anderson (2009), p. 102.

3. Related literature

This thesis is related to at least two research themes: (1) software piracy and its effect on the market outcome, and (2) competition between free and non-free software. With regard to the second theme, most relevant literature addresses more specifically competition between proprietary/commercial software and open-source software. However, in most cases, unless code contribution is explicitly relevant for the model, a passive competitor providing a free good is equivalent to open-source.

The piracy related literature focuses mainly on how piracy affects firm profits, welfare and prices. Uncontroversially, piracy will reduce firm profits in a static setting where a monopoly supplies the good (Jain, 2008; Bae and Choi, 2006). However, long run effects are more ambiguous as lower profits due to piracy mitigate the incentives of investment, which may lead to degradation in quality relative to the case where piracy is not feasible (Bae and Choi, 2006). On the other hand, it is possible that copiers end up purchasing the product later on, increasing firm profits (Takeyama, 2003). Furthermore, the impact of piracy on the market may differ substantially when firms face competition from other software vendors in the presence on piracy and/or network externalities.

Jain (2008) presents a model in which firms freely are able to choose copyright protection of their software in an environment with two types of consumers: one who never copies and one who might copy. In the monopoly case the firm's optimal strategy is to set perfect copyright protection as piracy will cause lost sales. When competition is introduced in the duopoly case, the optimal strategy for both firms is to set a low copy protection as this reduces competition for price sensitive “might copy consumers” and allows the firms to credibly coordinate by setting high prices and extract most of the profits from the non-copiers. In the case of strong network externalities, the model predicts an asymmetric equilibrium where one firm imposes weak copyright protection and the other firm strong copyright protection. The intuition behind this result is that the firm allowing piracy can increase its price among non-copiers as the product is regarded more valuable when user share is large. The optimal strategy for the other firm given the first firm's strategy might be to enforce strong copyright protection in order to sell their product to both “might copy consumers” and non-copiers who have strong preferences for the software supplied by the second firm. The analysis by Jain (2008) suggests that the optimal piracy rate might be positive. Although there exists copy protection of various degrees of sophistication, it may be unreasonable to assume that firms freely can choose the piracy rate of their product.

Bae and Choi (2006) show that a monopoly will reduce the price if the threat from piracy is strong. The price of the original product, search costs of obtaining an unauthorised copy, and the quality of the copy relative to the original product shape the incentives for users. Thus the easier it is to obtain a copy and the more equal the copy and original are in terms of quality, the more the monopoly has to reduce the price in order to induce users to purchase rather than copy.

Takeyama (1994) focuses on welfare effects and firm profits from piracy in the presence of network externalities. She argues that unauthorised copying of intellectual property is an efficient way of increasing the network size of a product. Piracy may contribute to a preemptive installed base of users, which in turn may increase demand of the software by potential buyers. Hence, if certain conditions hold, piracy in combination of network externalities may increase both firm profits and social welfare.

With regard to the literature concerning free versus non-free competition in the software market, Bitzer (2004) focuses on the role of product heterogeneity in strategic competition between a proprietary and an open-source software developer. He argues that the proprietary software is able to maintain a strong market position as long as sufficient differentiation between the products is ensured. The proprietary software developer can for instance provide complementary applications for its software to boost demand if this application is incompatible with the software of the open-source competitor.

A similar argument is provided by Economides and Katsamakas (2006). They focus on pricing strategies for software platforms such as operating systems and the interaction between the software platform and the demand for complementary third-party proprietary applications. They find that subsidising the complementary proprietary applications may be an optimal strategy for the proprietary software platform. Furthermore, they find that this strategy is more likely if applications are complementary to each other, and less likely if substitutability between applications are strong. In relation to this, they argue that Microsoft's dominance in e.g. markets for office suites will increase demand for Microsoft's OS, which in turn may decrease profitability of third-party proprietary application providers in selling its products on an open-source platform. In this manner Microsoft can maintain its market domination even if the switching costs to Linux are zero.

Lanzi (2009) models an environment with competition between closed and open-source software with perfect software compatibility. In contrast to other literature, Lanzi (2009) allows differences in skills among users. He argues that in a competitive environment, the proprietary software house will reduce its price relative to the monopoly benchmark if the network of proprietary software users is larger than the open-source network *and* its users are largely skilled in the program, *or* if the network is small and largely unskilled. On the other hand, a large and unskilled network will increase the price relative to monopoly benchmark. The reason for this result is that it is more costly for unskilled users to switch to open-source assuming opportunity costs of learning and understanding open-source is high for the unskilled. However, when this opportunity cost is low, an open-source dominance solution will emerge. Lanzi's (2009) observation is consistent with the open-source dominance in e.g. markets for web servers and supercomputers. The nature of open-source allows users to manipulate the software in order to fit the special needs of managing complex systems, but this versatility of open-source software may possibly be a disadvantage for the less advanced users.

The paper most closely related to the topic of my thesis is that of Casadesus-Masanell and Ghemawat (2006) as the it deals with a duopoly between proprietary and open-source software as well as extending the model to include piracy. By using dynamic modelling they find that the proprietary software vendor can exploit network externalities strategically in order to maintain its dominant market position by reducing the price of its product when the proprietary software is regarded more valuable than the open-source software at $t = 0$, and that the rate of decay (discount rate) of past sales is not too large. They argue that Linux's failure to dominate the desktop OS market is not due to switching or searching costs, nor is it related to demand side coordination issues, but rather to Microsoft's strategic actions by not pricing Windows too high in order to avoid demand side learning on behalf of Linux and thus maintaining a high user installed base which increases the value of Windows. In the welfare analysis, they find that is it possible that welfare is higher under Windows monopoly than under a duopoly. The reasoning lies in the effects from network externalities. Network externalities increase demand (shifts demand curve out) which means that the area under the demand curve is larger if only one type of software is available rather than if the market is shared between two (or more) software platforms. However, the fact that those who do not purchase Windows are not left empty-handed, as they can download Linux for free, as well as price of Windows is set lower in a duopoly, argues that the economic surplus increases with competition. In addition, the effect from increased network size on demand is likely to have a diminishing marginal return (Katz and Shapiro, 1985), which suggests that the difference in values of the network sizes between a monopoly case and duopoly case is small.

Casadesus-Masanell and Ghemawat (2006) then extend the model to include piracy of the proprietary software by assuming an exogenously given share of former proprietary software users and open-source users turning into pirates. The main finding is that the larger the piracy rate is, the larger is the steady-state difference in installed bases. Since the value of the two types of software correlates positively with the size of the installed base, Casadesus-Masanell and Ghemawat (2006) argue that piracy of proprietary software makes it harder for open-source software to remain competitive. They finally provide some empirical evidence by regressing Linux/Windows ratio based on shipments of Linux and Microsoft server OS on piracy rates from the 2001 BSA Study and GDP per capita. They find that piracy has a significant negative effect on Linux/Windows ratio whereas the coefficient of GDP is insignificant. Their findings and methodology are discussed further in section 6.

To the best of my knowledge, there has not been conducted any research on the effects of an endogenously determined rate of piracy in a duopoly with one provider of non-free software and one provider of free/open-source software. Moreover, the type of modelling in this thesis differs from those in the related literature. I present a model inspired by political economics where I reinterpret voters' party affiliations as preferences for types of software, and where winning an election becomes equivalent to obtaining a critical mass in installed base as an interpretation of network externalities. I believe this type of modelling can provide new insights in understanding the mechanisms of this segment of the software market.

4. The model

I present a model similar to the political economy model of Besley et al. (2010). In their paper they present a simple model explaining how election outcomes depend on the share of "partisans" and the preferences of "independent voters". Democratic institutions have the property that only those who vote on the winner of an election (meaning at least 50% of the votes) will, at least ideally, enjoy the policy of their preferred party, while the minority voters are "forced upon" an unpreferred (or not optimal) policy.

I present an alternative interpretation of Besley et al.'s model where I attempt to explain how consumers' choices may be affected when network externalities are present in the software market. Just as a minority socialist voter may be forced to "consume" conservative policy in a democracy, someone who would like to use e.g. a Linux operative system (OS) may feel forced to use Windows if the total user share of the Linux OS is below some critical mass due to lack of network externalities. This mechanism can in turn have serious implications for competition in the software market.

First I model a benchmark case with competition between a non-free proprietary type of software and a free type of software. I further discuss how the market outcome and pricing of the proprietary software may look like in this setting. Second I add piracy to the model and show how the results in the piracy case may differ from those in the no-piracy case.

4.1. The benchmark model: Competition without piracy

We consider an economy with population size normalised to one. There are two different types of software which do a similar task (for instance operating system or office software). Hence, the two goods are substitutes and consumers only use one type of software. We denote the two goods *W* and *L* (for Windows and Linux respectively). The price of *W* is $p \in [0, \bar{p}]$, and *L* is free.

The model describes two types of software located on opposite ends of a Hotelling line. However, opposed to classic duopoly models, *W*, the non-free proprietary software type, is the only active player in this "game" and can to some extent affect demand for both types of goods through price setting mechanisms (i.e. there is no best response function for the vendor of the free good). The optimal (profit maximising) price of the *W* type of software may differ depending on the importance of network externalities. It turns out that in certain cases the price can be set strategically so that *L*, the free type of software, is prevented from generating sufficient network externalities, which in turn leads to an increase in demand for *W*, the non-free proprietary type of software. Importantly, although such a pricing strategy is feasible, it might not be optimal. In many cases, the vendor of the *W* good is better off in an environment where both software types exist side by side.

4.1.1. Demand

There are three types of consumers, "*W* enthusiasts", "*L* enthusiasts" and "uncommitted users" formally defined in a set $C \in \{E_W, E_L, U\}$.

The *enthusiasts* make up a fraction $1 - \sigma$ of the economy. They always choose to consume either *W* or *L* regardless of quality and price $p \leq \bar{p}$ (so *p* cannot be set to infinity). They can be regarded as e.g. geeks or programmers (which may be a fitting description for some Linux users), or they can be regarded as someone who always chooses the default option (e.g. never switches from the software that is bundled with the computer).[30] Among the enthusiasts a fraction $(1 + \lambda)/2$ prefers *W*, meaning $(1 - \lambda)/2$ prefers *L*,[31] where $\lambda \in [-1,1]$. If $\lambda > 0$ (< 0) the larger share among the enthusiasts use software *W* (software *L*).

The *uncommitted users* make up a fraction σ of the economy. They choose which type of software to consume based on quality/valuation v_i ($i \in W, L$), price *p* as well as personal taste, ω, and a shock variable, η. Positive (negative) values of ω and η represent preferences in favour of good *W* (good *L*). Hence an uncommitted user will prefer to consume *W* whenever:

$$\eta + \omega + (v_W - p) - v_L > 0,$$

and *L* whenever:

$$\eta + \omega + (v_W - p) - v_L < 0.$$

Even though an uncommitted user may *prefer* one type of software to the other based on quality, price and personal taste, he will only choose to consume it if a sufficiently large share of the other individuals of the economy chooses to consume it as well. The critical value of the market share necessary for an independent to consume the type of software he prefers is denoted by *N*, where $N \in [0, ½]$. *N* is interpreted as the importance of network externalities in the market we are studying.[32] If $N = 0$, there are no network externalities present, and all independents consume

30 So some of the "enthusiasts" are not really literally *enthusiasts*, but passive users who always choose the software that is presented to them.
31 $1 - (1 + \lambda)/2 = (1 - \lambda)/2$
32 In the paper of Besley et al. (2010) $N = ½$ (critical value in order to win an election).

the type of software they prefer. If $N > 0$ network externalities are more important. In many markets for software the user share of a type of software is crucial for demand. For instance in the case of operating systems (OS) the market share may have a strong impact on what types of applications and hardware that are made compatible to the OS (which is the most important property of an OS). In the case of office software, network size may be important in the sense that two or more individuals can read and collaborate on the same document, and it would be an advantage if all individuals use the same type of software. Also, the file formats of one office suite are usually not perfectly compatible with other office suits. For instance, the OpenDocument format of OpenOffice.org/LibreOffice[33] is not perfectly compatible with Microsoft Office, just as the Office Open XML format from Microsoft is not perfectly compatible with OpenOffice.org/LibreOffice. Hence, the larger market share of a type of software, the less difficulties will individuals face from using this software, and demand is likely to increase in market share. I present a simplification of this mechanism by simply introducing a threshold value for market share necessary for a uncommitted user to consider using this type of software.

We assume that ω is uniformly distributed: $\omega \sim U[\frac{-1}{2\varphi}, \frac{1}{2\varphi}]$. The smaller ϕ is, the larger is the degree of heterogeneity in consumers' preferences. By calculating the cumulative density function (cdf) of ω we find the share of independents of the entire economy *preferring W* in absence of network externalities:[34]

$$\begin{array}{ll} 0 & \text{if } (v_W - p) - v_L + \eta < \frac{-1}{2\varphi} \\ \sigma\left(\varphi[(v_W - p) - v_L + \eta] + 1/2\right) & \text{if } \frac{-1}{2\varphi} < (v_W - p) - v_L + \eta < \frac{1}{2\varphi} \\ \sigma & \text{if } (v_W - p) - v_L + \eta > \frac{1}{2\varphi} \end{array}$$

and independents *preferring L:*

$$\begin{array}{ll} \sigma & \text{if } (v_W - p) - v_L + \eta < \frac{-1}{2\varphi} \\ \sigma\left(1/2 - \varphi[(v_W - p) - v_L + \eta]\right) & \text{if } \frac{-1}{2\varphi} < (v_W - p) - v_L + \eta < \frac{1}{2\varphi} \\ 0 & \text{if } (v_W - p) - v_L + \eta > \frac{1}{2\varphi} \end{array}$$

33 After Sun Microsystems was acquired by Oracle in 2010, LibreOffice separated from the OpenOffice.org project as an independent project under The Document Foundation due to fears of discontinuation of further development of OpenOffice.org. As things stand today, the two office suites are essentially the same.

34 cdf of ω is given by: $F(x) = \begin{cases} 0 & \text{if } x < \frac{-1}{2\varphi} \\ \frac{1}{2} + \varphi x & \text{if } \frac{-1}{2\varphi} < x < \frac{1}{2\varphi} \\ 1 & \text{if } x > \frac{1}{2\varphi} \end{cases}, x = v_W - p - v_L + \eta$

In order to keep things interesting we assume an interior solution, i.e. that $\frac{-1}{2\varphi} < (v_W - p) - v_L + \eta < \frac{1}{2\varphi}$ holds for any profit maximising value of $p \in [0, \bar{p}]$. This means that we assume that among the uncommitted consumers, there are some that prefer *W* to *L* and vice versa for the profit maximising price. If it did not hold, that could e.g suggest that the quality of the *L* good was so bad (low v_W) that no uncommitted consumers would choose to consume it anyway. Unless the share of *W* enthusiasts is large and/or $\bar{p}$ is very high it is not unreasonable to assume that the seller of the *W* good would set a price so that there is some degree of preference for the *W* good among the uncommitted consumers.

We define demand in absence of network externalities (alternatively if total user share always exceeds the critical value, *N*) as W_{-N} and L_{-N}, respectively. They are given by:

$$W_{-N}(v_W, v_L, p, \varphi, \eta) = \sigma\big(\varphi[(v_W - p) - v_L + \eta] + 1/2\big) + (1-\sigma)(1+\lambda)/2 \tag{1}$$

and

$$L_{-N}(v_W, v_L, p, \varphi, \eta) = \sigma\big(1/2 - \varphi[(v_W - p) - v_L + \eta]\big) + (1-\sigma)(1-\lambda)/2. \tag{2}$$

The first element in (1) and (2) represents the share of uncommitted users using *W* and *L*, respectively. The second element is the share of enthusiasts for *W* and *L*, respectively.

Demand for goods *W* and *L* in the presence of network externalities is defined as follows:

$$W(v_W, v_L, p, \varphi, \eta, N)$$
$$L(v_W, v_L, p, \varphi, \eta, N)$$

Total demand of *W* and *L* are then given by:

$$W = \begin{cases} \sigma\big(\varphi[(v_W - p) - v_L + \eta] + 1/2\big) + (1-\sigma)(1+\lambda)/2 & \text{if } W_{-N} > N \text{ and } L_{-N} > N \\ \sigma \min\big(1, \varphi[(v_W - p) + \eta] + 1/2\big) + (1-\sigma)(1+\lambda)/2 & \text{if } W_{-N} > N \text{ and } L_{-N} < N \\ (1-\sigma)(1+\lambda)/2 & \text{if } W_{-N} < N \end{cases} \tag{3}$$

$$L = \begin{cases} \sigma\big(1/2 - \varphi[(v_W - p) - v_L + \eta]\big) + (1-\sigma)(1-\lambda)/2 & \text{if } L_{-N} > N \text{ and } W_{-N} > N \\ \sigma + (1-\sigma)(1-\lambda)/2 & \text{if } L_{-N} > N \text{ and } W_{-N} < N \\ (1-\sigma)(1-\lambda)/2 & \text{if } L_{-N} < N \end{cases} \tag{4}$$

The first line of equation (3) is the demand for *W* when demand given price in the absence of network externalities exceeds *N* for both types of software. Consequently,

$W = W_{-N}$ and $L = L_{-N}$ is true when this is the case. The second line of equation (3) is the demand for W when demand in the absence of network externalities exceeds N for W (i.e. $W_{-N} > N$) and is below N for L (i.e. $L_{-N} < N$). In this case L is considered worthless among uncommitted users. For this reason the quality measure of L, v_L, is omitted from the demand function (i.e. $v_L = 0$), and the difference in quality between the two software types does not enter the demand function. The bottom line of the demand function represents the case when total demand for W in the absence of network externalities is below N (i.e. $W_{-N} < N$), meaning only enthusiasts will use the W type of software, provided that $p \leq \bar{p}$.

Equation (4) follows the same interpretations as equation (3). The difference between the second line of the two demand functions are due to the assumption that no software users will be left empty handed if $L_{-N} > N$, as the L software is free, and using L will yield higher utility than consuming no software at all. Hence, all uncommitted users will use L if $L_{-N} > N$ and $W_{-N} < N$. However, if $W_{-N} > N$ and $L_{-N} < N$, software L is considered worthless among uncommitted users, and they will purchase W as long as the price does not exceed their valuation of W. This is essentially a monopoly for software W among uncommitted users, and since the demand function is falling linearly, the profit maximising price may be so that some users will be left without any software.

4.1.2. Pricing and market share

Now that we have derived demand functions, we can calculate profit maximising behaviour for the developer of software W given the parameters and demand for L.

We consider the development cost of W as sunk cost and zero marginal cost. The assumption of marginal cost equal to zero makes sense as the only cost of reproducing an information good comes from the cost of bits on a hard drive, alternatively on a CD or DVD disk. Today the cost of a gigabyte is negligible, hence the assumption of zero marginal cost will not alter the argument. Considering development costs as a sunk cost makes sense in a static environment, but in a dynamic setting these should be taken into account as profits from software sales would be an decisive factor for further development.

The software developer of W will maximise the following expression:

$$\max_{p} \quad W(p)\,p\,.$$

The first order condition gives the following optimal p, when $L_{-N} > N$:

$$p_{+L} = \frac{v_W - v_L + \eta}{2} + \frac{1}{4\varphi} + \frac{(1-\sigma)(1+\lambda)/2}{2\sigma\varphi} \tag{5}$$

If the share of enthusiasts is sufficiently large and/or $\bar{p}$ sufficiently high, the profit maximising price will be $\bar{p}$ as only selling enthusiasts, and possibly a few uncommitted users,[35] would maximise profits. Generally, when $L_{-N} > N$ the profit maximising price will be:

$$p^*_{+L} = \begin{cases} \frac{v_W - v_L + \eta}{2} + \frac{1}{4\varphi} + \frac{(1-\sigma)(1+\lambda)/2}{2\sigma\varphi} =: p_{+L} & \text{if } \; p_{+L} W(p_{+L}) > \bar{p} W(\bar{p}) \\ \bar{p} & \text{if } \; p_{+L} W(p_{+L}) < \bar{p} W(\bar{p}) \end{cases} \tag{6}$$

If $L_{-N} < N$, so that no uncommitted users will demand L, the profit maximising price will *per se* be higher as the firm developing W does not have to worry about competition from L. On the other hand, since $\frac{\partial L_{-N}}{\partial p} > 0$ a too high price may allow L to enter the market for uncommitted users, which may push the profit maximising price in the opposite direction. Thus keeping the price low in a strategic pricing scheme in order to keep competition out may be the optimal strategy. The profit maximising price in absence of competition among uncommitted users is:

$$p^*_{-L} = \begin{cases} \frac{v_W + \eta}{2} + \frac{1}{4\varphi} + \frac{(1-\sigma)(1+\lambda)/2}{2\sigma\varphi} =: p_{-L} & \text{if } \; L_{-N}(p_{-L}) < N \\ \frac{N - (1-\sigma)(1-\lambda)/2 - \sigma[1/2 - \varphi(v_W - v_L + \eta)]}{\sigma\varphi} & \text{if } \; L_{-N}(p_{-L}) \geq N \end{cases} \tag{7}$$

where the last expression is the break-even price so that $L_{-N} = N$. In this case I have omitted the possibility that $\bar{p}$ is the profit maximising price, as this is unlikely to occur in a situation where the non-free type of software has a monopoly among uncommitted users and still keeping L_{-N} below N.

35 Provided that $\varphi[(v_W - \bar{p}) - v_L + \eta] + 1/2 > 0$.

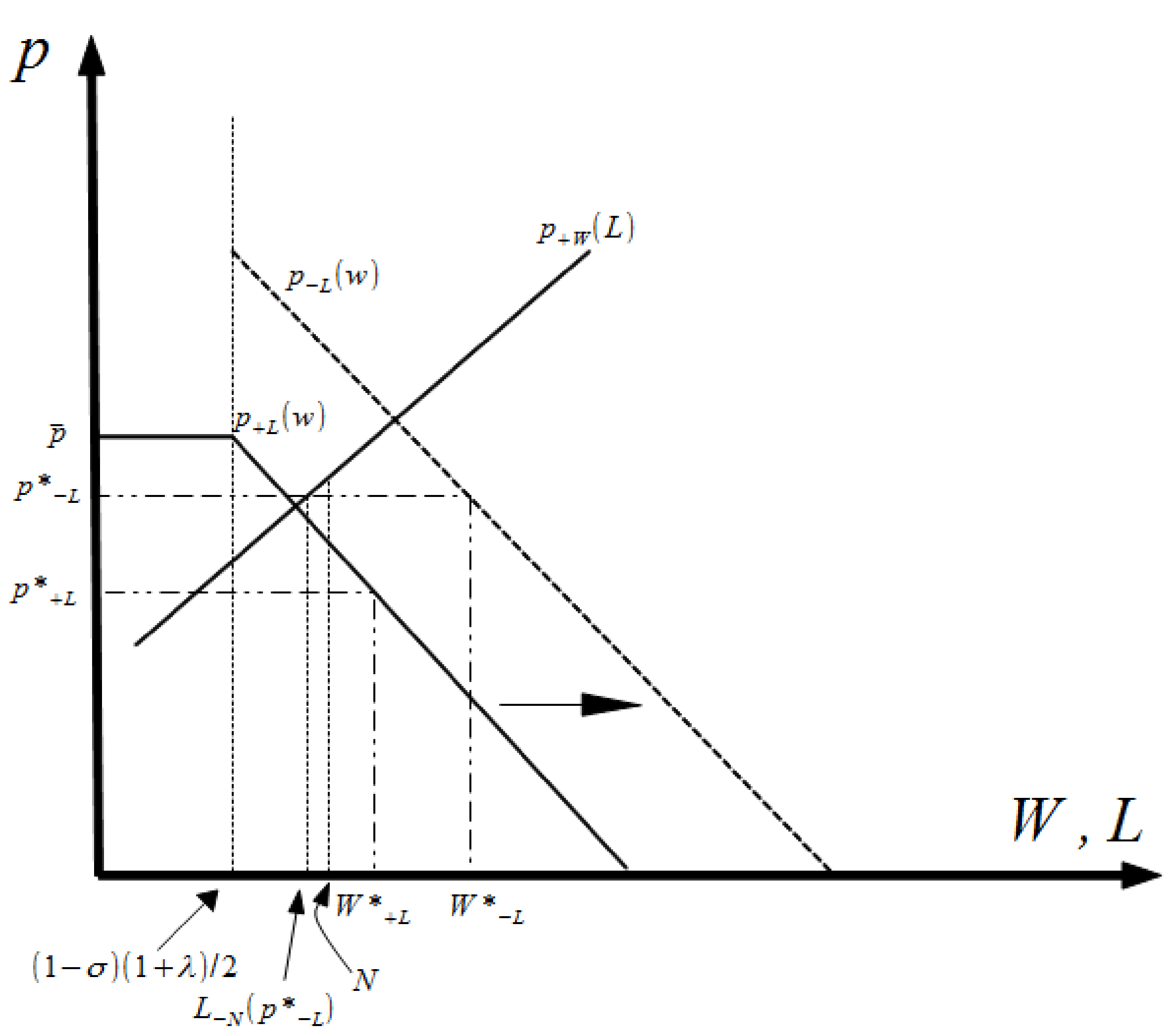

Figure 4.1: Profit maximising prices and quantities sold of W software depending on whether L is in the market for uncommitted users.

If software L exits the market, the demand curve for software W shifts outwards. This will lead to both a higher price and higher quantity sold. That is $p_{-L}(W) > p_{+L}(W)$ for all W. The effect is illustrated in figure 4.1 where p^*_{+L} and p^*_{-L} are set such that the marginal revenue is equal to zero.

It is worthwhile looking closer at the case where lowering the price can keep L out of the market for uncommitted users. This can occur if $v_W - v_L$ is large and/or λ and σ are large as well as N is not too small. If this will increase profits relative to the case where both types of software are chosen among the uncommitted users, this will be the profit maximising price. This special case, illustrated in figure 4.2, shows how the proprietary software vendor can use strategic pricing in order to keep the free software out of the market by taking advantage of network externalities, and is thus a simplified illustration of the effects presented by Casadesus-Masanell and Ghemawat (2006). In cases where N is large, the strategy of setting the price such that L_{-N} is just slightly below N, might be the profit maximising price. In figure 4.2 the price p_{+L} is set so that the marginal revenue is equal to zero. However, by lowering the price until $L_{-N} = N$, the demand for L becomes zero among uncommitted users, resulting in a positive shift in the demand for W. The price is lowered, but the amount of uncommitted W users is increased substantially, and profits increase from $p_{+L}\, W_{+L}$ to $p^*_{-L}\, W^*_{-L}$ as illustrated in figure 4.2.

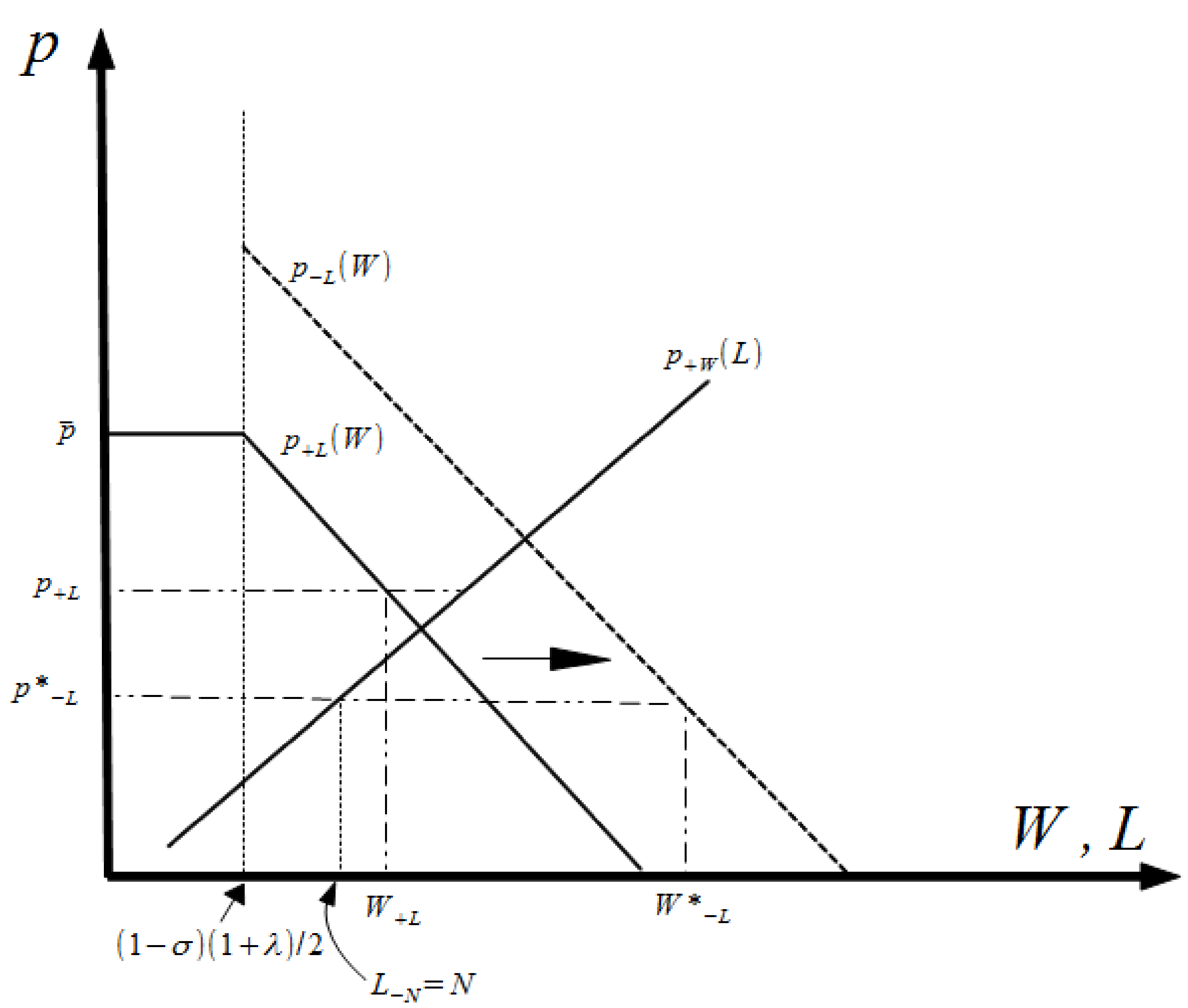

Figure 4.2: Special case when the profit maximising price of W is such that a lower price keeps L out of the market for uncommitted users.

As the reader hopefully has realised by now, the outcome of market share between L and W may have crucially different outcomes depending on whether preferences in the absence network externalities, L_{-N} and W_{-N}, exceeds N, the critical value for uncommitted consumers to use a software or not. Consider the case where $(1 - \sigma)(1 + \lambda)/2 > N$, which means that W will always be a possible choice among uncommitted users since the size of W enthusiasts exceed N. We further assume that there are very few L enthusiasts, and that many uncommitted L users are needed in order for $L_{-N} > N$ to hold. As illustrated in figure 4.3, we might have a case where a fairly large share of uncommitted users would prefer the free L software in the absence of network externalities, but choose not to use the free software due to the difficulties that arise by using a piece of software that is not used by a sufficiently large amount of other individuals.

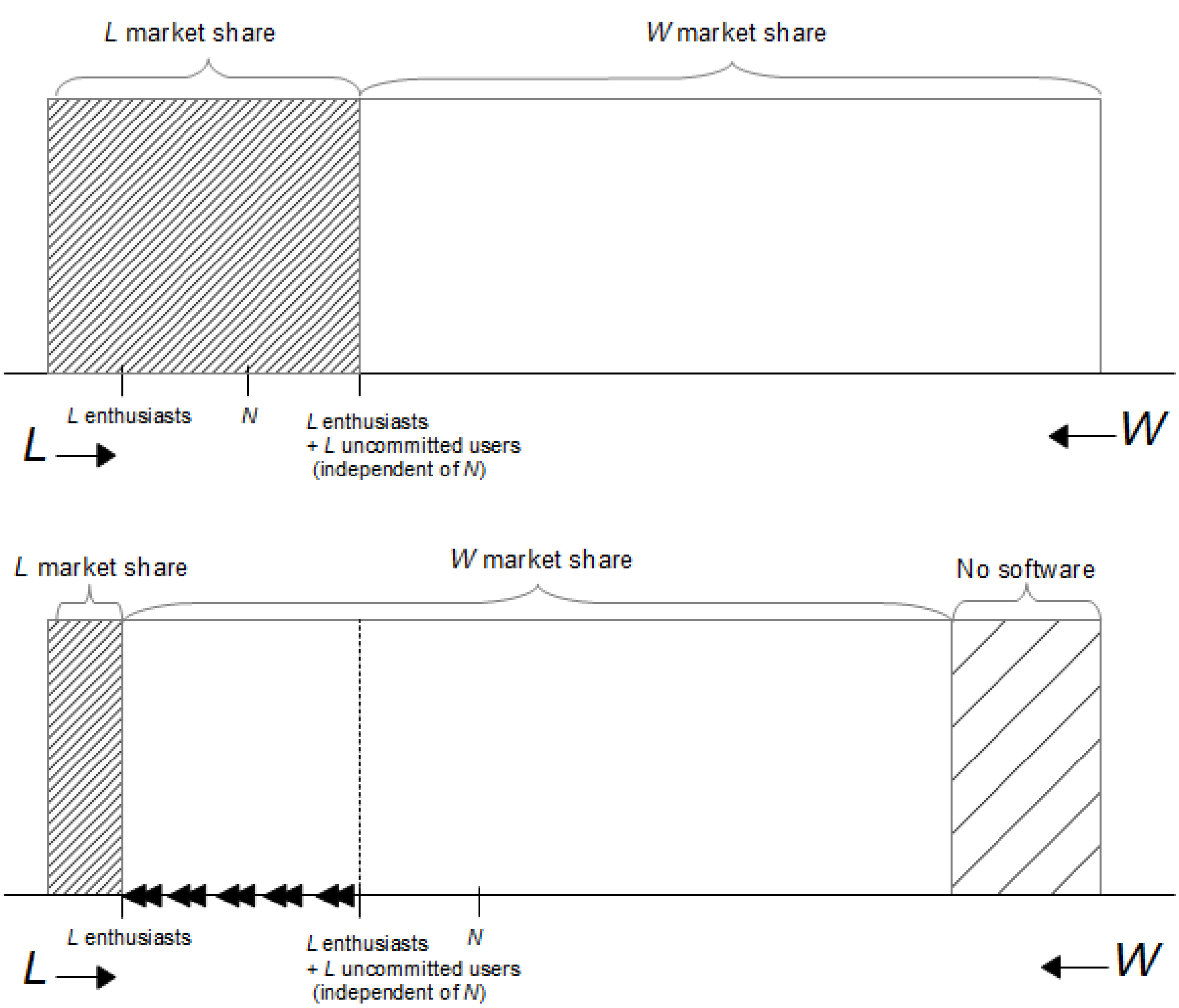

Figure 4.3: An illustration on how market shares for W and L may differ depending on L_{-N} exceeding N or not.

An important determinant for whether L_{-N} or W_{-N} exceed N, may be the shock parameter η. This may especially be important if the demand in absence of network externalities is close to N, and a small shock can determine demand among uncommitted users (*actual demand* among uncommitted users is zero when *demand in absence of network externalities* is below N). Let us consider a case where $W_{-N} > N$ and $L_{-N} = N - \varepsilon < N$, where ε is a small and positive number. Let us further assume that there is a negative shock of η (i.e. in favour of L). Keeping prices fixed, it can be calculated from (2) that $\partial L_{-N} / \partial(-\eta) = \sigma \varphi$. Hence, for few enthusiasts and low heterogeneity in preferences among uncommitted users, even small changes in η can cause substantial changes in L_{-N}. As can be seen from (4) demand for L does not depend on η when $L_{-N} < N$. But if the change in L_{-N} following the decrease in η is so that L_{-N} now exceed N, the software type L will now be considered among uncommitted users and we may have dramatic changes in demand for both W and L. However, if possible (and optimal), the developer of W may respond to such a shock by lowering the price strategically in order so ensure that L_{-N} does not exceed N. Hence, $\Delta\eta$ must be very negative and/or $\sigma\phi$ large in order for strategic pricing not to be optimal for the vendor of W following a negative shock to η.

Table 4.1 gives the explicit expressions for market shares between the software types. The case where $\bar{p}$ is the profit maximising price is omitted from the table. In this case market shares will be given by $W(\bar{p})$ and $L(\bar{p})$ in equations (3) and (4).

	W market share	L market share	No software share
$W_{-N} > N$, $L_{-N} > N$	$\frac{\sigma}{2}[\varphi(v_W - v_L + \eta) + \frac{1}{2}] + \frac{(1-\sigma)(1+\lambda)}{4}$	$\frac{\sigma}{2}[\frac{3}{2} - \varphi(v_W - v_L + \eta)] + \frac{(1-\sigma)(3-\lambda)}{4}$	0
$W_{-N} > N$, $L_{-N} < N$ (no strategic pricing)	$\frac{\sigma}{2}[\varphi(v_W + \eta) + \frac{1}{2}] + \frac{(1-\sigma)(1+\lambda)}{4}$	$(1-\sigma)(1-\lambda)/2$	$1 - \frac{\sigma\varphi}{2}(v_W + \eta) - \frac{\sigma}{4} - \frac{(1-\sigma)(3-\lambda)}{4}$
$W_{-N} > N$, $L_{-N} < N$ (strategic pricing)	$\min(1 + \sigma\varphi - N, 1 - (1-\sigma)(1-\lambda)/2)$	$(1-\sigma)(1-\lambda)/2$	$\max(0, N - \sigma\varphi v_L)$
$W_{-N} < N$, $L_{-N} > N$	$(1-\sigma)(1+\lambda)/2$	$1-(1-\sigma)(1+\lambda)/2$	0

Table 4.1: Market shares of W and L for different cases depending on profit maximising prices of W.

Now that I have derived a model explaining which mechanisms that might be in play in a competitive environment between a non-free software and a free software, the reader is now hopefully excited about moving on to the next section where I show how the mechanisms in the competition between a non-free and a free software might be amplified by allowing piracy of the non-free software.

4.2. Competition with piracy

We now add the possibility of copying *W* for free (piracy of *W*, denoted W^C). We assume that the piracy rate depends positively on the price of *W*. We further assume that those preferring *L* in the no-piracy benchmark are less likely to switch to piracy. Finally, it is assumed that only uncommitted users will consider piracy as an option.

Although the demand functions for *W* and *L* differ when piracy is a feasible choice for the users, the principles regarding the strategy of the vendor of *W* from the no-piracy benchmark still apply. In certain cases when network externalities are strong, the price can be set strategically in order to avoid *L* getting a sufficiently large installed base of users and is thus considered worthless among uncommitted users due to lack of network externalities. However, this may not necessarily be the profit maximising pricing scheme. It turns out that when piracy is included in the competitive environment, the demand for the *L* good more easily falls below the critical level of obtaining sufficient network externalities. Because of this, the strategic pricing scheme is also more often the profit maximising strategy when piracy is present relative to the no-piracy benchmark. Thus, the non-free proprietary type of software is more likely to dominate the market when piracy is present.

4.2.1. Demand

In addition to the non-free *W* software and the free *L* software, uncommitted consumers now have the option of copying *W*. Uncommitted users will prefer piracy to purchasing as long as they get a higher utility of using an unauthorised copy of *W* than what they get from purchasing *W* or obtaining *L* for free. The obvious benefit of piracy is the price, but there might be costs associated with piracy such as costs of finding a working product, deterioration of quality, no customer support, and perhaps even costs associated by the guilt and fear of breaking the law. These "costs of copying" are represented by the exogenously given variable *K*, where a large *K* indicates a high cost of copying.

For mathematical simplicity[36] I assume that the probability that piracy is preferred over *W* for uncommitted *W* users in the no-piracy benchmark is $1 - K/p^2$, and this is thus the share of the uncommitted *W* users that will choose piracy. Likewise a share $\alpha(1 - K/p^2)$ of the uncommitted *L* users in the no-piracy benchmark will switch to piracy, where $K < p^2$ for all profit maximising values of *p*, and $0 \leq \alpha \leq 1$. The parameter α ensures that users of *L* are less likely to switch to

36 That is to get an expression with a tractable solution for the profit maximising price.

piracy of *W* than the buyers of W in the no-piracy case. This comes from the reasonable assumption that users who already have strong preferences for a given software will be more likely to copy that software than users who have preferences for another type of software.

We now return to the total demand of *W*, W^C and *L* respectively. Users' preferences in absence of network externalities are given by:

$$W_{-N}(v_W, v_L, p, \varphi, \eta) = \frac{K}{p^2}\sigma\left(\varphi[(v_W - p) - v_L + \eta] + 1/2\right) + (1-\sigma)(1+\lambda)/2 \tag{8}$$

$$L_{-N}(v_W, v_L, p, \varphi, \eta) = (1-\alpha(1-\frac{K}{p^2}))\sigma\left(1/2 - \varphi[(v_W - p) - v_L + \eta]\right) + (1-\sigma)(1-\lambda)/2 \tag{9}$$

$$W^C_{-N}(v_W, v_L, p, \varphi, \eta) = (1-\frac{K}{p^2})\sigma\left(\varphi[(v_W - p) - v_L + \eta] + 1/2\right) + \alpha(1-\frac{K}{p^2})\sigma\left(1/2 - \varphi[(v_W - p) - v_L + \eta]\right) \tag{10}$$

where the first element of (10) is the no-piracy case *W* users that switch to piracy and the second element is the no-piracy case *L* users that switch to piracy.

It is clear that both *W* and *L* lose market shares to piracy when network externalities are not accounted for. Uncommitted users of the *W* software will switch to piracy as long as the cost of obtaining the pirated product does not exceed the retail price of the legitimate version, and uncommitted users of *L* will switch to piracy whenever the cost of copying does not exceed the retail price of W *and* that the effect of low copying cost is strong enough to make them prefer piracy over *L*.

Piracy and network externalities

Besides taking market share, the presence of piracy may have another interesting feature. As the pirated version of *W* is equal (or at least very similar) to the legitimate version, it may increase the value of *W* as it provides network externalities just as increased usage of the purchased *W* software increases the value of its pirate counterpart W^C, while fewer users will now prefer *L*, meaning less network externalities to the *L* software. As before, demand for a specific type of software depends on whether total user preference for a specific type of software exceeds *N* or not. Total demand for *W*, *L* and W^C respectively is now given by:

$$W=\begin{cases} W_{-N} & \text{if } W_{-N}+W^C_{-N}>N \text{ and } L_{-N}>N \\ \frac{K}{p^2}\sigma \min\left(1,\varphi[v_W-p+\eta]+1/2\right)+(1-\sigma)(1+\lambda)/2 & \text{if } W_{-N}+W^C_{-N}>N \text{ and } L_{-N}<N \\ (1-\sigma)(1+\lambda)/2 & \text{if } W_{-N}+W^C_{-N}<N \text{ and } L_{-N}>N \end{cases}$$

(11)

$$L=\begin{cases} L_{-N} & \text{if } L_{-N}>N \text{ and } W_{-N}+W^C_{-N}>N \\ \sigma+(1-\sigma)(1-\lambda)/2 & \text{if } L_{-N}>N \text{ and } W_{-N}+W^C_{-N}<N \\ (1-\sigma)(1-\lambda)/2 & \text{if } L_{-N}<N \text{ and } W_{-N}+W^C_{-N}>N \end{cases} \quad (12)$$

$$W^C=\begin{cases} W^C_{-N} & \text{if } W_{-N}+W^C_{-N}>N \text{ and } L_{-N}>N \\ (1-\frac{K}{p^2})\sigma \min\left(1,\varphi[v_W-p+\eta]+1/2\right) & \text{if } W_{-N}+W^C_{-N}>N \text{ and } L_{-N}<N \\ (1-\sigma)(1+\lambda)/2 & \text{if } W_{-N}+W^C_{-N}<N \text{ and } L_{-N}>N \end{cases} \quad (13)$$

Equations (11)–(13) have equivalent interpretations as (3) and (4). The main difference is that it is the *sum* of demand for originals of *W* and piracy of *W* in the absence of network externalities that determines whether there will exist demand for *W* among uncommitted users (i.e. W_{-N} + $W^C{}_{-N} > N$ must hold). Also, opposed to (10), the second element in the second line of (13) is now removed since, when $L_{-N} < N$, there simply are no uncommitted *L* users.

4.2.2. Pricing

As in the no-piracy case, the seller/developer of software *W* sets the price in order to maximise profits. However, some market power is now lost as a higher price may lead consumers to piracy, in addition to the competition from the free alternative *L*. Therefore, the demand functions are different in the piracy case. As before, the developer of *W* will maximise the following:

$$\max_p \quad W(p)p$$

The first order condition gives us the following expression for the optimal *p*, when $L_{-N} > N$:

$$p_{+L}=\sqrt{\frac{K\sigma\left(\varphi[v_W-v_L+\eta]+1/2\right)}{(1-\sigma)(1+\lambda)/2}} \quad (14)$$

Hence, K can alternatively be interpreted as the level of threat W faces from piracy. When K is low, many users are willing to obtain unauthorised copies instead of purchasing W. This threat carries over to price which is lowered when the threat from piracy is high. On the other hand if K is high, i.e. that few users are willing to copy, the price is set higher.

Furthermore, in order not to have a negative piracy rate we must have that $K < p^2$. It follows from (14) that this is true whenever $\sigma\left(\varphi[v_W - v_L + \eta] + 1/2\right) > (1-\sigma)(1+\lambda)/2$ holds, which simply states that if W is given away for free, there are more uncommitted W users than there are W enthusiasts.

As before, the profit maximising price is given by:

$$p^*_{+L} = \begin{cases} p_{+L} & \text{if } p_{+L} W(p_{+L}) > \bar{p}\, W(\bar{p}) \\ \bar{p} & \text{if } p_{+L} W(p_{+L}) < \bar{p}\, W(\bar{p}) \end{cases} \qquad (15)$$

The first order condition for a profit maximising price where $L_{-N} < N$ is derived in a similar fashion:

$$p_{-L} = \sqrt{\frac{K\sigma \min\left(1, \varphi[v_W + \eta] + 1/2\right)}{(1-\sigma)(1+\lambda)/2}} \qquad (16)$$

However, if this price is not feasible for $L_{-N} < N$ and a strategic pricing scheme to keep L_{-N} below N is optimal, the profit maximising price is given by (omitting the case where $\bar{p}$ is the profit maximising price):

$$p^*_{-L} = \begin{cases} p_{-L} & \text{if } L_{-N}(p_{-L}) < N \\ L_{-N}^{-1}(N) & \text{if } L_{-N}(p_{-L}) \geq N \end{cases} \qquad (17)$$

4.2.3. Effect on competition

Let us for now assume that $(1 - \sigma)(1 + \lambda)/2 > N$, $(1 - \sigma)(1 - \lambda)/2 < N$ and $1 - (1 - \sigma)\lambda > N$, meaning that the share of W enthusiasts is larger than N, the share of L enthusiasts is below N, and that there are sufficient uncommitted users in the economy so that it is possible that L_{-N} can exceed N. We must have at least $N - (1 - \sigma)(1 - \lambda)/2$ uncommitted users to prefer L over W in the absence of network externalities for uncommitted users to consider using *L*.

The following three possibilities regarding the market participation of the free good *L* arise:

Case 1: $L_{-N} < N$ in no-piracy case => $L_{-N} < N$ when piracy enters the market:
The *L* good is only used by enthusiasts in both cases. Due to competition from unauthorised copying, the price decreases and profits decline.

Case 2: $L_{-N} > N$ in no-piracy case, $L_{-N} > N$ when piracy enters the market:
The *L* good is used by some uncommitted users in both cases. The effect from piracy is not strong enough to push *L* out of the market for uncommitted users. Due to competition from unauthorised copying, the price decreases and profits decline.

Case 3: $L_{-N} > N$ in no-piracy case, $L_{-N} < N$ when piracy enters the market:
The *L* good is used by uncommitted users in the no-piracy case, but as piracy enters the market a sufficiently large share of *L* users switch to piracy or to purchasing *W* in order to push the user share of *L* below the critical level. Due to lack of network externalities from using *L*, the rest of the uncommitted users still preferring *L* (in absence of network externalities), will switch to either to piracy or purchasing *W*. The effect on price and profits is uncertain as competition with piracy has increased, but competition from *L* has decreased. The effect on market share is illustrated in figure 4.4.

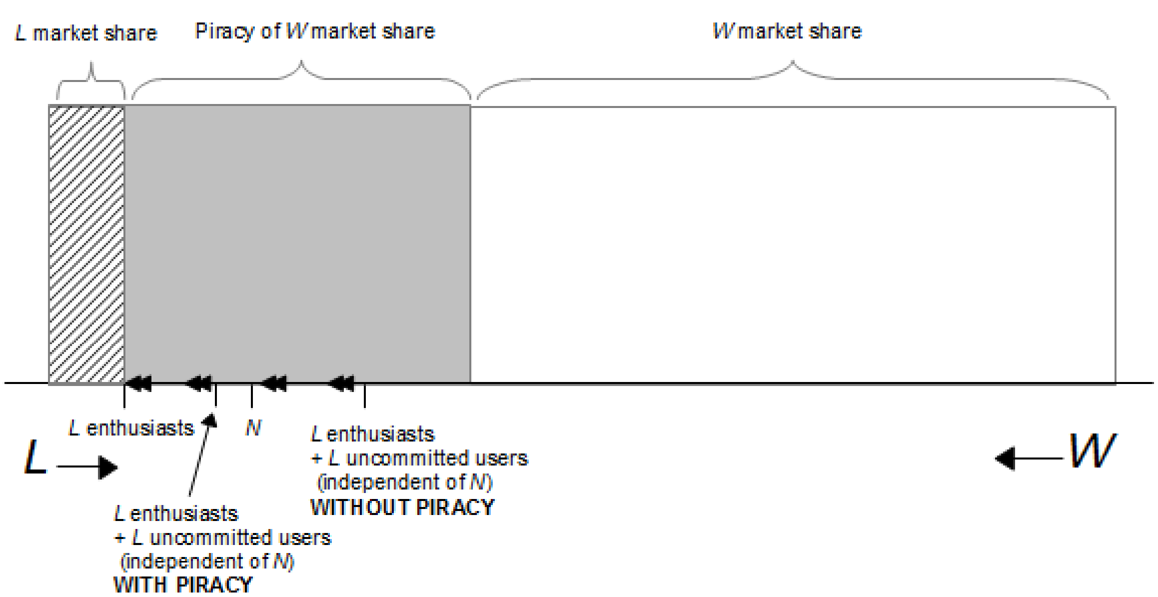

Figure 4.4: Piracy leads to market dominance (case 3).

4.2.4. Numerical examples

In order to make things a bit more tangible I provide some numerical examples to show how piracy may affect prices, competition and profits in the model that has been presented.

Consider the following values of parameters (set so that the assumption of an interior solution holds):[37] $v_W = 2$, $v_L = 1$, $\sigma = 0.7$, $\phi = 0.5$, $\eta = 0$, $\lambda = 0.9$, $K = 0.1$, and $\alpha = 0.5$. In this case, software W has a clear benefit in terms of both quality and by far the majority share of the enthusiasts. As it turns out, as long as the price is set so that $W_{-N} > N$, which automatically holds if $N < 0.285$ (which is the size of W enthusiasts, given the parameters), a slight majority of the market share will go to L with a share of 50.75% while W gets a share of 49.25% (figure 4.5). The profit maximising price is 1.4 (as long as N is small enough so that strategic pricing is not optimal) which yields a profit of 0.69 to W. A majority of the uncommitted users will choose L (approximately 70%), but as W can take advantage of a large share of enthusiasts relative to L, the price can be set higher, provided that $\bar{p} \geq 1.4$ and not too large, which we assume to be true.

When we allow for piracy, the outcome of the model changes substantially. With $K = 0.1$, the threat of piracy is fairly high. In response to this the vendor of W must lower his price substantially, which among other things leads users of L in the no-piracy case to switch to purchasing W (as a response to lower price) as well as a fair share turning into pirates. In this case the shares of W_{-N}, W^C_{-N} and L_{-N} are given by 49.94%, 36.36% and 13.7%, respectively (figure 4.6). As long as all software types exceeds N in demand, the price of W is now given by approximately 0.5, and profits are 0.25.

However, if N is large enough, e.g. $N = 0.15$, L_{-N} will be below N for the given price. It will be optimal for the vendor of W to increase the price slightly until $L_{-N} = 0.15$ holds. The profit maximising price will be 0.61 and the share of users purchasing W will decrease to 47.53%, but profits increase to 0.29. Pirates will take a share 50.97% of the economy, while only L enthusiasts (1.5%) will use the L software.

37 $\frac{-1}{2\varphi} < (v_W - p^*_{+L}) - v_L + \eta < \frac{1}{2\varphi}$

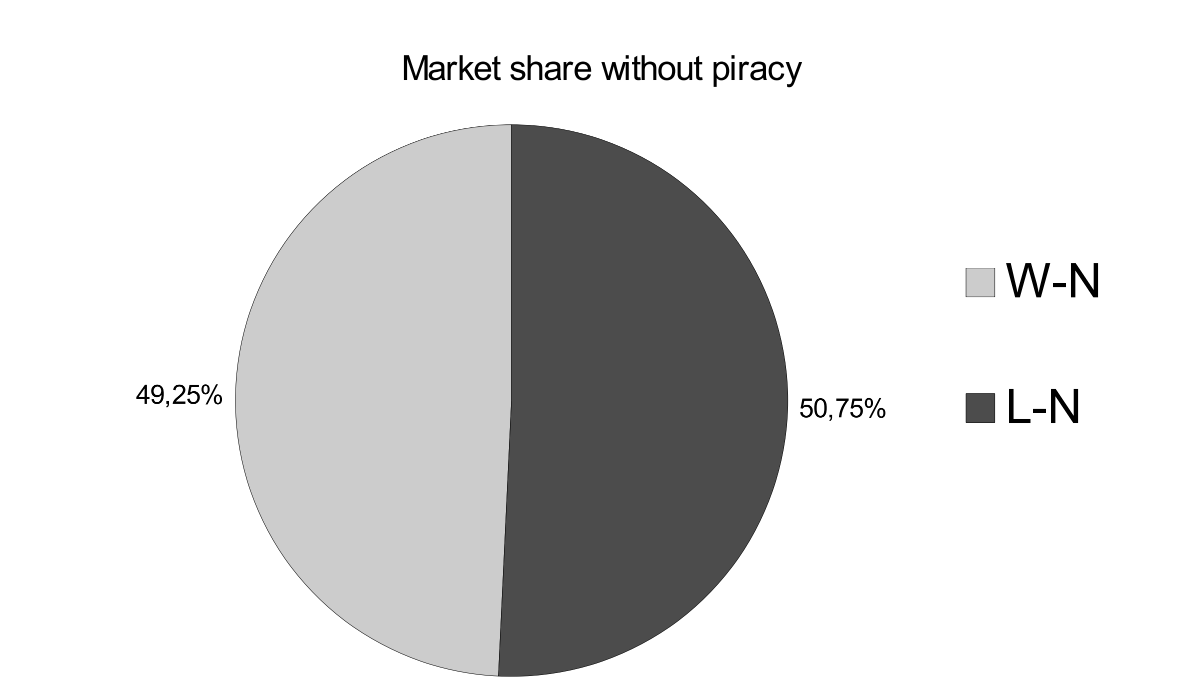

Figure 4.5: Market shares without piracy: Parameters: $v_W = 2$, $v_L = 1$, $\sigma = 0.7$, $\phi = 0.5$, $\eta = 0$, $\lambda = 0.9$.

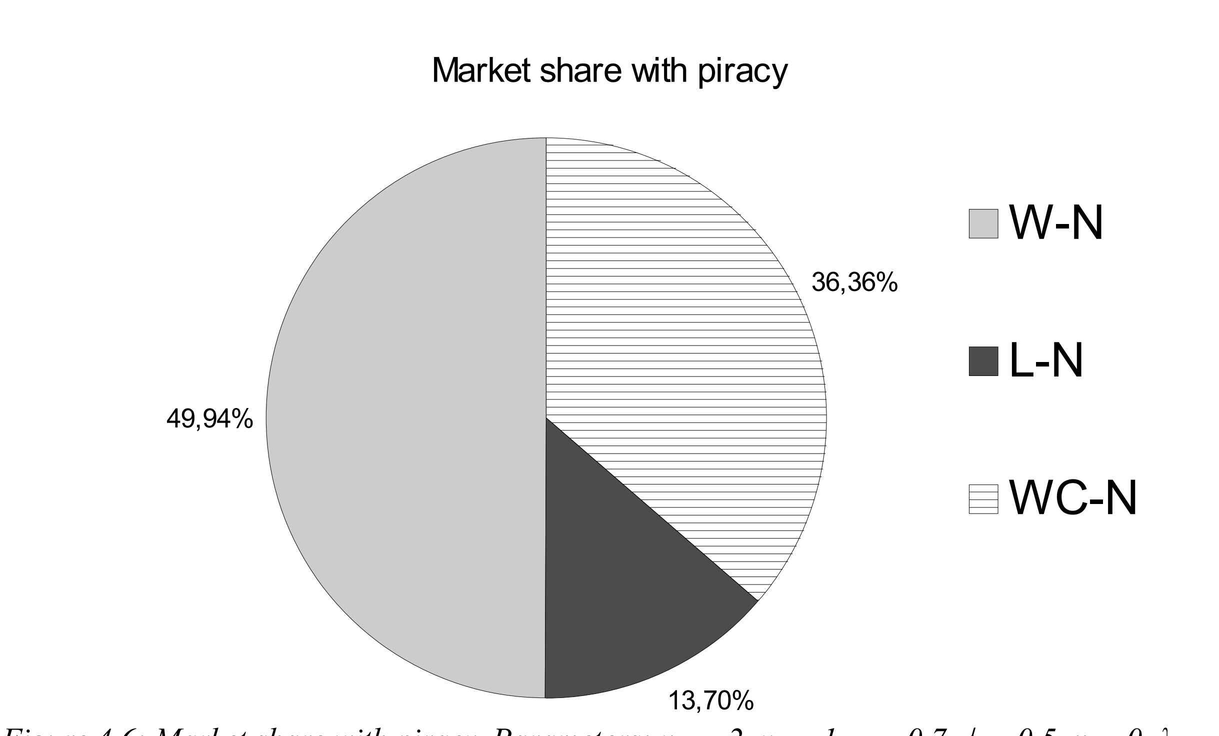

Figure 4.6: Market share with piracy. Parameters: $v_W = 2$, $v_L = 1$, $\sigma = 0.7$, $\phi = 0.5$, $\eta = 0$, $\lambda = 0.9$, $K = 0.1$, $\alpha = 0.5$ and N sufficiently small for uncommitted users to use L.

However, the results will differ substantially if K is increased. If we set $K = 1.5$, while keeping the rest of the parameters unchanged, the market shares of W, W^C and L will be 29.65%, 21.59% and 48.77%, respectively. When the threat of piracy is lower, W can increase the price to 1.92 and enjoy profits of 0.57 which is close to the result of the no-piracy case. However, in this case when the threat from piracy is lower, strategic pricing in order to set $L_{-N} = N$ is not possible. Table 4.2 shows the market outcome with a sufficiently low N for different values of the parameters.

	W_{-N} = W share	L_{-N} = L share	W^C_{-N} = W^C share	p	$W(p)p$
v_W=2, v_L=1, σ= 0.7, ϕ=0.5, η=0, λ=0.9, K=0.1, α=0.5	49.25% without piracy. 49.94% with piracy.	50.75% without piracy. 13.7% with piracy.	36.36% with piracy.	1.41 without piracy. 0.5 with piracy.	0.69 without piracy. 0.25 with piracy.
v_W=2, v_L=1, σ= 0.7, ϕ=0.5, η=0, λ=0.9, K=1.5, α=0.5	49.25% without piracy. 29.65% with piracy.	50.75% without piracy. 48.77% with piracy.	21.59% with piracy.	1.41 without piracy. 1.92 with piracy.	0.69 without piracy. 0.57 with piracy.
v_W=3, v_L=1, σ= 0.5, ϕ=0.5, η=0, λ=0.5, K=0.6, α=0.7	56.25% without piracy. 61.31% with piracy.	43.75% without piracy. 14.05% with piracy.	24.64% with piracy.	2.25 without piracy. 1.09 with piracy.	1.27 without piracy. 0.67 with piracy.
v_W=2, v_L=1, σ= 0.8, ϕ=0.5, η=0, λ=0.5, K=0.7, α=0.6	47.5% without piracy. 15.51% with piracy.	52.5% without piracy. 44.61% with piracy.	39.88% with piracy.	1.19 without piracy. 1.93 with piracy.	0.56 without piracy. 0.29 with piracy.
v_W=1.5, v_L=1, σ= 0.7, ϕ=2, η=0, λ=0.9, K=0.1, α=0.6	66.75% without piracy. 33.93% with piracy.	33.25% without piracy. 29.63% with piracy.	36.44% with piracy.	0.48 without piracy. 0.61 with piracy.	0.32 without piracy. 0.21 with piracy.
v_W=1.3, v_L=1, σ= 0.6, ϕ=0.8, η=0, λ=0.5, K=0.5, α=0.5	37.2% without piracy. 32.1% with piracy.	62.8% without piracy. 57.67% with piracy.	10.23% with piracy.	0.76 without piracy. 0.86 with piracy.	0.29 without piracy. 0.28 with piracy.

Table 4.2: Market shares, prices and profits for different parameter values and sufficiently small N such that L_{-N} and W_{-N} (or W_{-N} + W^C_{-N}) always exceed N.

Let us continue by assuming $N = 0.15$, and the developer of W attempts to set the price so that $L_{-N} = 0.15$. The solution for this price turns out to be a complicated cubic equation. However, with the appropriate software this can be solved without too much effort.[38] Table 4.3 gives the results of strategic pricing of the W with the same parameters as in table 4.2. The results that have changed are presented in **bold** font, and in the results not in bold font it is not possible to strategically keep L out of the market for uncommitted users. All the results from the no-piracy case are such that strategic pricing is not optimal or possible with positive prices for $N = 0.15$.

In the numerical examples of tables 4.2 and 4.3 three observations are worth mentioning: (1) L loses market share when piracy is present in all the cases, (2) profits for W decrease in all the cases, and (3) strategic pricing from the vendor of W is more often optimal. Also, prices can go both up and down in the piracy case relative to the no-piracy benchmark. The W vendor can either fight piracy by reducing prices, or simply charge high prices to the enthusiasts and let the rest use pirated versions or the L software. The fact that L always loses market share in the presence of piracy means that the total user share of L more easily fall below the critical level N which leads to market dominance for the W software. However, in spite of having a dominant market position, the vendor of the W software must be careful in increasing the price too much because higher prices will lead uncommitted users to piracy. Hence, market dominance does not necessarily imply higher profits.

38 In this case LibreOffice Calc.

	W share	*L* share	W^c share	*p*	*W(p)p*
v_W=2, v_L=1, σ=0.7, ϕ=0.5, η=0, λ=0.9, K=0.1, α=0.5	**49.25% without piracy. 47.53% with piracy.**	**50.75% without piracy. 1.5% with piracy.**	**50.97% with piracy.**	**1.41 without piracy. 0.61 with piracy.**	**0.69 without piracy. 0.29 with piracy.**
v_W=2, v_L=1, σ=0.7, ϕ=0.5, η=0, λ=0.9, K=1.5, α=0.5	49.25% without piracy. 29.65% with piracy.	50.75% without piracy. 48.77% with piracy.	21.59% with piracy.	1.41 without piracy. 1.92 with piracy.	0.69 without piracy. 0.57 with piracy.
v_W=3, v_L=1, σ=0.5, ϕ=0.5, η=0, λ=0.5, K=0.6, α=0.7	**56.25% without piracy. 59.64% with piracy.**	**43.75% without piracy. 12.5% with piracy.**	**27.86% with piracy.**	**2.25 without piracy. 1.16 with piracy.**	**1.27 without piracy. 0.69 with piracy.**
v_W=2, v_L=1, σ=0.8, ϕ=0.5, η=0, λ=0.5, K=0.7, α=0.6	47.5% without piracy. 15.51% with piracy.	52.5% without piracy. 44.61% with piracy.	39.88% with piracy.	1.19 without piracy. 1.93 with piracy.	0.56 without piracy. 0.29 with piracy.
v_W=1.5, v_L=1, σ=0.7, ϕ=2, η=0, λ=0.9, K=0.1, α=0.6	**66.75% without piracy. 81.74% with piracy.**	**33.25% without piracy. 1.5% with piracy.**	**16.76% with piracy.**	**0.48 without piracy. 0.36 with piracy.**	**0.32 without piracy. 0.3 with piracy.**
v_W=1.3, v_L=1, σ=0.6, ϕ=0.8, η=0, λ=0.5, K=0.5, α=0.5	37.2% without piracy. 32.1% with piracy.	62.8% without piracy. 57.67% with piracy.	10.23% with piracy.	0.76 without piracy. 0.86 with piracy.	0.29 without piracy. 0.28 with piracy.

Table 4.3: Market shares, prices and profits for different parameter values and N = 0.15.

5. Discussion

In section 4 I presented a static model illustrating the mechanisms in play in a static competitive environment between a free and a proprietary non-free type of software, and showed how the equilibrium prices and market shares may change when piracy becomes an option for the consumers.

In this section I discuss limitations and possible extensions of the model. More specifically I will take a closer look at how reasonable the interpretation of network externalities presented in this model is, as well as discussing coordination issues, implications by introducing dynamics, issues regarding pre-installed piracy and endogenous quality.

5.1. Network externalities

The model in this thesis presents a simplified interpretation of network externalities, where demand for a type of software depends crucially on whether the total demand in absence of network externalities exceeds a given critical mass of the market share. This assumption is most likely too strong. In the numerical example in table 4.3 I assumed $N = 0.15$, and that no uncommitted user would prefer software L if $L_{-N} < N$. Clearly, in a more realistic setting, the utility from using a piece of software characterised by demand side economics of scale where $L_{-N} = 0.14$ would be fairly equal to the utility one would get if $L_{-N} = 0.16$, and we would not get the large differences in user share as predicted by the model. However, I will argue that the assumption regarding network externalities used in the model from the previous section might not be too unrealistic in certain cases.

A common assumption regarding demand for goods with network externalities is the one presented by Katz and Shapiro (1985) where utility $U(x)$ depending on network size x is twice continuously differentiable with $U' > 0$, $U'' < 0$, and $\lim U'(x) = 0$ as $x \to \infty$. Such a function yields diminishing marginal utility with regard to network size, and as the network size grows, the marginal utility of an increase in network size limits to zero (panel (a) in figure 5.1). One can easily argue that this representation of network externalities is reasonable for goods exhibiting *direct network externalities*. E.g. a telephone will be useless and thus worthless if you are the only user. If your best friend gets a phone, the utility you get from owning a telephone will increase a lot, and when your colleagues and acquaintances start using phones the utility you get from the phone will increase further. When the network size becomes sufficiently large, utility will seize to increase.

However, in the markets for software, although there might be direct network externalities associated with a piece of software (e.g. social media and online games), *indirect network externalities* are often more important. With indirect network externalities, the utility of using a good does not increase directly from an increase in network size, but an increased network size may result in a larger supply of complementary goods to the network good which in turn increase your valuation of the network good. E.g. in the case of operating systems direct network externalities are weak, as it is more or less unproblematic for e.g. a Linux user to communicate with a Windows user. However, in order to be profitable for an independent software developer to invest in the production of a piece of software compatible with Linux, the software developer must expect that the value of sales he gets from Linux users exceeds the fixed cost of developing Linux compatible applications.

Let us assume that there are more Windows users than Linux users ($W > L$), and that there exists a computer game compatible with Windows that sells at a price p. The developers of this game must now decide whether they should develop a Linux compatible version of this game. At price p he expects to sell this game to a share s of the Linux users.[39] The cost of developing a Linux compatible version of the game is F_L. The condition for expected profits to be positive is $psL - F_L \geq 0$, which translates to a critical value of $L \geq F_L/ps =: \overline{L}$ for the software developer to be willing to invest in making a Linux compatible version of the game. Hence, the utility by using Linux will increase once $L \geq \overline{L}$, and $\overline{L}$ is thus equivalent to the N found in the model from section 4. If we assume one representative independent software developer with the participation constraint described above, the utility with respect to network size will be as in panel (d) in figure 5.1.

If we more realistically assume that there exists a continuum of independent software developers with different values of $\overline{L}$, the utility function with respect to network size might look like those in panel (b) or (c) in figure 5.1. For only a few Linux users, very few complementary goods to Linux will exist as producing any further such goods will not be profitable. Then, as the network size increases a little more, the types of applications with the highest values of p and/or s and/or low F_L, will exist in Linux compatible versions. In particular if these goods are characterised with a high s, they can be interpreted as the most popular types of applications (or necessities), such as web browsers and PDF readers. Once these necessities are Linux compatible, the utility from using Linux increases a lot, and it may be reasonable to assume that

39 If the potential Linux compatible software is free, p might instead represent the expected revenue the software developer will get per user (from e.g. advertising revenues, sales of upgraded premium versions or paid support).

the utility function is convex with respect to network size for a small and medium sized network. As the network size increases further, the additional complementary goods available are less useful to most of the Linux users, hence decreasing marginal utility with respect to network size is more likely for large networks.

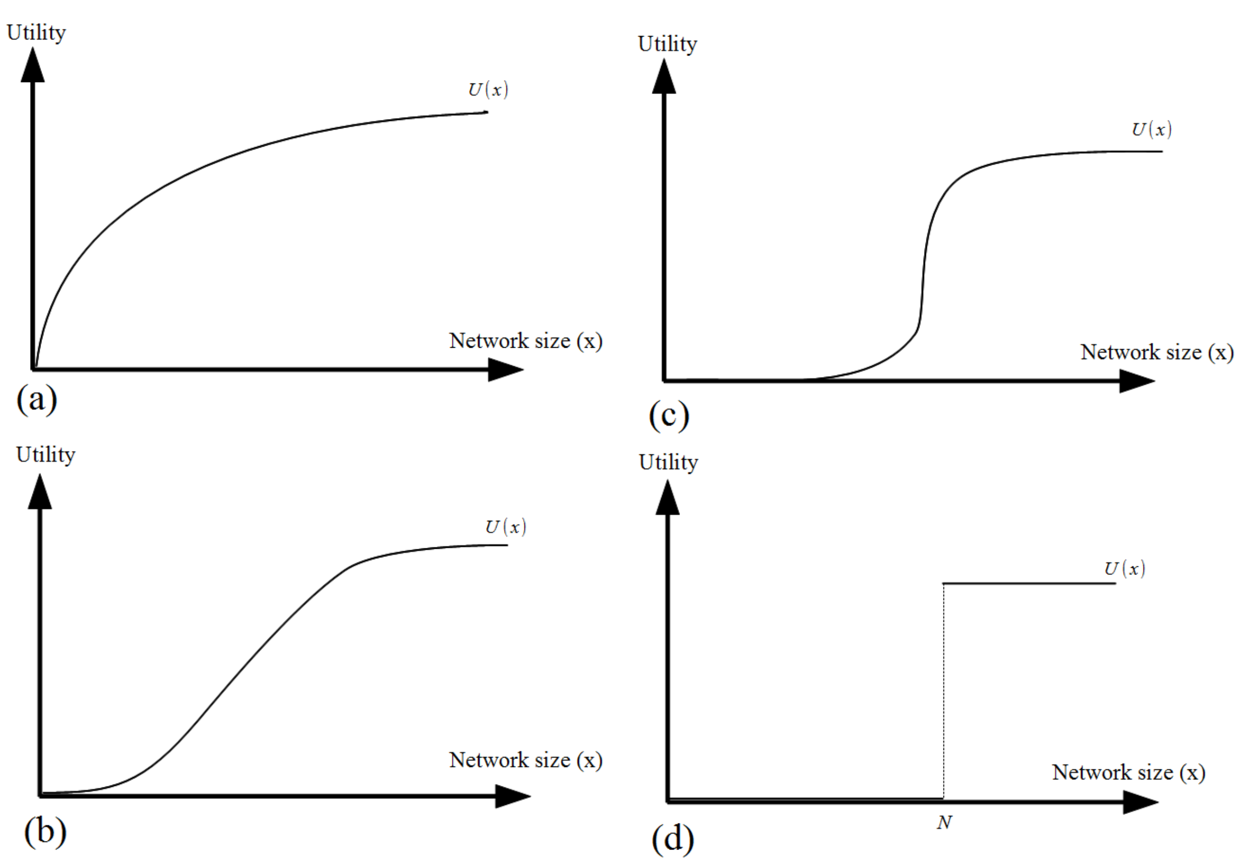

Figure 5.1: Examples of utility functions depending on network size.

If we for the sake of concreteness assume that the utility function described in the previous paragraph is similar to the cdf of the normal distribution, i.e.

$$U(x)=K\int_{-\infty}^{x} e^{\frac{-(x-\mu)^2}{2\sigma^2}}\,dx,$$

then the utility function will look something like that in panel (b) for large values of σ^2, and like the one in panel (c) of figure 5.1 for small values of σ^2, where σ^2 is interpreted as Var($\overline{L}$). From the figure, it can be seen that the utility function in panel (d), which is equivalent to the representation of network externalities in the model of section 4, may work as a fairly good approximation of the utility function in panel (c). However, if network externalities are represented more accurately by the utility functions of panel (b) (large variance in the critical value $\overline{L}$ for potentially Linux compatible applications), the effects from network externalities are not as “tippy” as assumed in the model, and the model's implication will be less reasonable.

5.2. Coordination

One important assumption in the model presented, is that an uncommitted user will use software L whenever $\eta + \omega + (v_W - p) - v_L < 0$ and $L_{-N} > N$. If we go from a situation where $L_{-N} < N$ to one of $L_{-N} > N$, the market will tip in the sense that demand for L will increase substantially. It follows from this assumption that all uncommitted users expect the other uncommitted users to switch software types whenever this is true, i.e. that $L_{-N} > N$ implies $L > N$.

Let us consider a simple game with two uncommitted W users where $\eta + \omega + (v_W - p) - v_L < 0$ is true for both players (i.e. they both prefer L to W in absence of network externalities). They both get the pay-off of 1 from using W when $W > N$ and pay-off of 2 from using L when $L > N$. They will get pay-off of 0 from using software L (software W) when $L < N$ ($W < N$). We assume that $W > N$ holds independently of what type of software they use, and that $L > N$ only holds when both players switch to L, and $L < N$ otherwise.

1\2	*W*	*L*
W	**<u>1</u>,<u>1</u>**	1,0
L	0,1	**<u>2</u>,<u>2</u>**

Table 5.1: A coordination game

As seen in table 5.1, this becomes a classic stag hunt coordination game with two Nash equilibria: $\{W,W\}$ and $\{L,L\}$. Hence, the implication $L_{-N} > N \Rightarrow L > N$ is true only if player 1 believes that player 2 believes that player 1 will choose software L, and vice versa, and that the beliefs are correct.

Moreover, if we include uncertainty, e.g. that player 1 thinks player 2 chooses L with probability q (the probability that player 2 believes that player 1 will choose L), and that player 2 is aware of player 1's beliefs, we must have $q \geq ½$ in order for $\{L,L\}$ to be the actual outcome. If we furthermore assume risk aversion, so that we have diminishing marginal returns of utility to pay-offs, e.g. $U(0) = 0$, $U(1) = 1$ and $U(2) < 2$, even greater values of q is required in order to get the Pareto optimal Nash equilibrium $\{L,L\}$. This is because opposed to choosing software L, there is no uncertainty in pay-offs by choosing software W given the assumption that we have $W > N$ independent of the players' strategies. We thus have one pay-off dominant Nash equilibrium, $\{L,L\}$, and one risk dominant Nash equilibrium, $\{W,W\}$.

5.3. Dynamics

While the modelling in this thesis covers the competitive aspects of software piracy, it fails to capture the possibility that piracy potentially may be profitable over time. As mentioned earlier in this paper, this possibility certainly has been recognised by executives in Microsoft, and is also, at least theoretically, covered in the economics literature. It is argued that piracy may work as a quality revelation device (Darmon et al., 2009; Takeyama, 2003) and as an efficient method of committing to high prices in an intertemporal setting (Takeyama, 1997).

In the theoretical framework of Takeyama (1997) a two period model is used. It is shown that if users with low valuation for the piece of software regard the pirated product as a very close substitute to the original, and those with high valuation never will copy, the software vendor can credibly commit to higher prices and sell only to high-valuation customers. When piracy is not possible, the firm cannot credibly commit to high prices as high valuation users rationally will anticipate that prices will be reduced in the next period in order to extract revenues from low valuation users. Hence, profits may increase in the presence of piracy if selling for high prices to the few is more profitable than selling for low prices to more individuals in the no-piracy case.

In a similar modelling set-up Takeyama (2003) shows that if copies and originals are imperfect substitutes and there are informational asymmetries between firms and consumers with regard to product quality, copiers may turn into legitimate users in a subsequent period as piracy reveals the quality of the product. Because piracy removes the informational asymmetries between firms and consumers, firms selling high quality products may increase the price relative to the no piracy benchmark. In fact, allowing piracy to some extent may work as a signalling device for high quality and in this way force low quality firms to exit the market.

Because software is an experience good, Darmon et al. (2009) claim that piracy is equivalent to giving away a free sample that reveals the quality of the piece of software. In their model, software quality is in part revealed by experience and in part by word-of-mouth effects from users (pirates and buyers) to potential users. Hence, pirates help reveal quality which in turn may increase demand, and may in certain cases increase firm profits, even in the absence of network externalities.

Consumer lock-in is another important factor that the model does not explicitly capture.[40] Piracy may certainly help locking in consumers, and if these consumers some day for some reason cannot or will not get a pirated product, chances are that they will purchase the piece of

40 The "enthusiasts" could be interpreted as locked-in users. However, the model of this paper would still not capture the dynamics of consumer lock-in.

software they used to copy illegally. This could for instance apply to amateurs having to purchase a piece of software as they start their professional careers. Anti-piracy laws are rarely enforced in the case of individuals obtaining unauthorised copies of proprietary software for private consumption. However, when it comes to software used in professional businesses, at least in developed countries, firms risk having to pay millions of dollars in fines if caught using unauthorised copies of software.[41] For instance a hobby web designer using a pirated version of Adobe CS who later on starts his own business, would better get a legitimate piece of software if he wants to turn his talents into profits rather than fines. If he is talented (and lucky) enough to have a successful business, purchasing Adobe CS for around 700 USD might actually be cheaper than the cost of learning how to properly use e.g. the free and open-source program GIMP.[42] Had piracy not been feasible, he might had used GIMP from the very start, unless Adobe priced its Creative Suits software cheaper in response to the open-source competition. Thus it is not unthinkable that piracy may turn into profits in certain cases, as piracy may be equivalent to the "freemium" business strategy.

5.4. "Enthusiasts" and pre-installed piracy

As briefly suggested earlier, the enthusiasts from the model may not always be *enthusiasts* in the true sense of the word. Rather they might be users who purchase computers with pre-installed software who never consider using alternative software to what is already bundled with the computer. Pre-installed software, such as an operating system in addition to extra pre-installed software that may be included for a fee after a persuasive sales pitch at the local department store, are obviously not pirated goods and may give a competitive advantage to proprietary software developers, as this may increase the share of "enthusiasts".

However, this is not necessarily true for developing countries with more lenient enforcement of copyright laws. In China, for instance, computers with unlawfully pre-installed pirated software are sold in the open, and are perhaps even more easily obtainable than computers with legitimate pre-installed software.[43] Furthermore, individuals may not even be aware that they are using pirated software, and, according to a survey conducted by the Business

41 "BSA Members Secure Largest Ever Worldwide Settlement of US$5.7 million in Japanese Software Piracy Case", Business Software Alliance, September 6, 2011: http://www.bsa.org/country/News%20and%20Events/News%20Archives/en/2011/en-09062011-japan.aspx, retrieved 29 February 2012.

42 This short story could also be about the economics student using a pirated version of STATA in his master's thesis, and choosing to purchase STATA for 1400 USD when he becomes a professional analyst rather than downloading Gretl for free.

43 "China (PRC): 2012 Special 301 Report on Copyright Protection and Enforcement", International Intellectual Property Alliance (IIPA), p. 40.

Software Alliance, half of the business decision makers in developing countries believe that it is legal to purchase one legal copy of a piece of software and install it on multiple computers.[44]

Hence, the enthusiasts, assumed never to use pirated software in the model, may in China and other comparable countries almost exclusively consist of pirates. Obviously, pre-installed piracy most likely have serious negative effects on profits for companies such as Microsoft. Furthermore, unlawfully pre-installed Windows computers is yet another hindrance for people in developing countries to actively choose if they want to purchase a legitimate version of Windows or if they want to use e.g. a Linux OS for free.

If piracy of W is the default option for a large share of users in the theoretical analysis (i.e. “enthusiasts”), and there are very few ordinary W enthusiasts, the only way to respond to the competition from piracy is to lower the price, and most likely lower the price in a much larger extent than in the cases discussed in the model section. As we have seen, a lower price of W leads to lower demand for the free L software type, and with sufficiently strong network externalities, dominance by W is likely to occur.

5.5. Endogenous quality

In a static modelling environment exogenous levels of quality may not be an unreasonable assumption since these are likely to be fixed in the short run. However, in the asymmetric competitive environment between a free and a non-free type of software, endogenous levels of quality may in fact be particularly interesting. When the free type of software also is open-source, the development of the two software types happens in very different manners.

For the proprietary type of software quality must be assumed to be costly to produce (but free to reproduce). Since (uncommitted) users' preferences to a large degree depend on relative quality between the two software types, a Hotelling-like model would predict that when there is low heterogeneity in preferences (large ϕ), i.e. fierce competition for the uncommitted users, the proprietary software developer will invest more in quality. Moreover, an increase in quality by the open-source type of software will typically be answered by investing in higher quality of the proprietary type of software in combination with reduced price.

Open-source software development works in a completely different manner. As the product to a large extent is developed for free by its own users, it is natural to assume that quality increases with user share. Without going too deep into the motivation of voluntary source code contributions, it is reasonable to assume that contributions and thus quality increase with the size

44 Business Software Alliance (2011), pp. 3-4.

of installed base. This can be argued for at least two reasons: (1) More users increase the size of possible contributors, and (2) reputational benefits from contributing are most likely stronger for more popular types of open-source software. As argued by Lerner and Tirole (2002) reputational benefits may be important in explaining source code contributions. Hence, active code contributors of popular types of open-source projects, such as the Linux kernel, will get more attention (i.e. higher reputation) relative to active contributors of more obscure open-source projects.[45]

Hence, a large user share of the open-source type of software, which translates into strong competition for the proprietary type of software, is predicted to cause high quality for both types of software. However, if the situation is so that the proprietary type of software is dominating in sense of market share, low quality of both types of software is a likely outcome. As the implications of the model suggest, piracy combined with strong network externalities may result in market dominance for the proprietary type of software. Taking endogenous quality into account, piracy may also lead to lower quality for both types of software due to lower degree of competition for the proprietary type of software and low user share for the open-source type.

However, it can be argued that the developer of the proprietary type of software may strategically invest in high quality as a measure of keeping user share and thus quality of the open-source competitor low. If this is the case, the market will be characterised by dominance by a high-quality proprietary type of software over a low-quality open-source type of software.

45 Although the user share of Linux desktop OS is low, software built on the Linux kernel is widely used in various sorts of technology such as servers, mobile phones and routers. There are also examples of technology built on the Linux kernel such as GPS devices, cars, robots and even refrigerators. Measured in contributors, lines of code and code edits, the Linux kernel project is by far the largest open-source project.

6. Empirical evidence

In section 3 I mentioned the brief empirical analysis presented by Casadesus-Masanell and Ghemawat (2006), where they find that piracy has a significant negative effect on Linux/Windows server OS shipments ratio, and further that per capita income has no significant effect.

I find reason to criticise the methodology for three reasons: (i) The market for server OS differs from the market for desktop OS, as server OS primarily is used in businesses and organisations rather than for personal use. It is likely that desktop OS is more prone to piracy and will perhaps be a better dependent variable in such an analysis. (ii) Their OS data is based on *shipments* of Windows and Linux server OS. As Linux is available for free by downloading on the internet, Linux shipments are likely to be an inaccurate estimate of Linux usage. (iii) The Linux/Windows ratio is used as dependent variable. This ratio is possibly sensitive to other types of OS usage such as OS X (Apple), meaning that an increase in OS X usage may translate into an increase in Linux/Windows ratio without Linux usage having increased.[46] Moreover, the empirical results are based on only 45 observations.

I have compiled data on piracy rates, GDP per capita (PPP) (USD) and Linux desktop OS user shares from 107 countries. The piracy rates are collected from the 2010 BSA Global Piracy Study; GDP data from 2010 are, with a few exceptions, collected from the World Bank;[47] and Linux shares are based on web counter data from October 2011 by Net Market Share.[48] The piracy data and the Linux share data may in part suffer from measurement error and sampling bias, respectively. As mentioned in section 2, national income has been used in estimating piracy rates in countries where software usage has not been directly measured (Png, 2010). This suggests that the GDP data and piracy rates data may be too strongly correlated relative to the correlation one would find relative to the "real" piracy rates. The Linux share data from Net Market Share is based on approximately 160 million unique visits per month to approximately 40,000 sites using the web analytics services by Net Applications.[49] Some of the observations differ substantially from statistics published by similar web analytics services. This may be due

46 E.g. in my dataset the Linux shares of Costa Rica and Paraguay are estimated to be 2% and 2.01%, respectively. Linux/Windows ratios on the other hand are estimated to be 8% higher in Costa Rica than in Paraguay due to large differences in Windows shares.

47 The following countries/territories have other sources of 2010 GDP (PPP) per capita data: Puerto Rico (CIA World Factbook), Taiwan (IMF) and Zimbabwe (IMF).

48 Desktop Operating System Market Share, Net Market Share: http://marketshare.hitslink.com/operating-system-market-share.aspx?qprid=8&qpcustomd=0

49 "Net Market Share Frequently Asked Question": http://www.netmarketshare.com/faq.aspx , retrieved 13 March, 2012.

to sampling bias as the vast majority of the sites using the Net Market Share service is US based and in English. If Linux usage within a country correlates with surfing behaviour to English language web sites, the data may not reveal "true" Linux usage within countries. Also, as Linux user shares in generally are fairly low, we might get fairly large relative differences in estimates, especially for countries with low internet activity, meaning that there are few observations behind the Linux share estimates.

I regress Linux share on piracy rate, GDP per capita and regional dummies where the dummy coefficients represent differences in the intercept relative to Western Europe. Table 6.1 regresses Linux shares on piracy rates and control variables in absolute terms, while table 6.2 regresses the logarithmic form of Linux shares and piracy rates. The regression outputs are given by (1) and (4) in table 6.1 and 6.2. Regression outputs (2) and (5) come from using sequential elimination with two-sided p-values of 0.10 of (1) and (4), respectively. Finally, I regress Linux shares on piracy rates and GDP per capita without regional dummies in regressions (3) and (6).

As the regression outputs (1)–(6) suggest, piracy has a significant negative effect on Linux desktop OS user share. When having controlled for piracy, per capita income gets the expected negative sign.[50] Furthermore, where regional dummies are not included, per capita income has no significant effect on Linux usage. Regression output (3) shows that the Linux user share is predicted to decrease by approximately 0.016 percentage points following a 1 percentage point increase in the piracy rate (p-value 0.0562). By using natural logarithms of piracy rate and Linux user share as in regression output (6), Linux user share is predicted to decrease by 0.66% following a 1% increase in the piracy rate.

These empirical findings support the prediction from my theoretical model, namely that piracy affects use of open-source software negatively. Furthermore, my findings are consistent with those of Casadesus-Masanell and Ghemawat (2010) in the sense that piracy is a more important determinant of Linux usage than per capita income. It is also worthwhile noticing that even in the case of a piracy rate of zero, the Linux share is predicted to be quite low (i.e. the intercept where per capita income is insignificant). A likely reason for the low predicted Linux usage is the lack of network externalities generated by use of Linux desktop OS. Preferably similar data on open-source type of software where network externalities are less important in explaining user share would be used.[51]

50 Corr(Linux share, ln(GDP/capita)) = 0.1177 and Corr(ln(Linux share), ln(GDP/capita)) = 0.1026.

51 The OpenOffice.org user share data only got 27 observations, and there is too little variation in the data to find any significant effect from piracy when having controlled for GDP/capita.

	Linux share (1)	Linux share (2)	Linux share (3)
Intercept	**0.0548394 *****	**0.054716 *****	**0.0328743 ***
	(0.0192413)	(0.0188954)	(0.0198037)
Piracy rate	**-0.0236893 *****	**-0.0216238 *****	**-0.0157453 ***
	(0.00852962)	(0.00782467)	(0.00815481)
ln(GDP/capita) (PPP)	**-0.00522359 ***	**-0.00311412 ***	-0.00132939
	(0.00160693)	(0.00158633)	(0.00166504)
Asia Pacific	0.00666548	**-0.00626468 ****	
	(0.00360065)	(0.00286683)	
Central and Eastern Europe	**0.00216757 ***	**0.00548235 ***	
	(0.003627)	(0.00276387)	
Latin America	-0.00532519		
	(0.00392031)		
Middle East and Africa	-0.00130609	**-0.00653328 ***	
	(0.00359581)	(0.00265568)	
North America	-0.0031088		
	(0.00625958)		
R^2	0.215132	0.211904	0.047968
N	107	107	107

*Table 6.1: OLS regression output with Linux share as dependent variable. Standard errors are in parentheses. *, ** and *** represent significance at the 10%, 5% and 1% level, respectively. By White's test there is evidence for heteroskedasticity of output (2). Due to small magnitude of heteroskedasticity and for the sake of consistency non-robust standard errors are given.*

	ln(Linux share) (4)	ln(Linux share) (5)	ln(Linux share) (6)
Intercept	**-2.20491** ** (0.868148)	**-2.24514** *** (0.818425)	**-4.03535** *** (0.948454)
ln(piracy rate)	**-0.823826** *** (0.266668)	**-0.723923** *** (0.227664)	**-0.661282** ** (0.275404)
ln(GDP/capita) (PPP)	**-0.293106** *** (0.0971777)	**-0.277959** *** (0.0951554)	-0.126005 (0.112842)
Asia Pacific	**-0.785088** *** (0.235993)	**-0.822413** *** (0.1764)	
Central and Eastern Europe	0.137225 (0.24218)		
Latin America	0.0227774 (0.262146)		
Middle East and Africa	**-0.99231** *** (0.239626)	**-1.0436** *** (0.158696)	
North America	-0.321352 (0.409567)		
R^2	0.386337	0.378738	0.062492
N	107	107	107

*Table 6.2: OLS regression output with ln(Linux share) as dependent variable. Standard errors are in parentheses. *, ** and *** represent significance at the 10%, 5% and 1% level, respectively.*

7. Conclusion

This thesis examines how piracy affects competition between a non-free proprietary type of software and a free/open-source type of software. Furthermore, I examine more specifically how software piracy may have an impact on prices, profits as well as strategies towards the free competitor for the non-free proprietary software vendor.

In approaching these issues, a model has been developed where a proprietary software vendor faces competition from a free/open-source type of software. I find that, when the proprietary software has a relatively large constant installed user base ("enthusiasts") and qualitative differences between the two software types are not too large in the presence of network externalities, a pricing scheme driving the free software type out of competition is usually not optimal. Rather, profits will be higher by taking advantage of the users with strong preferences for the proprietary software and consequently set fairly high prices. When this is the case, the proprietary software developer can enjoy high profits, and users are free to choose the software of their liking.

As piracy enters competition, the proprietary software vendor can react either by increasing prices and extract revenues from those still willing to purchase software and let the rest copy software for free, *or* by reducing prices as a measure of fighting piracy. If the latter strategy is optimal, the free/open-source type of software may lose a fairly large user share to both piracy and purchasers of the proprietary software type. Furthermore, the optimal strategy for the proprietary software vendor may be a pricing scheme where the free/open-source software type is deliberately prevented from getting sufficient network externalities in order to be competitive, whereas this is not necessarily the optimal pricing scheme had piracy not existed. Even though piracy may lead to market dominance whenever network externalities are strong, profits, at least in a static environment, are still likely to decline in the presence of piracy.

Obviously, the conclusions hinge on the assumptions of the model. For the sake of simplicity I sacrificed the more traditional assumptions in describing network externalities. If utility is assumed to be continuously growing in network size, the optimal pricing strategies may differ somewhat. However, I believe that what would have been gained in realism would not exceed the loss in complexity in this specific modelling environment. Moreover, I have argued that the interpretation of network externalities used may in certain cases be a good approximation.

The most critical limitation to the model is perhaps the lack of dynamics, which may have implications for the results. Piracy may lead to consumer lock-in as well as providing an instrument of quality revelation. In turn, piracy may increase sales as locked-in consumers of pirated software may end up as legitimate users. Also, due to word-of-mouth effects, pirates may help reveal quality to non-users, who in turn may end up purchasing the software. Furthermore, in cases where unauthorised copies and originals are imperfect substitutes, piracy may be equivalent to a “freemium” business strategy.

Coordination issues and institutional factors should perhaps also be accounted for in an ideal description of a competitive environment with proprietary software, piracy and free/open-source software. It can be argued that the latter may be important in explaining both the widespread use of piracy in developing countries and, equally important, the limited success of open-source software caused by piracy.

It can also be argued that if endogenous quality is assumed, piracy may lead to lower quality of both types of software if the free software also is open-source. This is because piracy may lead to market dominance of the proprietary type of software and thus decrease the degree of competition which gives disincentives for costly investment in quality. It is also likely that quality of open-source software depends positively on the size of its user share. Since piracy is likely to reduce demand of free/open-source software, it can be argued that piracy may have a negative effect on quality of open-source software as well.

The empirical analysis provided in this paper supports the main prediction from the theoretical model: that usage of open-source software is negatively affected by the prevalence of piracy.

Irrespective of how piracy affects prices and firm profits, one thing seems certain: piracy hurts free software, and open-source software in particular. Open-source software may represent a threat to segments of the proprietary software industry, but similarly represent a great opportunity for firms and consumers to increase productivity and may even be important for economic growth. Advocates for open-source software thus have good reasons to fight piracy.

References

Anderson, Chris (2009), *Free: The Future of a Radical Price*, 1st paperback edition, Hyperion.

Bae, Sang Hoo and Jay Pil Choi (2006), “A Model of Piracy”, *Information Economics and Policy*, Vol. 18, pp. 303-320.

Besley, Timothy, Torsten Persson and Daniel M. Sturm (2010): “Political Competition, Policy and Growth: Theory and Evidence from The United States”, *Review of Economic Studies*, Vol. 77, No. 4, pp. 1329-1352.

Bitzer, Jürgen (2004), “Commercial Versus Open Source Software: The Role of Product Heterogeneity in Competition”, *Economic Systems*, Vol. 28, No. 4, pp. 369-381.

Business Software Alliance (BSA) (2011), *Eight Annual BSA Global Software 2010 Piracy Study*.

Casadesus-Masanell, Ramon and Pankaj Ghemawat (2006), “Dynamic Mixed Duopoly: A Model Motivated by Linux vs. Windows”, *Management Science*, Vol. 52, No. 7, pp. 1072-1084.

Darmon, Eric, Alexandra Rufini and Dominique Torre (2009), “Back to Software “Profitable Piracy”: The Role of Information Diffusion”, *Economics Bulletin*, Vol. 29, No. 2, pp. 543-553.

Economides, Nicholas (1996), “The Economics of Networks”, *International Journal of Industrial Organization*, Vol. 14, No. 6, pp. 673-699.

Economides, Nicholas and Evangelos Katsamakas (2006), “Two-Sided Competition of Proprietary vs. Open Source Technology Platforms and the Implications for the Software Industry”, *Management Science*, Vol. 52, No. 7, pp. 1057-1071.

Jain, Sanjay (2008), “Digital Piracy: A Competitive Analysis”, *Marketing Science*, Vol. 27, No. 4, pp. 610-626.

Katz, Michael L. and Carl Shapiro (1985), "Network Externalities, Competition, and Compatibility", *The American Economic Review*, Vol. 75, No. 3, pp. 424-440.

Lakhani, Karim R. and Robert G. Wolf (2005), "Why Hackers Do What They Do: Understanding Motivation and Effort in Free/Open Source Software Projects", in: Joseph Feller, Brian Fitzgerald, Scott A. Hissam and Karim R. Lakhani (eds.), *Perspectives on Free and Open Source Software,* Cambridge, Massachusetts: The MIT Press, pp. 3-21.

Lanzi, Diego (2009), "Competition and Open Source with Perfect Software Compatibility", *Information Economics and Policy*, Vol. 21, No. 3, pp. 192-200.

Lerner, Josh and Jean Tirole (2002), "Some Simple Economics of Open Source", *The Journal of Industrial Economics*, Vol. 50, No. 2, pp. 197-234.

Png, Ivan P. L. (2010), "On the Reliability of Software Piracy Statistics", *Electronic Commerce Research and Applications*, Vol. 9, No. 5, pp. 365-373.

Shapiro, Carl and Hal R. Varian (1999), *Information Rules: A Strategic Guide to the Network Economy*, Harvard Business School Press.

Takeyama, Lisa N. (1994), "The Welfare Implications of Unauthorized Reproduction of Intellectual Property in the Presence of Demand Network Externalities", *The Journal of Industrial Economics*, Vol. 42, No. 2, pp. 155-166.

Takeyama, Lisa N. (1997), "The Intertemporal Consequences of Unauthorized Reproduction of Intellectual Property", *Journal of Law and Economics*, Vol. 40, No. 2 , pp. 511-522.

Takeyama, Lisa N. (2003), "Piracy, Asymmetric Information and Product Quality", in: Wendy J. Gordon and Richard Watt (eds.), *The Economics of Copyright: Developments in Research and Analysis*, Northampton, Massachusetts: Edward Elgar Publishing Ltd, pp. 55-65.

Varian, Hal R. (2000), "Buying, Sharing and Renting Information Goods", *The Journal of Industrial Economics*, Vol. 48, No. 4, pp. 473-488.

Varian, Hal R., Joseph Farrell and Carl Shapiro (2004), *The Economics of Information Technology*, Cambridge University Press.

Yalta, A. Talha and A. Yasemin Yalta (2010), "Should Economists Use Open Source Software for Doing Research?", *Computational Economics*, Vol. 35, No. 4, pp. 371-394.

Appendix: Dataset

Obs. no.	Country	Piracy rate (2010)	Windows share	Linux share	GDP (PPP)/capita, USD	OpenOffice user share	MS Office user share
1	Australia	24%	81.79%	0.66%	39406.6656595216	10.00%	78.00%
2	Bangladesh	90%	97.34%	0.38%	1555.3160689124		
3	Brunei	66%	81.88%	0.39%	49493.9323061487		
4	China	78%	94.46%	0.26%	6802.5023885724	0.20%	68.00%
5	Hong Kong	45%	83.97%	0.40%	43133.7261486081		
6	India	64%	95.94%	2.18%	3280.8856106271	5.00%	88.00%
7	Indonesia	87%	95.80%	0.89%	4049.3577076857		
8	Japan	20%	91.34%	0.18%	32005.5357654636		
9	Malaysia	56%	94.09%	0.40%	13705.9129403992		
10	New Zealand	22%	82.81%	1.00%	29258.0075029361		
11	Pakistan	84%	98.61%	0.45%	2584.3686879188		
12	Philippines	69%	94.81%	0.78%	3687.4263755813		
13	Singapore	34%	80.26%	0.97%	50650.2896746534		
14	South Korea	40%	93.74%	0.27%	27133.278269876		
15	Sri Lanka	86%	96.79%	1.30%	4665.0178361239		
16	Taiwan	37%	92.68%	0.62%	35604		
17	Thailand	73%	89.85%	0.46%	7848.0237816174		
18	Vietnam	83%	96.78%	0.48%	2982.0902492079		
19	Albania	75%	97.03%	0.60%	8596.2450462362		
20	Armenia	89%	95.77%	0.43%	5265.9678393484		
21	Azerbaijan	88%	96.76%	0.36%	9414.9045510577		
22	Belarus	88%	96.34%	1.35%	13029.9569839605		
23	Bosnia	66%	98.53%	0.49%	8537.4562846478		
24	Bulgaria	65%	94.99%	1.78%	13764.1750583894	7.00%	77.00%
25	Croatia	54%	96.81%	0.88%	20037.1461775017		
26	Czech Republic	36%	84.24%	9.17%	25572.278099978	22.00%	76.00%
27	Estonia	50%	92.75%	1.13%	19690.1539330833		
28	FYROM	66%	96.17%	1.66%	11097.93144916		
29	Georgia	93%	97.08%	0.37%	4732.8776761401		
30	Hungary	41%	97.78%	0.91%	20274.9684127986	11.00%	77.00%
31	Kazakhstan	76%	96.55%	0.39%	11429.2799285097		
32	Latvia	56%	92.32%	1.74%	16166.3904239417		
33	Lithuania	54%	93.62%	4.45%	17058.9238924864		
34	Moldova	90%	94.63%	1.90%	2856.1853444531		
35	Montenegro	79%			12898.7893989948		
36	Poland	54%	94.07%	1.93%	18925.2720005825	22.00%	68.00%
37	Romania	64%	96.85%	1.02%	14215.5608816863	8.00%	67.00%
38	Russia	65%	93.38%	2.18%	18877.7072676318	6.00%	76.00%
39	Serbia	74%			11471.852168878		
40	Slovakia	42%	96.17%	1.14%	22874.7332669447		
41	Slovenia	47%	92.72%	1.65%	27556.3601322207		
42	Ukraine	86%	97.81%	1.05%	6304.2454240273	7.00%	80.00%
43	Argentina	70%	94.75%	1.58%	14547.3311055868		
44	Bolivia	80%	94.27%	0.89%	4651.763681109		
45	Brazil	54%	92.63%	1.47%	10344.2199511292	8.00%	85.00%
46	Chile	62%	92.00%	1.03%	14297.7315895356		
47	Colombia	54%	94.98%	0.58%	9044.0774308052		
48	Costa Rica	58%	86.88%	2.00%	11025.0866554635		
49	Dominican Rep.	76%	89.46%	0.75%	8645.0227036466		
50	Ecuador	67%	93.43%	0.99%	7861.5083191116		
51	El Salvador	80%	94.05%	1.27%	6601.0236559647		
52	Guatemala	80%	88.87%	0.80%	4692.6512569245		
53	Honduras	73%	90.24%	0.86%	3831.8056823821		
54	Mexico	58%	91.12%	0.82%	13846.2988216552		
55	Nicaragua	79%	94.85%	0.68%	2657.8236979119		
56	Panama	72%	88.61%	0.70%	12994.6663941225		
57	Paraguay	83%	94.62%	2.01%	4506.2749751138		
58	Peru	68%	96.39%	0.58%	8715.6211145888		
59	Uruguay	69%	92.70%	3.46%	13083.724834527		
60	Venezuela	88%	94.25%	1.65%	12265.0724485877		
61	Algeria	83%	98.56%	0.51%	8121.3777616587		
62	Bahrain	54%	88.76%	0.22%	25799		
63	Botswana	79%	94.45%	0.59%	12902.8217910406		
64	Cameroon	82%	97.81%	0.80%	2234.1949612381		
65	Egypt	60%	96.29%	0.38%	6019.3216543738		

66	Iraq	85%			3577.5814999399		
67	Israel	31%	92.88%	0.93%	27493.7490497974		
68	Ivory Coast	79%	97.28%	0.37%	1849.0507925365		
69	Jordan	57%	95.18%	0.39%	5604.6196594784		
70	Kenya	79%	94.40%	2.10%	1578.670570082		
71	Kuwait	60%	88.50%	0.26%	52657		
72	Lebanon	72%	92.47%	0.18%	13007.7595022873		
73	Libya	88%	98.42%	0.33%	16836.9814947689		
74	Mauritius	56%	92.25%	0.95%	12986.4020337069		
75	Morocco	65%	96.26%	0.45%	4523.1270390084		
76	Nigeria	82%	97.92%	0.25%	2225.5412399611		
77	Oman	62%	96.50%	0.17%	26554.0307387328		
78	Qatar	49%	88.53%	0.15%	80228.7292305446		
79	Reunion	40%	86.38%	3.19%			
80	Saudi Arabia	52%	93.96%	0.34%	22035.1003731388		
81	Senegal	78%	95.74%	0.73%	1871.8692918872		
82	South Africa	35%	92.16%	1.13%	10237.2886137366		
83	Tunisia	72%	97.50%	0.46%	8228.0653278183		
84	Turkey	62%	95.08%	0.52%	14228.4671724282	5.00%	82.00%
85	UAE	36%	89.67%	0.17%	38089.1628450341		
86	Yemen	90%	98.74%	0.05%	2484.548207363		
87	Zambia	82%	94.53%	0.90%	1449.8880739199		
88	Zimbabwe	91%	91.34%	1.86%	418		
89	Canada	28%	84.55%	0.64%	37808.0201591482	11.00%	79.00%
90	Puerto Rico	42%	89.46%	0.67%	16300		
91	United States	20%	82.93%	2.99%	45744.559570303	9.00%	75.00%
92	Austria	24%	88.02%	1.53%	38803.6410985557	12.00%	85.00%
93	Belgium	25%	84.69%	1.44%	36278.5250919169	14.00%	85.00%
94	Cyprus	48%	95.87%	0.68%	30728.4533870101		
95	Denmark	26%	81.84%	1.12%	37672.3503581586	14.00%	79.00%
96	Finland	25%	82.64%	3.39%	35254.3609420391		
97	France	39%	83.87%	2.47%	33348.7775745294	19.00%	76.00%
98	Germany	27%	90.27%	1.63%	36319.6581914198	21.00%	72.00%
99	Greece	59%	96.43%	0.87%	29303.3432044188		
100	Iceland	49%	72.59%	1.13%	36732.6802268388		
101	Ireland	35%	89.14%	1.10%	39643.0679589962		
102	Italy	49%	84.62%	2.34%	32413.1887686866	18.00%	81.00%
103	Luxembourg	20%	84.31%	1.37%	84765.8554854562	12.00%	81.00%
104	Malta	43%	92.51%	0.60%	24803.90698571		
105	Netherlands	28%	84.94%	1.28%	40795.5545591057	8.00%	88.00%
106	Norway	29%	87.21%	0.78%	55717.4408627762	18.00%	71.00%
107	Portugal	40%	94.01%	0.78%	25058.3502329093		
108	Spain	43%	90.69%	1.84%	32261.5419767969	15.00%	80.00%
109	Sweden	25%	84.00%	1.41%	37156.8611251402	13.00%	68.00%
110	Switzerland	26%	77.74%	1.44%	45139.6546087782	11.00%	85.00%
111	United Kingdom	27%	88.93%	0.82%	35145.492345807	9.00%	80.00%

Regional dummies (=1 if belongs to following regions):
Asia Pacific: obs. 1-18, Eastern and Central Europe: obs. 19-42, Latin America: obs. 43-60, Middle East and Africa: obs. 61-88, North America: obs. 89-91, Western Europe: obs. 92-111.

Sources:
Piracy rates: 2010 Global Piracy Study, BSA.
User share OS: http://marketshare.hitslink.com, October 2011.
GDP PPP per capita 2010: The World Bank. Where data is missing I have taken from previous years: Bahrain (2008), Kuwait (2007).
Following observations are taken from other sources: Puerto Rico (2010 CIA World Factbook), Taiwan (2010 IMF), Zimbabwe (2010 IMF).
MS Office and OpenOffice.org user share:
http://www.webmasterpro.de/portal/news/2010/02/05/international-openoffice-market-shares.html