Open Source versus Proprietary Software: A Case of Fully Covered Market

Yipeng Liu[1] and Hsing K. Cheng[2]

[1] 320 Brennan Hall, Kania School of Management, University of Scranton
Scranton, PA, 18510, U. S. A.
[2] 361C Stuzin Hall, Warrington College of Business, University of Florida
Gainesville, FL, 32611, U. S. A.
yipeng.liu@scranton.edu, hkcheng@ufl.edu

Abstract. This paper analyzes the impact of network externalities on the competition between open source software (OSS) and proprietary software (PS) in a fully covered market. The installed base and the profit of proprietary software are found increasing at the expense of decreasing user base for OSS. Furthermore, we find that a threshold corresponding to the quality ratio between OSS and proprietary software can be derived such that if the network effect intensity of the OSS is greater than that of the proprietary software multiplied by this threshold value, then OSS benefits from the presence of network externality; Finally, we find that making software products compatible with competing rival is not desirable by proprietary software vendors but favored by OSS venders.

Keywords: Open source software, network externalities, competition, compatibility.

1 Introduction

As opposed to commercial proprietary software (PS), open source software (OSS) refer to those programs "whose licenses give users the freedom to run the program for any purpose, to study and modify the program, and to redistribute copies of either the original or the modified program without having to pay royalties to previous developers" [16]. Apache, OpenOffice, mySQL and Linux are some notable examples of OSS that have achieved remarkable success against their closed source or proprietary alternatives as is evident in their large user base.

Increasing number of OSS products have become available in the market, ranging from major operating systems, database systems, to thousands of specialized scientific applications. The increasing adoption of OSS at the same time poses a challenge for proprietary software vendors. For decades, commercial software makers charge consumers licensing fees of their programs for a steady flood of ever-cheaper computers. But now, the suppliers face a growing threat from "the open source", which gives customers a free or low-cost alternative to commercial products. "It's very hard to compete with free," says John Chen, chief executive of Sybase Inc., the No. 4 largest database provider. "It lowers the price point."

R. Sharman, T.S. Raghu, and H.R. Rao (Eds.): WEB 2009, LNBIP 52, pp. 279–288, 2010.

Several factors contribute to the lower total ownership cost of using OSS. For instance, “OSS costs less to initially acquire, upgrade and maintain, and runs more efficiently on older hardware, among others” [16]. Cyber source found that using OSS saves 24 to 34 percent of the total ownership cost over a three-year time span compared with using Microsoft’s proprietary approach for a typical organization of 250 computer-using employees (www.cybersource.com).

2 Literature Review

The successes of OSS have led to an increasing interest in understanding this form of software about its development and its impact on the software industry. Pioneering research on OSS to date focuses mainly on why developers participate in OSS projects (e.g., [7, 8, 11, 12]), the incentives of firms to adopt open source initiatives (e.g., [13]), and the success factors of OSS projects (e.g., [3, 4, 6, 15]). As OSS proves to be a viable technology solution for businesses, another stream of OSS research turns attention to the competition dynamics between OSS and proprietary software. This stream of research includes analysis – both empirical and theoretical – of the public and free nature of OSS products and their impact on the marketplace for software products. However, the majority of these studies either focus primarily on the result of competition between particular operating systems (e.g., [2]) or consider open source to be privately provided public goods and concentrate on software vendor’s choice between producing open source or proprietary software (e.g., [1, 9]). Furthermore there are few studies that analyze the extent of this competition under the presence of network externalities.

Our paper aims at a similar topic but differs from the past literature in two major aspects. First, our work focuses on the impact of network externalities on the competition rather than the result of competition between open source and proprietary software. We model the heterogeneous consumers’ preferences for products to depend on two major factors – the software quality and the network size, and we allow the extent of network effect to be different between OSS and proprietary software. Second, we investigate the strategic choices and incentives for compatibility between OSS and proprietary software venders – a topic majorly overlooked in previous studies on open source software.

Table 1. Four different cases of analyses

	OSS and Proprietary Software are compatible	OSS and Proprietary Software are incompatible
Both OSS and Proprietary Software exhibit same network effect (γ)	Case 1	Case 3
OSS and Proprietary Software exhibit different network effect (γ_o & γ_p)	Case 2	Case 4

We use a purely analytical approach to study the competition between open source and proprietary software with focus on the following research questions: Does the positive network effect always yield positive impact? How would the choice of compatibility affect network externality's impact? Furthermore, which software vendor (PS or OSS) has the most incentive to make its product compatible? We address these research questions by separating our analysis into four different cases (see Table 1) depending on whether the rivalry software are compatible with each other and whether they share the same network effect intensity.

3 The Model

Assume that the total number of potential customers who will adopt either open source software (OSS) or proprietary software is N, which is normalized to 1 without loss of generality. Let θ represent a customer's preference for software quality and θ is uniformly distributed between 0 and 1. In the absence of network externalities, a customer derives:

$$u = \theta \cdot s - p$$

utility of adopting the software of quality s charging price p.

Let γ model the intensity of the network externalities effect. When Q is the installed base of the software (i.e., the total number of users adopting the software), network externalities effect increases each consumer's preference θ by γQ in accordance with the increase in willingness to pay when an additional user joins the network. When purchasing a software product of quality s in a network of size Q at price p, consumer θ obtains net utility:

$$u = (\theta + \gamma Q)\, s - p$$

under the influence of network externalities. In [5], $\theta \cdot s$ is the product's "network-independent" or standalone value, which depends on both consumer's preference and product quality. The "network-generated" value is γQs and exhibits complementary effects between network size and quality. Therefore in the same network, a high quality product has both higher standalone and network values than a lower quality one. In [10], and most other research on network externalities, product are homogeneous and consumers differ in their total willingness to pay. The network-generated benefit therefore only depends on the network size.

Let s_o and s_p describe respectively the quality of open source software and proprietary software. Generally, software quality can be measured in several dimensions, such as performance, ease of use, customer support, reliability and security, etc. For the sake of generality, our study is not restricted to any particular dimension. Rather, we define software quality as the characteristics of software other than the price. For practical purpose, we assume $s_p > s_o$, since all customers will adopt OSS if the quality of the free OSS exceeds that of the commercial counterpart. In reality, we indeed observe certain quality advantages enjoyed by the proprietary software, such as more user-friendly interface, more reliable and professional customer support, continuous product upgrade and so forth. Let Q_o and Q_p denote the installed user base of open source software and proprietary software respectively.

4 Competition between OSS and Proprietary Software

We first present the case where there is no network externalities effect in Section 4.1 as a benchmark, followed by analyses of competition between OSS and proprietary software in the presence of network externalities under different scenarios. Several key propositions are derived to examine the impact of network externalities on the competition.

4.1 Benchmark Case – No Network Externalities

In the benchmark model (see Figure 1), we assume that the market is fully covered, i.e., all consumers choose to use one of the two software products. This is always true when the benefit of the product is sufficiently large. In addition, we assume that the open source software is freely available hence there is no price component in the net utility. Without network effect, a consumer of type θ derives net utility

$$u_o = \theta \cdot s_o$$

by using the open source software and the net utility of buying the proprietary software is

$$u_p = \theta \cdot s_p - p .$$

Suppose consumer $\theta_b \in [0,1]$ is indifferent between the open source and proprietary software products, then one has

$$\theta_b s_p - p = \theta_b s_o \ .$$

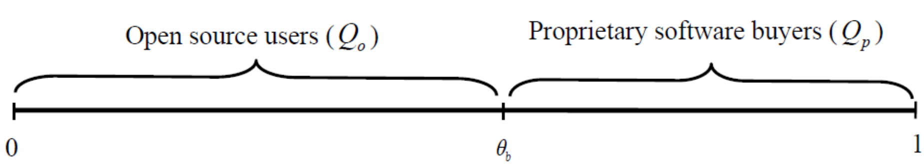

Fig. 1. Demand of Open Source and Proprietary Software

Thus, $\theta_b = p / \left(s_p - s_o\right)$ and the resulting installed bases for OSS and proprietary software are θ_b and $1-\theta_b$ respectively. In contrast with conventional Hotelling model, only one party—the proprietary software producer aims at maximizing profit and the open source software vender is passive. Due to the negligible marginal cost of production, proprietary software firm solves the following profit maximization problem:

$$\max_{p} p(1-\theta_b) \quad s.t. \; 0 \le p \le s_p - s_o .$$

Accordingly, the optimal price of the proprietary software in the benchmark case is,

$$p_b^* = \left(s_p - s_o\right)/2 ,$$

and the demands of proprietary and open source software and the profit of the proprietary software vender are given by

$$Q_{bp}^* = Q_{bo}^* = \frac{1}{2}$$

$$\pi_b^* = \left(s_p - s_o\right)/4$$

where the subscript b denotes the benchmark case of no network externalities, p for proprietary software, and o for OSS. Even though proprietary software is of higher quality, since OSS is free, the benchmark-level demands (e.g. installed user base) of proprietary and open source software are the same.

4.2 The Impact of Network Externality

We first present how consumers' utility functions of the open source and proprietary software change according to different network effect intensities and compatibility strategies. Next, we summarize the impact of network externalities by comparing the demands of OSS and proprietary software with the benchmark demands reported in Section 4.1.

4.3 Case 1: OSS and Proprietary Software Are Compatible and Exhibit Same Network Externality Effect

When the rivalry products are compatible with each other, the network size for either software product is equivalent to the sum of the two networks (i.e. $Q = Q_o + Q_p = 1$, since the market is fully covered). In addition, both OSS and proprietary software exhibit the same network effect intensity in Case 1 (i.e., $\gamma_o = \gamma_p = \gamma$). Therefore, in the presence of direct network effect, the net utility for OSS and proprietary software consumers are given by:

$$u_o = \left(\theta + \gamma Q\right) s_o \text{ and } u_p = \left(\theta + \gamma Q\right) s_p - p \text{, where } Q = 1 .$$

Marginal consumer θ_o who is indifferent between open and proprietary software is characterized by

$$\left(\theta_o + \gamma Q\right) s_o = \left(\theta_o + \gamma Q\right) s_p - p \ .$$

The demands of open source and the proprietary software are therefore defined as:

$$Q_o = \theta_o \text{ and } Q_p = 1 - \theta_o \ .$$

Simple algebra indicates that:

$$\theta_o = \left(\left(s_o - s_p\right)\gamma + p\right) / \left(s_p - s_o\right) \ .$$

Accordingly, the number of proprietary software buyers is:

$$Q_p = \left(\left(s_p - s_o\right)\left(1+\gamma\right) - p\right) / \left(s_p - s_o\right) \ .$$

In order to guarantee non-negative demand for both OSS and proprietary software, the following condition must hold

$$\gamma\left(s_p - s_o\right) \le p \le \left(1+\gamma\right)\left(s_p - s_o\right) \ .$$

Proprietary software vender maximizes its revenue by setting the price optimally. The decision problem of the proprietary software vender is therefore formulated as:

$$\begin{aligned} &\max_p \ pQ_p = p \cdot \frac{\left(s_p - s_o\right)\left(1+\gamma\right) - p}{s_p - s_o} \\ &\text{s.t. } \gamma\left(s_p - s_o\right) \le p \le \left(1+\gamma\right)\left(s_p - s_o\right) \end{aligned} \ .$$

Kuhn-Tucker conditions lead to:

$$p^* = \frac{\left(s_p - s_o\right)\left(1+\gamma\right)}{2}, \ Q_o^* = \frac{1-\gamma}{2}, \ Q_p^* = \frac{1+\gamma}{2}, \text{ and } \pi_p^* = \frac{\left(s_p - s_o\right)\left(1+\gamma\right)^2}{4}.$$

By comparing these optimal results to the benchmark level demands and profit, we have the following proposition.

Proposition 1. *In Case 1, the installed base of proprietary software increases, the installed base of open source software decreases, both the optimal price and the optimal profit of proprietary software increase in the presence of network externalities.*

4.4 When the Network Effect Intensity Is Different

Unlike past literature, our model allows the possibility that the OSS has higher network effect intensity than the proprietary software. To address this issue (i.e., Cases 2 and 4 in Table 1), we use different parameters γ_o and γ_p ($\gamma_o > \gamma_p$) to capture the different network effect intensities for OSS and proprietary software respectively.

4.5 When Software Products Are Not Compatible

Users in OSS network will not be able to benefit from the contribution of the users in the proprietary software network and vice versa. As a result, the network size, from which the network value is generated, shrinks for both software products. To address this issue (Cases 3 and 4 in Table 1), we use Q_o and Q_p to denote the installed base of OSS and proprietary software respectively. Full market coverage leads to $Q_o + Q_p = 1$. Following the same approach in Case 1, Table 2 reports the optimal outcomes of the open source and proprietary software under different scenarios. A series of key propositions are thus derived from Table 2.

Table 2. Fully covered market: optimal outcomes

	Case 1	Case 2	Case 3	Case 4
u_o	$(\theta+\gamma Q)s_o$	$(\theta+\gamma_o Q)s_o$	$(\theta+\gamma Q_o)s_o$	$(\theta+\gamma_o Q_o)s_o$
u_p	$(\theta+\gamma Q)s_p - p$	$(\theta+\gamma_p Q)s_p - p$	$(\theta+\gamma Q_p)s_p - p$	$(\theta+\gamma_p Q_p)s_p - p$
p^*	$\frac{(s_p - s_o)(1+\gamma)}{2}$	$\frac{s_p(1+\gamma_p) - s_o(1+\gamma_o)}{2}$	$\frac{s_p - s_o(1+\gamma)}{2}$	$\frac{s_p - s_o(1+\gamma_o)}{2}$
Q_o^*	$\frac{(1-\gamma)}{2}$	$\frac{s_p - s_o + s_o\gamma_o - s_p\gamma_p}{2(s_p - s_o)}$	$\frac{s_p - s_o - s_o\gamma - 2s_p\gamma}{2(s_p(1-\gamma) - s_o(1+\gamma))}$	$\frac{s_p - s_o - s_o\gamma_o - 2s_p\gamma_p}{2(s_p(1-\gamma_p) - s_o(1+\gamma_o))}$
Q_p^*	$\frac{(1+\gamma)}{2}$	$\frac{s_p - s_o + s_p\gamma_p - s_o\gamma_o}{2(s_p - s_o)}$	$\frac{s_p - s_o - s_o\gamma}{2(s_p(1-\gamma) - s_o(1+\gamma))}$	$\frac{s_p - s_o - s_o\gamma_o}{2(s_p(1-\gamma_p) - s_o(1+\gamma_o))}$
π^*	$\frac{(s_p - s_o)(1+\gamma)^2}{4}$	$\frac{(s_o(1+\gamma_o) - s_p(1+\gamma_p))^2}{4(s_p - s_o)}$	$\frac{(s_o - s_p + s_o\gamma)^2}{4(s_p(1-\gamma) - s_o(1+\gamma))}$	$\frac{(s_o - s_p + s_o\gamma_o)^2}{4(s_p(1-\gamma_p) - s_o(1+\gamma_o))}$
Valid γ range	$0 \le \gamma \le 1$	$0 \le \frac{1+\gamma_o}{1+\gamma_p} \le \frac{s_p}{s_o}$	$0 \le \gamma \le \frac{s_p - s_o}{2s_p + s_o}$	$0 \le \frac{1+\gamma_o}{1-\gamma_p} \le \frac{s_p}{s_o}$

4.6 The Effect of Network Externalities – Positive or Negative

By comparing the optimal outcomes of OSS and PS under Cases 3 and 4 with the benchmark results, we find that network externalities have positive impact on the demand and profit of proprietary software, but negative impact on the demand of open source software.

Proposition 2. *When OSS is incompatible with its proprietary software counterpart (Case 3 and 4), the installed base of open source software decrease, the installed base of proprietary software increases, the optimal price of proprietary software drops,* *but the optimal profit of proprietary software improves in the presence of network externalities.*

Comparing the optimal outcomes of Case 2 with benchmark results, however, shows mixed results. We find that if the ratio of network effect intensities between OSS and proprietary software (i.e. γ_o / γ_p) falls short of the inverse quality ratio between the two (i.e. s_p / s_o), then proprietary software benefits from the presence of network externalities; otherwise, OSS is favored by more users in the presence of network externalities. Our findings can be summarized by the proposition below.

Proposition 3. *Define the threshold value* $T_1 = \frac{s_p}{s_o}$. *In case 2, When* $\gamma_o < T_1\gamma_p$ *, the installed base, the optimal price and the overall profit for proprietary software all increase and the installed base of OSS decreases in the presence of network externalities. The results are reversed when* $\gamma_o > T_1\gamma_p$.

4.7 What Is the Best Strategy – Compatible or Incompatible?

For either OSS or proprietary software vender, making its product compatible to its competitor allows the user of that product benefit from the network of its opponent, which therefore adds to the value of the product and increases consumer's willingness to pay and their net utility of adopting the software. However, in a fully covered market, the increase of one party's user leads to the decrease of the other party's user and the best response for the competitor in such a situation is to make its product compatible too. Therefore in equilibrium we will see software products are either incompatible with each other or fully compatible with each other. One-way compatibility (one product compatible with the other, but not the other way around) is therefore ignored in our analysis. Furthermore, we are more interested in the question about which party has the most incentive to make its product compatible. The answer to this question can be found by a head to head comparison (e.g. Case 3 vs. Case 1, Case 4 vs. Case 2) between the payoffs (e.g. market shares and profits) for both OSS and proprietary software venders when the software is made from incompatible to compatible with its rival product.

Proposition 4. *Whether OSS and proprietary software exhibit the same network effect intensity or not, OSS vender always has the most incentive to make its product compatible with its proprietary counterpart, as manifested by the expanded market share after the change. Proprietary software vender, on the other hand, is not willing to make its product compatible with the OSS as doing so will decreases its market share and profit.*

5 Conclusion

We investigate the impact of network externalities on the competition between open source and proprietary software. We analyze four different scenarios depending on whether OSS and proprietary software are compatible with each other and whether they exhibit the same network effect intensities. When the market is fully covered, network effect benefits the proprietary software more than OSS in all cases except Case 2. Compared with the benchmark case of no network externalities, the installed base and the profit of proprietary software both increase at the expense of OSS losing user base in the presence of network externalities. In Case 2, we find that a threshold corresponding to the quality ratio between OSS and proprietary software can be derived such that if the network effect intensity of the OSS is greater than that of the proprietary software multiplied by this threshold value, then OSS benefits from the presence of network externality; otherwise, proprietary software benefits from the presence of network effect. Finally, we find that making software products compatible with competing rival is not desirable by proprietary software vendors but highly favored by OSS venders. Hence, in order to maintain its dominant position in market shares, the proprietary software vendor should make it hard for the OSS software to be compatible with its product.

References

1. Bessen, J.: What good is free software?. In: Hahn, R. (ed.) Government Policy Towards Open Source Software. AEI Brookings Joint Center for Regulatory Studies, Washington (2002)
2. Casadesus-Masanell, R., Ghemawat, P.: Dynamic mixed duopoly: a model motivated by Linux vs. Windows. Management Science 52(7), 1072–1084 (2006)
3. Chengalur-Smith, S., Sidorova, A.: Survival of open-source projects: a popular ecology perspective. In: Proceedings of the International Conference on Information Systems, pp. 782–787 (2003)
4. Crowston, K., Annabi, H., Howison, J.: Defining open source software project success. In: Proceedings of the International Conference on Information Systems, pp. 329–340 (2003)
5. Farrell, J., Saloner, G.: Installed base and compatibility: innovation, product preannouncements, and predation. American Economic Review 76(5), 940–955 (1986)
6. Fitzgerald, B., Kenny, T.: Open source software in the trenches: lessons from a large-scale OSS implementation. In: Proceedings of the International Conference on Information Systems, pp. 316–326 (2003)
7. Hann, I.-H., Roberts, J., Slaughter, S.A., Fielding, R.: Economic incentives for participate in open source software projects. In: Proceedings of the International Conference on Information Systems, pp. 365–372 (2002)
8. Hann, I.-H., Roberts, J., Slaughter, S.A.: Why developers participate in open source software projects: an empirical investigation. In: Proceedings of the International Conference on Information Systems, pp. 821–830 (2004)
9. Johnson, J.P.: Economics of open source software. Journal of Economics and Management Strategy 11, 627–662 (2002)
10. Katz, M., Shapiro, C.: Network externalities, competition and compatibility. American Economic Review 75, 420–440 (1985)
11. Lerner, J., Tirole, J.: The open source movement: key research questions. European Economic Review 45, 819–826 (2001)
12. Lerner, J., Tirole, J.: Some simple economics of open source. Journal of Industrial Economics 50, 197–234 (2002)

13. Mustonen, M.: Copyleft – the economics of Linux and other open source software. Information Economic Policy 15, 99–121 (2003)
14. Rosenberg, D.K.: Open source: The unauthorized white papers. IDG Book Worldwide, Inc. (2000)
15. Sagers, G.W.: The Influence of network governance factors on success in open source software development projects. In: Proceedings of the International Conference on Information Systems, pp. 427–438 (2004)
16. Wheeler, D.A.: Why open source software/free software (OSS/FS)? Look at the number! (2003), `http://www.dwheeler.com/oss_fs_why.html`