Implications of Negative Network Effect in the Security Software Market

Debabrata Dey
Foster School of Business
University of Washington
ddey@uw.edu

Atanu Lahiri
Foster School of Business
University of Washington
lahiria@uw.edu

Guoying Zhang
Dillard College of Business
Midwestern State University
grace.zhang@mwsu.edu

Abstract
The market for security software has witnessed an unprecedented growth in recent years. A closer examination of this market reveals certain idiosyncrasies that are not observed in a traditional software market. For example, it is a highly competitive market involving many vendors, often with a very aggressive pricing strategy adopted by new entrants. Yet, the market coverage seems to be quite low. Prior research has not attempted to explain what aspects of security software make this market deviate from the traditional ones. In this paper, we develop a quantitative model to study this market. Our model identifies a possible reason behind this behavior—that of a negative network effect, which pulls the market in exactly the opposite direction when compared to the positive network effect observed in traditional software markets. Overall, our results highlight the unique nature of the security software market, furnish rigorous explanation for several counter-intuitive observations in the real world, and provide managerial insights for vendors on market competition and product development strategies. Keywords: Security software, network effect, market structure, pricing

1. Introduction
Ever since information systems became the underlying infrastructure for modern businesses, the security of information systems has been recognized as one of the most important aspects in assuring business continuity and protecting valuable information assets. As a result, the industry of security software, along with that of security hardware and security services, has grown rapidly in response to the continually higher demand for the protection of an ever-increasing base of information technology (IT) infrastructure. Figure 1, based on Gartner reports, shows that the worldwide security software revenue has increased quite rapidly from US$6.4 billion in 2004 to about US$16.6 billion in 2010; and even a worldwide recession in 2009 did not much slow this market down. The rapid growth has far exceeded all predictions about this market. IDC reports that the security software market in Asia (excluding Japan) alone has demonstrated about 23% growth in 2005 and is keeping pace with the rest of the world in its growth. Security software market thus has been regarded as one of a few prominent software markets with double-digit growth rate. Understanding the nature of this market, along with its evolution and trend, is of importance to vendors as well as consumers.

In general, security software can be classified into two main categories: (i) third-party standalone tools, such as antivirus software and spyware remover, and (ii) software components that are often bundled with the operating system, such as encryption software and firewall. In this paper, we examine the former market.

In the typical off-the-shelf software market, users usually enjoy a higher network utility derived from a larger market share, which is often referred to as the positive network effect [15, 16]. This positive network effect primarily arises from users’ need for compatibility—the need to share files and information, to edit and critique documents created by others, and, most importantly, to work in a collaborative setting. It is well-known that positive network effect can lead to a near-monopolistic market condition: if a vendor’s market share is large enough to exceed a critical mass, other competitors will lose opportunities to enter the market. Studies on the markets for off-the-shelf application software (such as Spreadsheet and Word processor) empirically validate this near-monopolistic structure [6, 19]. However, the market for off-the-shelf security software is markedly different. It is characterized by many vendors, with no single dominant player. For example, the market of antivirus software has several
major players including Symantec, McAfee, Trend Micro, and Computer Associates, besides dozens of other smaller companies. Symantec, which once led this market with more than 50% of the sales, currently holds less than 20% of the market share. In fact, the total market share of the top 6 antivirus software vendors is well below 50%. Many of the current antivirus products in the market did not exist even a couple of years ago. Similar trend is observed with other security software products as well—Fosfuri and Giarratana[8] found that, between 1989 and 1998, at least 270 vendors entered this market, with a very high percentage not surviving beyond two years [11]. At the same time, despite the large number of existing security software vendors, a very high percentage of information systems and individual computers are still lacking in basic protection [13]. These observations naturally lead to several questions:

- Why are there so many vendors in the security software market?
- What makes an incumbent like Symantec lose market share so steadily?
- Why is the market coverage still low?

Despite being so competitive and turbulent, what makes this market so attractive to new entrants?

- Why does a new entrant price aggressively, to the extent that it may be deemed as predatory?

The objective of this research is to develop a quantitative model to find possible answers to these questions and obtain useful insights about the market.

It is difficult to explain the competitiveness observed in this market under the assumption of a positive network effect. In fact, from the perspective of a user of a security software, there is little additional utility to be derived from the market share of the product. A security software is simply used to prevent security exploitations, and there is hardly any benefit from the compatibility of user data. Therefore, the market of security software does not exhibit as strong a positive network effect as other software markets do.2

We should point out that, while compatibility concern is the primary force behind the positive network effect in off-the-shelf software markets [10], there may well be other contributing factors for this effect. For example, a larger market share of a product may provide additional utility to a user in terms of better product support, availability of helpful tips and resources (such as end-user forums), and better maintenance (in terms of more frequent updates and upgrades). Since these are all applicable to a security software, there may indeed be some positive network effect in this market as well. However, since the major factor (compatibility) is missing in this context, the positive network effect is significantly weaker here, when compared to the typical software market.

1 Just to name a few, consider products such as Avast, Sophos, AVG, Bitdefender, Kaspersky, Panda, Avira, G-Data, F-Secure, Webroot, ESET, Lavasoft, Vipre, PCTools, SystemShield, SpySweeper, SystemTech, SystemWorks, Virex, Iolo, Forefront, Avira, G-DATA, Kaseya, Ad-Aware, Outpost, Digital Defender, Immunet Protect, Zone Alarm, and VCatch.
Instead, our analysis incorporates a negative network effect similar to that found in recent literature [3, 4, 5]. In our setting, when a user adopts a security software, there are two benefits: (i) a direct benefit—representing the value of thwarting direct security attacks by hackers, and (ii) an indirect benefit—arising from the prevention of indirect attack or infection from other users in the network [21]. In an indirect attack, a system is not a direct target, but could become an eventual target from the security exploitation of another system. Typical examples of indirect attacks include the prevalence of Internet worms and the wide presence of botnet agents, which could launch large-scale attack with the ability to convert ordinary nodes into malicious agents. A recent report finds that most of the well-known botnets involve several million bots or compromised computers—for example, the Breddolab and Conflicker botnets are supposed to have about 30 and 13 million infected machines, respectively [12]. Each month, hundreds of thousands of new machines are being added to some botnet. The total annual loss from this type of indirect attacks is currently estimated at a whopping US$10 billion [12]. The user’s indirect benefit eventually leads to a negative network effect—the larger the total market coverage of security software, the less is the indirect benefit because the indirect threats are already mitigated, and the chance of getting infected from others reduces. Such indirect effects have also been recognized by Anderson [2] as the “tragedy of commons,” by Png et al. [22] as the “the reason of users’ inertia of taking security precautions,” and by August and Tunca [3, 4] as an important factor in changing the users’ incentive to apply security patches. Incorporation of this diminishing indirect benefit into our model leads to an increasingly less network valuation by users from a larger market coverage. We find that this negative network effect provides an explanation for the unique nature of the security software market.

We examine different market situations. We start by analyzing a monopoly market. We show that a monopoly security software market has lower coverage, higher price, and higher revenue compared to a traditional monopoly market (without the negative network effect). We then extend our analysis to an oligopoly competition. There is a unique symmetric equilibrium in the oligopoly market, and, as the negative network effect increases, it leads to a higher price and a lower market coverage. This market is found to have more competitors in equilibrium when compared to a traditional market. We also study the effect of a new entry into the oligopoly market, and find that a new entrant is likely to adopt a more aggressive pricing strategy in this market than in a traditional one. Overall, our results highlight the unique nature of the security software market, furnish explanation for several counter-intuitive observations in the real world, and provide managerial insights for vendors on market competition and product development strategies.

The rest of the paper proceeds as follows. Section 2 develops the consumer model and examines the market behavior under monopoly and oligopoly settings. Section 3 evaluates the market structure under the influence of negative network effect and tries to explain the real-world observations about this market. Section 4 concludes the paper and offers future research directions.

2. The Model

Consumers (users) of security software are heterogeneous because the amount of benefit from thwarting an attack would vary from user to user. In order to capture this, consumers are indexed by a parameter $u$ that indicates their relative expected benefit if an attack is thwarted; we assume that $u$ is uniformly distributed over the interval $[0, 1]$. The absolute expected benefit to user $u$ from thwarting an attack can then be expressed as $Lu$, where $L$ is a constant; $Lu$ can also be viewed as a proxy for the potential loss to user $u$ from an attack [13].

As mentioned in Section 1, there are two types of benefits derived from adopting a security software—direct and indirect. First, consider the direct benefit. Assume that hackers could launch successful attacks on an unprotected system at an average rate of $\lambda_D$. Therefore, by adopting a security software, user $u$ has a direct mitigation benefit rate of $\lambda_D Lu$.

Next, we consider the indirect benefit. Given the current level of Internet adoption and the increasing affordability of the broadband technology, users’ computers are considered to be interconnected. Therefore, unprotected systems might replicate malicious codes and pass them to connected peers. At times, a hacker may attack a system indirectly, after first breaching the security of several other systems and using them as intermediate nodes to launch the attack. In other words, the existence of security software in one system can, indirectly, reduce attacks to others. Let $x$ be the fraction of users who have adopted security software. Then, an indirect attack is possible from the $(1 - x)$ unprotected fraction of users, so we model the rate of indirect attack as $\lambda_I(1 - x)$, where $\lambda_I$ is a base rate of
indirect attack (when no user is protected). Therefore, a user adopting a security software avoids indirect attacks from the unprotected users and derives an indirect utility of $\lambda_I(1 - x)L_u$. It is now obvious that a larger market share (larger $x$) leads to a reduction in this indirect utility. At the extreme, if all the users are equipped with security software, no user derives an indirect benefit from adopting the security software. This is similar to the free riding behavior in network systems and the feature of public goods in economics [2, 22].

The total benefit (per unit time) to user $u$ from adopting the software, in a market with coverage $x$, can then be written as:

$$B_u = \lambda_D L_u + \lambda_I (1 - x) L_u = \lambda_D L_u (1 + g(1 - x)),$$

where $g = \lambda_I / \lambda_D$. Clearly, the parameter $g$ is a proxy for the negative network effect—the larger the $g$, the larger is the potential indirect benefit and, hence, the more significant is the negative network effect. Writing the above expression in this form provides us with the flexibility to easily capture various levels of the relative indirect utility, which can be attributed to software characteristics as well as the network connectivity. For example, antivirus and anti-spyware software have a higher indirect effect and hence a higher $g$, whereas an encryption software might have a lower $g$. A well-connected network is likely to have a higher value of $g$, when compared to a sparser network.

Two points are worth mentioning here. First, not all attacks result in security breaches. At the same time, a security software may not be fully effective in thwarting an attack. These considerations may lead one to suspect that the benefit of a security software to a user is overestimated by $B_u$ above. However, this issue does not impact our modeling choice because both $\lambda_D$ and $\lambda_I$ can be suitably scaled to reflect this situation. As a second related point, individual users or organizations may deploy complementary technologies that may have an impact on a user’s benefit from a security software. For example, if firewalls or intrusion prevention systems are widely used, they may reduce the effective attack rates and, therefore, reduce the overall valuation of an antivirus software. Again, this issue can be addressed by similar scaling as above.

Security software products are usually licensed as a subscription for a year. Upon expiration, the user must renew the license to continue getting the service. Let $P$ be the subscription price (per unit time). A user would adopt a security software if the total benefit from the software is larger than its subscription price: $B_u \geq P$. The marginal user $u$ who is indifferent between adopting and not adopting the security software must then satisfy the following condition:

$$\lambda_D L_u (1 + g(1 - x)) - P = 0.$$

As shown in Figure 2, any user to the right of this marginal user adopts the software, whereas anyone to the left does not. Therefore, $u = 1 - x$. Substituting this and letting $p = \frac{P}{\lambda_D L_u}$, we get:

$$p = (1 + g(1 - x))(1 - x). \quad (1)$$

In other words, $p$ in Equation (1) represents the normalized price associated with a market coverage of $x$. For the rest of the paper, we will use this normalized price, with appropriate subscripts, as necessary.

### 2.1. Monopoly Market

Existing literature on the economics of software and information goods often view a software market as a natural monopoly. We, therefore, start our analysis with the monopoly market. Let the monopoly

---

3 The assumption that the rate of indirect attacks is proportional to the fraction of vulnerable nodes in the network is quite similar to the assumptions made by August and Tunca [3, 4, 5].

4 We should point out that, in this analysis, we are using the concept of fulfilled expected equilibrium [15]—the user makes the adoption decision based on an expected value of the market size, and the realized market size in equilibrium equals this expected value.
market coverage be \( x_{\text{mon}} \); from Equation (1), the corresponding normalized price is given by:

\[
p_{\text{mon}} = (1 + g(1 - x_{\text{mon}}))(1 - x_{\text{mon}}).
\]

(2)

Therefore, with a zero marginal cost for each user’s subscription, the monopolist’s objective is to select the optimal market coverage to maximize her revenue. More specifically, the monopolist solves the following optimization problem:

\[
\max_{x_{\text{mon}}} R_{\text{mon}} = p_{\text{mon}} x_{\text{mon}}; \ 0 \leq x_{\text{mon}} \leq 1.
\]

(3)

Solving (3), we get the following result:

**Proposition 1.** In a monopoly market of security software, the optimal market share and price are:

\[
x^*_{\text{mon}} = \frac{(2g + 1) - \sqrt{g^2 + g + 1}}{3g}, \quad \text{and}
\]

\[
p^*_{\text{mon}} = \frac{(2g^2 + 2g - 1) + (2g + 1)\sqrt{g^2 + g + 1}}{9g}. \quad (5)
\]

It can be shown that the results in Proposition 1 converge to those in a traditional monopoly market, i.e., \( \lim_{g \to 0} x^*_{\text{mon}} = \frac{1}{2} \), \( \lim_{g \to 0} p^*_{\text{mon}} = \frac{1}{2} \), and \( \lim_{g \to 0} R_{\text{mon}}(x^*_{\text{mon}}) = \frac{1}{4} \).

**Corollary 1.** A monopoly security software market has a smaller market coverage, higher price, and higher revenue, when compared to those in a traditional monopoly market.

In other words, the negative network effect makes the market more profitable, despite the free-rider problem. We now turn our attention to an oligopoly market, which is more relevant to the context of security software, and the issue of new entry.

### 2.2. Oligopoly Market

We apply the concept of fulfilled expected Cournot equilibrium [15] to study the oligopoly competition. We use the Cournot competition to model the relatively long-term decisions by vendors and use the market size as the decision variable for vendors. Cournot competition fits this context a lot better than Bertrand competition. A large majority of subscriptions to security software happens through preloading contracts with computer manufacturers. All leading security software vendors engage in such long-term contractual agreements with them. Some of these contracts are exclusive, while others are not. For example, Dell has a non-exclusive contract with Symantec, McAfee, and Trend Micro to preload their antivirus software on all the computers they sell to consumers. On the other hand, HP currently has an exclusive contract with Symantec. Consumers get a free trial period for the preloaded software, but must purchase the subscription if they wish to continue using it beyond the trial period. Engaging in this kind of long-term preloading contracts means that a vendor plans how many subscriptions it intends to sell, since the price she is willing to pay for the preloading contract is contingent on the expected sales. As shown by Kreps and Scheinerman [18], when vendors plan for a certain quantity in the first stage, a Bertrand-like price competition would still lead to a Cournot equilibrium, even if the marginal production cost is zero.

It is also instructive to see why the alternative, Bertrand competition, does not work in this setting. First, a pure Bertrand would drive the price down to zero—not something that is observed in practice. Second, a differentiated Bertrand—the circular city or the linear city models—would require a complete coverage of the market for real competition to set in. As mentioned earlier, assuming complete coverage would be contrary to real-world observations.

Suppose that, in equilibrium, there are \( n \) identical vendors in the market with non-negative revenue. The aggregate market size is \( M \), where \( M_1 = \sum_{i=1}^n x_i \), and \( x_i \) is vendor \( i \)'s market size. We define \( \sum_{i=1}^n x_i \) as the total market size of all vendors except that of vendor \( i \), i.e., \( \sum_{i=1}^{n-1} x_i = M - x_i \). Extending Equation (1) for a total market coverage of \( M \), we find that the price in this case would be \( (1 + g(1 - M))(1 - M) \), which is the valuation of the marginal user indifferent between adopting and not adopting security software. For vendor \( i \), the revenue maximization problem can, therefore, be formulated as:

\[
\max_{x_i} R_{\text{olig}} = \left( 1 + g \left( 1 - \sum_{i=1}^n x_{i-1} - x_i \right) \right) \left( 1 - \sum_{i=1}^n x_{i-1} - x_i \right) x_i; \quad 0 \leq \sum_{i=1}^n x_i \leq 1.
\]

We can solve this to obtain the following result:

**Proposition 2.** In an oligopoly market of security software with \( n \) identical vendors, the equilibrium market size and price for each vendor are given by:

\[
x^*_{\text{olig}} = \frac{(2g + 1)(1 + n) - G}{2gn(2 + n)}, \quad \text{and}
\]

\[
p^*_{\text{olig}} = \frac{4g(g + 1) - (1 + n) + (2g + 1)G}{2g(2 + n)^2}, \quad (7)
\]

where \( G = \sqrt{4g(g + 1) + (1 + n)^2} \).
It can be shown through algebraic manipulations that the results in Proposition 2 converge to those in a traditional oligopoly market: \( \lim_{y \to 0} x_{\text{olig}}^* = \frac{\lambda_{d}}{(n + 1)} \), \( \lim_{y \to 0} p_{\text{olig}}^* = \frac{1}{(n + 1)} \), and \( \lim_{y \to 0} R_{\text{olig}}(x_{\text{olig}}^*) = \frac{1}{(n + 1)} \). 

**Corollary 2.** The oligopoly market has the following characteristics:

\[
\frac{\partial x_{\text{olig}}^*}{\partial y} \leq 0, \quad \frac{\partial x_{\text{olig}}^*}{\partial n} \leq 0, \quad \frac{\partial p_{\text{olig}}^*}{\partial y} \geq 0, \quad \text{and} \quad \frac{\partial p_{\text{olig}}^*}{\partial n} \leq 0.
\]

Corollary 2 indicates that a higher negative network effect (i.e., a larger \( g \)) leads to a smaller market coverage and a higher price. Also, as expected, the market coverage and price are decreasing with the number of vendors in the market. Next, we consider two important issues for this market—the competition level in the market and the effect of a new entry on price.

### 3. Market Structure

#### 3.1. Level of Competition

As mentioned earlier, the security software market exhibits a higher level of competition when compared to other off-the-shelf software markets. Based on our model, we now investigate why this is the case. Before we proceed, the following result is necessary:

**Lemma 1.** The revenue for a vendor in an oligopoly market of security software has the following characteristics: \( \frac{\partial R_{\text{olig}}(x_{\text{olig}}^*)}{\partial y} \geq 0 \) and \( \frac{\partial R_{\text{olig}}(x_{\text{olig}}^*)}{\partial n} \leq 0 \).

We now try to derive the number of vendors operating in this market in equilibrium. In order to incorporate the cost of operating in the security software market, we assume that the marginal cost to the vendor for an additional software subscription is zero. This is a reasonable assumption since, once the software is developed and updating facilities are established, the marginal cost of supporting an additional subscription in terms of production, distribution, and updating is negligible. Therefore, we only consider the fixed cost incurred by vendors to develop, market, and maintain the software. Since the vendors are all identical, this cost should be the same for all vendors. We use \( d \) to denote the normalized fixed cost. Therefore, if \( n^* \) denotes the number of vendors participating in the market in equilibrium, then \( R_{\text{olig}}(n^*) \geq d \) and \( R_{\text{olig}}(n^* + 1) \leq d \). Since \( R_{\text{olig}} \) is a decreasing function of \( n \), this implies that \( n^* = \lceil \tilde{n} \rceil \), where \( \tilde{n} \) is a decreasing function of \( n \).

Below, we state this more formally:

**Theorem 1.** The security software market is more competitive than the traditional market.

Theorem 1 indicates that the negative network effect in the security software market makes it more competitive. In other words, a higher \( g \) makes the industry more profitable and thereby capable of accommodating a larger number of vendors. In order to illustrate the result in Theorem 1 more clearly, we plot in Figure 3 how the number of vendors, \( n^* \), changes with \( g \). Two observations can be made from this plot. First, as expected, the number of vendors increases when the fixed cost \( d \) decreases. Second, \( n^* \) is a step-wise increasing function of \( g \), clearly indicating that the long-run equilibrium competition level increases with the negative network effect.

---

5 Recall that we are using a normalized price in the model. Therefore, it is necessary to normalize the fixed cost in the same manner as the price. More specifically, if the absolute fixed cost (per time unit) is \( D \), we use \( d = D/(\lambda_D L) \).

---

**Figure 3. No. of Vendors as a Function of Negative Network Effect**
We should mention that, even though we are considering the number of vendors in the market as an indicator for the level of competition, prior literature has proposed other metrics for this purpose, such as the Herfindahl-Hirschman index [14] and concentration ratio [1]. Even when measured against these indices, we find that negative network effect translates to a higher level of competition.

It is also worth mentioning that, although Theorem 1 and Figure 3 are all about the long-term equilibrium of the market, their implication must also be noted in terms of how the market is likely to behave during the transitory period—the initial years during which the industry matures. During this transitory phase, vendors typically make decisions with incomplete information about their own costs, the level of competition, and the consumers' reservation prices. A decision-maker must carefully consider the chance of success and profitability before entering a market [9, 23]. If it is believed that the market would be able to support a larger number of vendors, for a new entrant, the chance of succeeding or surviving increases. This, in turn, should attract more vendors to the market during the early transitory period—a result quite consistent with the empirical observation by Fosfur and Giarratana [8] that the security software market has attracted a very large number of vendors.

3.2. New Entry

We now turn our attention to what happens to the price when there is a new entry to the market. In 2006, when Microsoft entered the security software market, there was a large outcry about Microsoft practicing predatory pricing to drive out competition [7, 17]. Of course, it is well understood, both in theory and practice, that a new entry is supposed to drive prices down in a traditional market [9, 15, 20]. The question we would like to address here is whether the existence of negative network effect makes the new entrant more aggressive and encourages her to reduce the price more drastically when compared to a traditional market.

In order to analyze this in a rigorous manner, we calculate the relative price reduction when a new entrant enters the oligopoly security software market, currently with $n$ players, as:

$$
\Delta p(n) = \frac{p_{\text{olig}}(n) - p_{\text{olig}}(n + 1)}{p_{\text{olig}}(n)}.
$$

**Proposition 3.** The relative price reduction, $\Delta p(n)$, is an increasing function of $g$: $\frac{\partial (\Delta p(n))}{\partial g} \geq 0$, and is bounded: $\frac{1}{2n+1} \leq \Delta p(n) \leq \frac{3}{(2n+2)^2}$.

It is clear from Proposition 3 that the negative network effect induces a higher reduction in price with a new entry when compared to a traditional market. If this is not taken into consideration, the entrant's pricing policy may seem “predatory,” as perceived by Eckelberry [7]. Furthermore, the higher the negative network effect, the higher is the extent of this price reduction, although the effect is bounded. In order to see this more clearly, we plot $\Delta p(n)$ as a function of $g$ in Figure 4. This figure clearly illustrates that the impact of negative network effect on the relative price reduction can be significant. For example, for $n = 2$, $\Delta p(n) = 25\%$ for $g = 0$, but it increases to about 35% for $g = 5$.

4. Conclusion and Future Directions

A security software is a tool employed by individuals and organizations alike to prevent security exploitations of computerized systems. Over the last decade, the market for this type of software has seen a tremendous growth, both from the supply as well as the demand side. Unlike the typical software market, where the supply side is dominated by only a few providers, the market for security software is highly competitive with several major players. In this paper, we study why the security software market behaves differently from the other ones.

The positive network effect enjoyed by other software is much weaker for security software; this is because the compatibility issue of application data across users is not a big concern. On the contrary, there is a negative network effect for
security software—as the market coverage declines, the chance of an indirect attack from an unprotected computer increases. Though prior research has looked at this negative network effect in the context of application software security and patching [3, 4, 5], we are the first to study its possible impacts on the security software market. We observe that most of the idiosyncrasies in the prevailing market conditions of security software can be explained by this negative network effect.

More specifically, we find that the negative network effect leads to a free-rider problem, resulting in lower equilibrium market coverage, both in monopoly and oligopoly settings. On the other hand, it leads to higher prices and higher revenues for vendors. These results are consistent with observations from the security software market.

In the oligopoly market, negative network effect leads to a higher number of vendors in equilibrium. Yet, the price remains high enough that the industry overall is more profitable, when compared to a traditional market. This is also in line with realities of the security software market. Despite an insignificant marginal cost, an annual subscription of an antivirus software can cost as much as $40. At the same time, a three-year subscription can cost up to $90, comparable to the cost of Microsoft Word 2010, a product that enjoys substantial monopoly power.

We also find that negative network effect forces a new entrant to price more aggressively when compared to a traditional market. Perhaps, this explains why Microsoft—a late entrant in this market—priced its OneCare software at a level that, in 2006, was deemed predatory by many contemporary commentators [7, 17]. Thus, the incorporation of negative network effect in our model appears to capture the unique structure of the security software market and explain several counter-intuitive observations in practice, thereby providing useful insights for security software vendors on market competition and strategies.

There are several directions in which our results can be extended. For example, in this paper we have assumed that the level of negative network effect, \( g \), is the same for all consumers. It is possible for it to vary across individual users or organizations, depending on how they deploy complementary security technologies, such as firewalls and intrusion prevention systems. Further, nonlinear pricing through volume licensing is common in many software markets. One could investigate how nonlinear pricing might help security software vendors leverage the network effect. In addition, our model of a symmetric competitive equilibrium can be extended to asymmetric settings. We are examining some of these issues in our ongoing efforts.

References


