Software exclusivity and the scope of indirect network effects in the U.S. home video game market

Kenneth S. Corts, Mara Lederman *

Rotman School of Management, University of Toronto, 150 St. George Street, Toronto, ON, Canada M5S 3E6

1. Introduction

The home video game market has long been recognized as one in which network effects exist. As in many high-tech industries, these network effects are indirect. Home video games systems consist of a console (the hardware) and games that can be played on that console (the software). Because the value of a console is derived from the games that can be played on it, consumers prefer to buy a system with a greater variety of software. Because there are fixed costs to developing software, game publishers prefer to develop games for a console with a large base of users. Thus, consumers’ value of a particular console depends positively—though indirectly—on the number of other users of that console.

This paper explores the scope of indirect network effects in the home video game industry. While previous work has assumed that network effects exist only between users of a given console (for example, Clements and Ohashi, 2005), we argue that in recent years network effects have also come to exist between users of competing platforms in the same technological generation. Interestingly, this change in the scope of indirect network effects has occurred without any change in the degree of hardware compatibility. Home video games systems have always been, and continue to be, incompatible with one another in the sense that a console from one platform cannot run another platform’s software. As Katz and Shapiro (1985) explain, the scope of indirect network effects in an industry is typically determined by the degree of hardware compatibility because that determines the size of the potential market that a software firm can appeal to when trying to recoup its fixed costs of software development. That is, the degree of hardware compatibility determines the set of technologies whose users are potential purchasers of a given piece of software.

While changes in the degree of compatibility between video game systems have not been responsible for the change in the scope of network effects in this industry, we believe that changes in the degree of software exclusivity have. Over the past 20 years, the fraction of games titles that are released on more than one console has increased from about 13% to almost 40%. Just as compatible hardware allows software providers to spread the fixed costs of porting the game across platforms (for example, Clements and Ohashi, 2005), we argue that in recent years network effects have also come to exist between users of competing platforms in the same technological generation. Interestingly, this change in the scope of indirect network effects has occurred without any change in the degree of hardware compatibility. Home video games systems have always been, and continue to be, incompatible with one another in the sense that a console from one platform cannot run another platform’s software. As Katz and Shapiro (1985) explain, the scope of indirect network effects in an industry is typically determined by the degree of hardware compatibility because that determines the size of the potential market that a software firm can appeal to when trying to recoup its fixed costs of software development. That is, the degree of hardware compatibility determines the set of technologies whose users are potential purchasers of a given piece of software.

We thank Avi Goldfarb, Núria Mas, Tim Simcoe, Tom Hubbard, Neil Gandal, two anonymous referees, and seminar participants at IESE for helpful comments. We thank David Wright and Trevor Tombe for outstanding research assistance. The NET Institute provided financial support.

* Corresponding author.
E-mail addresses: kenneth.corts@rotman.utoronto.ca (K.S. Corts), mara.lederman@rotman.utoronto.ca (M. Lederman).

© 2008 Elsevier B.V. All rights reserved.

DOI: 10.1016/j.ijindorg.2008.08.002
released. This gives rise to indirect network effects between users of incompatible video game consoles.1

Following existing work in this area, we estimate the relationship between hardware demand and software availability and the relationship between software availability and the installed base of hardware. We begin by specifying a standard discrete choice model of hardware demand. One important benefit of this demand model is that it allows us to account for both exclusive and non-exclusive software in a straightforward way. In particular, exclusive and non-exclusive titles need not be distinguished in the utility function because consumers’ utility from having a particular game available on a console does not depend on whether that game is available on other consoles. Of course, whether a game is exclusive will affect the relative utilities of different consoles and, in turn, their market shares. The nested logit model that we use accounts directly for this as each console’s market share is a function of the characteristics (including software availability) of all products in the market. We can then use these market share expressions to illustrate the differential effects that exclusive and non-exclusive titles have on demand.

We then estimate a reduced-form software supply equation in which we allow the supply of games for a particular console to depend on both the installed base of that console and the installed base of competing consoles in the same technological generation. This allows for the possibility that the installed base of competing (and incompatible) hardware platforms can increase the supply of games for a console because the fixed costs of non-exclusive releases can be spread across users of multiple platforms. Furthermore, we allow the coefficient on the competitors’ installed base term to vary over time to capture the fact that software publishers’ incentives to produce non-exclusive software have increased. As we explain in greater detail in the next section, we believe that a rise in the importance of licensed content and other content costs that are not platform-specific, a decrease in “porting” costs, and a rise in the size and sophistication of independent game publishers have increased the attractiveness of non-exclusive releases.

Our empirical results support the existence of both a significant platform-level and an increasingly important generation-level network effect. As expected, we find that the supply of games for a console depends positively on the installed base on that console. However, we also find that the supply of games for a console depends on the installed base of other consoles, with this relationship being negative in the early part of our data and positive in the later part. Moreover, when we split our sample to look separately at exclusive and non-exclusive software, we find that—as hypothesized—our findings are driven by the presence of non-exclusive software. The relationship between the supply of exclusive games to a platform and its rivals’ installed base is never positive, while the pattern for non-exclusive games is the same as that of the pooled regression.

The video game market is often cited as the canonical example of a “tippy” market—one in which indirect network effects lead to dominance by a single firm. The complete dominance of the industry by Nintendo’s NES in the 1980s and early 1990s is often cited as evidence for this claim.2 However, with successive technological generations, this market has become significantly less dominated by any single console and, in each of the two most recent technological generations, three competing platforms (those of Nintendo, Sony, and Microsoft) have retained sizable market shares (see Fig. 1 for a depiction of market shares in four technological generations).3 While we do not estimate a dynamic model and therefore cannot use our results to illustrate how changes in software exclusivity affect the evolution of market shares over time, we believe that our findings provide at least suggestive evidence as to why this market has become less prone to tipping. Our results indicate that non-exclusive software affects a market much in the same way that hardware compatibility does, by changing the scope of indirect network effects. If network effects exist

\[\text{Long-Run Installed Base Market Share, by Generation}\]

![Long-run installed base market share, by generation. Source: Generation 3, Power Play (A); Generation 4, Power Play (B); Generations 5 and 6, NPD data.](image)

---

1 In fact, the theory literature (for example, Katz and Shapiro, 1985; Farrell and Saloner, 1992) has long recognized that this kind of indirect network effect can span incompatible platforms in the presence of technologies such as adapters and converters. The empirical literature has largely ignored this possibility, at least in part because the industries studied have typically been characterized by complete compatibility (e.g., CD players) or complete incompatibility (e.g., early home video games).


3 Fig. 1 shows the long-run installed base (IB) market shares of the major platforms in each technological generation. We define a platform’s “long-run” IB market share as its IB market share in the month in which the first major platform of the next generation is launched. We believe that the launch of the next generation acts as a good signal that the previous generation has reached a point of maturation.
Table 1

<table>
<thead>
<tr>
<th>Generations of the U.S. modern home video game industry</th>
</tr>
</thead>
<tbody>
<tr>
<td>U.S. launch date</td>
</tr>
<tr>
<td>------------------</td>
</tr>
<tr>
<td></td>
</tr>
<tr>
<td>Generation 3</td>
</tr>
<tr>
<td>NES</td>
</tr>
<tr>
<td>Generation 4</td>
</tr>
<tr>
<td>Super NES</td>
</tr>
<tr>
<td>Genesis</td>
</tr>
<tr>
<td>Generation 5</td>
</tr>
<tr>
<td>N64</td>
</tr>
<tr>
<td>Saturn</td>
</tr>
<tr>
<td>PlayStation</td>
</tr>
<tr>
<td>3DO</td>
</tr>
<tr>
<td>Jaguar</td>
</tr>
<tr>
<td>Generation 6</td>
</tr>
<tr>
<td>GameCube</td>
</tr>
<tr>
<td>Dreamcast</td>
</tr>
<tr>
<td>PlayStation2</td>
</tr>
<tr>
<td>Xbox</td>
</tr>
</tbody>
</table>

across users of different platforms, then it should be little surprise that
the tendency of this market to tip towards a single platform has fallen.

Though our primary emphasis is on how the software supply
relationship changes over generations, our demand analysis also
contributes new insights. We estimate demand using a number of
novel measures of game quality, providing evidence on just how much
more important hit titles are than typical titles in boosting demand for
a platform. We also use our demand estimates to illustrate how
important exclusive games are relative to non-exclusive games,
exploiting the fact that the nested logit demand model accounts for
titles both on one’s own platform and on others’ platforms.

This paper contributes to a growing literature that seeks to
estimate the role of indirect network effects in a variety of technology
industries. A number of early papers (including Gandal, 1995; Gandal
et al., 1999; Park, 2004) provide suggestive evidence of the role of
indirect network effects, but do not explicitly model the provision of
complements. Gandal et al. (2000) significantly advances the
literature by introducing, in the context of the CD market, the first
estimation of a simultaneous equations system that explicitly captures
the interrelationship of hardware demand and software supply.
Recent papers (including Rysman, 2004; Nair et al., 2004; Kaiser and
Wright, 2006, and the three papers on video games discussed below)
extend this approach by considering both sides of an indirect network
effects industry in the presence of competing platforms. Our emphasis
is different from this prior literature in two important ways. First, our
approach to identifying the importance of non-exclusive games
through estimation of cross-platform spillovers in a reduced-form
software supply equation is novel. Second, our main results focus on a
comparison across multiple generations that are characterized by very
game types in the consumer utility function. Perhaps not surprisingly, they
do not get sensible results from this specification.
Lee (2007) explores the impact of exclusive titles by examining the
importance of first-party software in the sixth generation console
market. He develops structural demand and supply models that allow
him to estimate porting costs under the assumptions that all first-
party titles are exogenously supplied and that all third-party software
providers freely choose which platforms to write for, given uniform
licensing terms (that is, absent preferred licensing terms for providers
of exclusive games). He then carries out a counterfactual exercise that
demonstrates that first-party titles benefited the smaller consoles in
the sixth generation, whose market shares would have been even smaller in the absence of their ability to obtain exclusive titles in this
manner.

The remainder of this paper is organized as follows. In the next
section, we provide relevant background information on the industry.
Section 3 describes the empirical approach. Section 4 describes the
data. Our results are presented in Section 5. A final section concludes.

2. Industry background

The home video game market is comprised of a small number of
competing, incompatible video game systems (or “platforms”). A
video game system consists of hardware (a console that is attached to a
television set) and software (game titles on either cartridge or CD).
Software produced for a given hardware platform cannot be played on
an alternate platform; however, distinct versions of the same software
title may be produced for multiple hardware platforms.

Platforms with similar technological characteristics are grouped into “generations” by industry observers. There have been seven
generations of platforms in the “modern” home video game industry,
spanning 1975 to the present. We focus our analysis on the years 1995
to 2005 inclusive. This time period covers the launch of most of the
platforms in generations five and all of the platforms in generation six.
It also includes several platforms from generations three and four
which were still actively selling during this period.4 Table 1 presents the
platforms that are included in our sample, grouped into
generations. The table also shows their date of introduction, basic
technological characteristics and the locations of their manufacturing
plants. Three technical factors determine the quality of a home video
game system: (1) instruction word length (in bits) of either the central

4 We ignore handheld game devices and PC games.
processor (CPU) or graphics processor (GPU); (2) clock speed (in MHz); and (3) the amount of RAM (in MHz). As the table indicates, platforms within a generation are typically quite similar on these three characteristics.

Each video game platform is controlled by what we call a “console manufacturer” and, as is evident from Table 1, many of the same console manufacturers appear in each successive generation (after the firm’s initial entry, of course). In addition to developing the hardware and operating system, a console manufacturer typically also produces some software that will run on this platform (so-called “in-house” or “first-party” titles). The console manufacturer will also enter into contracts with independent software publishers to provide games for the platform (known as “third-party” titles). Software publishers finance the development of the game (including obtaining and paying for any licensed content the game may use) and perform the marketing and distribution of the title. Game development (the actual programming) may be carried out by a development team internal to the publisher or may be contracted out to an independent game developer. Contracts between console manufacturers and software publishers generally stipulate that the console manufacturer is to provide software development tools to the publisher, while the publisher agrees to protect this intellectual property. The console manufacturer retains the right to approve games before they are developed and released for the console. The contract also specifies the per-unit royalties to be paid by the publisher to the console manufacturer. Finally, the contract may specify whether or not the game under development is exclusive to the console manufacturer.

As mentioned in the Introduction, over the past 20 years this industry has seen a significant increase in the prevalence of non-exclusive software. In Table 2, we document this change in software exclusivity. Note that the level of observation in Table 2 is the title (not the platform-title). The trends we observe would be even more pronounced if reported at the platform-title level since non-exclusive games would be double- or triple-counted depending on the number of platforms they were released for. Table 2 indicates that there has been a significant decrease in software exclusivity. 87% of titles released in generation three were never released on more than one platform, while 61% of titles were “always exclusive”. When we define exclusivity based on whether a game is initially released on just a single platform regardless of whether it is later released on an additional platform (what we call “begin exclusive” games), we see a similar pattern. In generation three, 98% of games were initially released on just one platform. By generation six, only 74% of games started on a single platform.

Interestingly, the changes in these averages have not been monotonic, as a greater fraction of generation five games than generation four games were exclusive. This is due at least in part to changes in the composition of console manufacturers from generation to generation. Specifically, Sony first entered the industry in generation five with its PlayStation and accounted for the majority of games in that generation. Perhaps because it was a new entrant, Sony also had a higher proportion of exclusive games than all other console manufacturers in that generation. However, examining the variation in exclusivity within a console parent demonstrates that the basic pattern of decreasing exclusivity does generally hold apart from these composition effects. For example, Sony’s generation six PlayStation 2 had a substantially lower fraction of exclusive titles (43%) than did its generation five PlayStation (70%).

Since exclusive titles are often games that the console manufacturer publishes itself (i.e., in-house games), one might wonder whether the observed decrease in the extent of exclusive software is simply reflecting a decrease in the prevalence of in-house games. In the third row of Table 2, we calculate the fraction of all titles that are published in-house. As this row indicates, over time, in-house games have come to account for a smaller fraction of total software titles (25% in generation three compared to 16% in generation six). However, as the next row reveals, this alone is not driving the observed reduction in exclusivity. The fraction of third-party titles that are exclusive has fallen substantially (from 88% in generation three to 54% in generation six). Together, these suggest that two trends in this industry. First, third-party titles are accounting for an increasingly large share of overall software available. Second, these titles are increasingly non-exclusive.

The patterns apparent in Table 2 clearly beg the question of why non-exclusive software has become more prevalent in this industry. Since our empirical approach essentially exploits variation across generations in the incentives for non-exclusive software, it is important for us to discuss the various changes that have taken place in the software side of the market. To motivate this discussion, we consider the incentives of a software publisher to release a game for one or more platforms.

---

5 Note that while we use the term “console manufacturer”, it is control of the operating system, rather than literal manufacturing of the hardware, that is relevant for our purposes. For example, Sony is the console manufacturer for the PlayStation2. This means that Sony owns the operating system for this platform and is responsible for the R&D that goes into the development and maintenance of PlayStation2 platform. Whether Sony outsources manufacturing of some or all of the components of the hardware is of no consequence for our purposes.

6 For example, suppose there are three titles with the first being exclusive to one platform, the second being exclusive to the other platform and the third being available on both platforms. Then the fraction of titles that are exclusive is two-thirds while the fraction of title-platforms that are exclusive is one-half.

7 When we use platform-titles as the level of observation, we calculate that while 81% of platform-titles in generation three were exclusive, only 37% of platform-titles in generation six were.
Suppose that there are two consoles in the market, one with a larger installed base than the other. Even absent contractual exclusivity, the decisions about whether to publish for one or both platforms and, in the case of only one, which one, are complex. With high enough porting costs, the decision will of course be to publish a particular title for at most one platform. However, Church and Gandal (1992) emphasize that it will still not be a dominant strategy to publish for the larger platform. If too many of the rival software publishers are writing for the larger platform, the smaller potential market of the smaller console will be more than offset by the reduced competition that the publisher faces there. In equilibrium, some subset of publishers will choose to write exclusive titles for each platform, balancing the fixed costs of game development against both the potential market size and intensity of competition associated with each platform. With lower porting costs (but still assuming no contractual exclusivity), publishers will choose to port titles to the second platform if the expected gross profits generated by sales of a title to the second platform’s customers exceed the fixed porting costs.

With heterogeneity in porting costs across types of titles, this process will generally lead to equilibrium with some combination of exclusive and non-exclusive titles on each platform. Now consider the effects of contractual exclusivity—that is, the ability of a platform manufacturer to write software contracts that stipulate exclusivity of a title in exchange for a lump sum transfer or a favorable licensing fee. In considering whether to contractually require exclusivity, the platform manufacturer must assess both the benefits gained from denying the title to another platform and the costs associated with the licensing fee concessions required to make exclusivity acceptable to the publisher. Hognestad and Yuen (2007) and Mantena et al. (2007) develop theoretical models to analyze precisely this tradeoff. Both papers emphasize that a platform with a larger installed base advantage will generally find it cheaper to induce exclusivity since the publisher has a weak outside option, and both papers argue that one should observe exclusive contracts with the dominant platform in some circumstances. However, Mantena et al. (2007) also argue that a larger platform may have less to gain from exclusivity than does a smaller competitor who views exclusive software as its only hope for establishing itself as a legitimate rival to the larger firm. As a result, both platforms may forgo exclusivity and accept the higher licensing fees associated with a non-exclusive game or the smaller firm may even negotiate an exclusive contract, depending on the relative installed bases of the platforms, the relative size of the market of new console purchasers, and other characteristics of the market.

The context for the software entry and exclusivity decisions described above of course depends critically on the technological environment, including for example the magnitude of the porting costs. These are to some extent under the platform manufacturers’ control at the product design stage. These firms can also affect the magnitude of porting costs somewhat as the market evolves, through decisions such as the extent of their investment in programming tools that facilitate porting or ease the use of cross-platform middleware for specific programming tasks. While it might seem that maximum incompatibility and high porting costs are most desirable for the platform manufacturers since they increase differentiation and increase the strength of platform-level network effects, Clements (2004) and Malueg and Schwartz (2006) both provide theoretical models of hardware–software markets in which platforms optimally choose some level of partial compatibility. In Clements’ model, this is done ex ante to reduce the strength of network effects with the objective of mitigating the intense rent-dissipating competition (e.g., penetration pricing) that strong network effects induce. In Malueg and Schwartz’s model, it is done in the process of competition to reap the spillover benefits of intense software competition among the publishers of titles on other (typically smaller) platforms.

Based on our reading of the trade press and other industry sources, we believe that changes over time in the cost structure and technology of game development, an increase in the use of licensed content in games, and changes in the market structure of the software industry have all influenced software publishers’ incentives to develop non-exclusive games. We discuss these three factors in turn below. Clearly, each of these factors is endogenous in the sense that the actions of software publishers and developers influence them. We do not take a stand on causality—for example, whether higher spending on licensed content led firms to pursue multi-platform release strategies or whether multi-platform releases enabled by lower porting costs led to an escalation in competition for licensed content. Rather, we emphasize that the context for software competition was different across generations, and we explore how these differences affected the strength and scope of indirect network effects in each generation.

Game development costs rose dramatically from each technological generation to the next. Industry analysts estimate that game development costs averaged well under $1 million in generation four, around $1 million in generation five, and $5–7 million in generation six (Loftus, 2006; Reimer, 2005). When the fixed costs of game development increase, more projects become viable only when they reach a very large audience—larger perhaps than any one platform can provide. In order to recoup these massive development costs, publishers have an incentive to release a title on multiple platforms.

The composition of development costs has also changed in important ways. With the rise to dominance of the CD-based console in generation five, games have become relatively more “content”—intensive. CDs make it cheaper to store vast quantity of graphical and musical data in a game, compared to the prior technology that used semiconductor chip-based cartridges. As a consequence, a larger fraction of the development costs has become attributable to tasks like music licensing or composition and performance, motion-capture studies, and background art and design (see Loftus, 2006; Reimer, 2005). Since these costs are not specific to a platform (for example, music is not operating system-specific), the fraction of initial development costs that must be duplicated in order to port a game to a second platform has shrunk. In addition, porting costs have fallen because of the rise of sophisticated cross-platform development tools called “middleware” that can dramatically reduce the costs of writing a game for multiple platforms compared to the traditional complete rewrite (Reimer, 2005).8 Middleware allows specific technical aspects of the game—for example, 3-D animation—to be developed within a programming tool that can provide output usable by the operating systems of more than one platform. In sum, the technology of game development has changed so that more of the initial costs incurred when writing a game for its first platform are avoidable in porting the game for a second. This reduction in the relative cost of porting clearly increases the attractiveness of non-exclusive releases.9

Furthermore, as shown in Table 2, software publishers are increasingly relying on licensed content and sequels. Industry observers attribute this to an attempt to mitigate risk of failure in an environment with skyrocketing development costs (Reimer, 2005). This move to a “blockbuster” model mimics an often cited development in Hollywood filmmaking, which experienced a similar simultaneous run-up in production budgets and increased reliance on sequels. Not only may ballooning budgets make the relative predictability of a licensed game seem attractive, but it also introduces another player into the game development process. If the owner of the relevant intellectual property (e.g., the Batman franchise) is pursuing a broad, multi-product or multi-channel strategy for disseminating its content and building/exploiting its brand, and if there are spillovers across markets (e.g., video game sales stimulate action-figure sales), it may provide publishers with incentives (perhaps with lower licensing fees) to develop a non-exclusive game based on its content (even if, in the

8 Also see the “Porting in gaming” entry in Wikipedia.
9 Mantena et al. (2007) state that industry figures indicate that porting costs are now typically in the range of 15% to 25% of the initial development costs.
narrow context of video games sales alone, it might be more profitable to license its content for development exclusively on one platform).

One final change that may have facilitated the rise in non-exclusive games is the growth and maturation of the software publishing industry, which could contribute to the decline in the proportion of (almost always exclusive) in-house games evident in the third row of Table 2. The final row of Table 2 shows that the average number of titles released by an independent publisher has grown more than three-fold over this period. As game development has increasingly become financed by publishers rather than by developers, and as development costs have soared, publishers have grown larger and better capitalized, with a number of the largest becoming publicly traded. This should correct some capital market imperfections likely present in the industry’s earlier days and make in-house publishing less necessary as a means of stimulating software development due to lack of financing. Of course, there may be other reasons that console manufacturers wish to be involved in in-house publishing, but the relative increase in the independent provision of games may be in part due to this development. And, since in-house games are almost always exclusive (apparently because console manufacturers are reluctant to share their IP and development tools with rival platforms, in addition to the fact that it may in their interest to stimulate platform-specific demand), a rise in independent publishing may increase the prevalence of non-exclusive games.

3. Empirical approach

Indirect network effects can be estimated in two ways. One can treat them as direct network effects and estimate a direct relationship between the demand for a given hardware platform and its installed base (see Ohashi, 2003 for an example). Or, one can explicitly account for the feedback between hardware and software by estimating both a hardware demand equation (in which hardware demand depends on software availability) and a software supply equation (in which software supply depends on the installed base of hardware). Finding a positive effect of software availability on hardware demand and a positive effect of hardware installed base on software supply establishes the (indirect) positive relationship between the demand for a hardware platform and the existing number of users of that hardware. This is the basic approach followed in the existing literature cited above, and we employ it as well. However, as we describe in the next two subsections, we modify it to account for the changing scope of indirect network effects in this industry.

The data we observe are the equilibrium outcomes of decisions by three types of agents: consumers, console manufacturers and software publishers. However, it is important to emphasize that we are only explicitly modeling two types of decisions: consumers’ decisions about what hardware to purchase and software publishers’ decisions about what platforms to supply games for. Console manufacturers’ decisions regarding pricing, advertising, product characteristics and entry and exit are not explicitly modeled. However, since these variables clearly influence the outcomes that we observe, we will attempt to highlight any assumptions we need about these decisions for the identification of our empirical models.

3.1. Hardware demand

We model a consumer’s choice of which (if any) hardware platform to buy in a given month as a discrete choice problem in which the consumer evaluates the utility that he would receive from each potential platform and chooses the one platform that maximizes his utility.10 We include an explicit outside good so that consumers also have the option of buying none of the platforms.11 Following Berry (1994), consumer i’s utility from purchasing console j, in month t is written as,

\[ u_{ij}^t = x_j' \beta + \alpha p_j^t + SW_j \lambda + \xi_i + v_{ij}^t \]

where \( x_j \) is a vector of observed characteristics for console j in month t, \( p_j \) is the price of console j in month t, \( SW_j \) is a vector of measures of software availability and/or quality on console j in month t, \( \xi_i \) is a vector of unobserved (to the econometrician) characteristics for console j in month t, and \( v_{ij}^t \) is an idiosyncratic error term. Each consumer is assumed to choose the product that maximizes his utility based on the observable and unobservable characteristics of the platform in that month.12

Continuing to follow Berry (1994), we let \( \delta t = x_j' \beta + \alpha p_j^t + \lambda SW_j + \xi_i \) denote the mean valuation of console j across all consumers, meaning we can interpret \( v_{ij}^t \) as the difference between consumer i’s valuation of console j in month t and the mean valuation. The distribution assumed for \( v_{ij}^t \) determines the choice probabilities and substitution patterns. We adopt a nested logit framework and group all inside goods (i.e., all consoles) into one nest and the outside good into another. This allows for correlation in \( v_{ij}^t \) across the inside goods, allowing them to be closer substitutes with each other than they are with the outside good. As Berry (1994) shows, with these assumptions on \( v_{ij}^t \) and by setting the mean utility of the outside good to zero, the following linear estimating equation for one-level nested logit can be derived,

\[ \ln(s_{ij}^t) - \ln(s_{ij}^t) = x_j' \beta - \alpha p_j^t + SW_j \lambda + \alpha \ln(s_{j-1}^t) + n^t \]

where, \( s_{ij} \) is console j’s within-group share (this is console j’s share of all consumers who purchase any console in month t) and \( \xi_i \) is the econometric error term. In our empirical specifications, we divide \( \xi_i \) into a time invariant component, which we will estimate as a platform fixed effect, and a time-varying component. The platform fixed effects will capture unobserved—as well as observable—elements of console quality that are time invariant. As a result, console characteristics that do not change over time but which we do have data on—e.g. hardware characteristics such as processor speed or whether the console is cartridge or CD-based—will not be separately identified.

Because of limited data on time-varying console characteristics, our empirical model primarily focuses on the effects of software availability and quality on utility. Indeed, we expect that, from a consumer’s perspective, these are likely the most important time-varying elements of platform quality. In our first set of demand specifications, we assume that consumers care only about the number of games available on a platform and expect that a greater number of games will increase the utility of a platform. We allow there to be

---

10 This modeling assumption would be violated if consumers purchased more than one console. While consumer “multi-homing” has increased over time, an NPD Group report estimates that, even in generation seven, only 5% of households owned more than one console (see http://www.gamedaily.com/articles/news/npd-72-of-us-plays-games-online-gaming-still-relatively-small/?biz=1).

11 The outside good also includes the option of using a PC for game playing. While a PC is technically a substitute for a home console, during the time period that we consider, the types of games played on PCs were typically very different from those played on home systems. We suspect this may change in the current generation of systems as seventh generation consoles focus more on online gaming. Even if substitutability between PCs and consoles is changing over our sample period, our platform fixed effects should account for this. Previous work on the home video game industry has also treated it as distinct from the PC market.

12 While we model this as a static purchase decision, in reality consumers may make decisions in one month based on their expectations of hardware quality and software availability in future months. To the extent that future developments are known to consumers from the platform’s initial launch, (for example, if it is always known that a particular blockbuster sports game will eventually be released for the platform, though it is not available at the date of platform launch), the value of these anticipated developments will be reflected in the platform fixed effects. If future developments become anticipated as the platform evolves (e.g., several months into the life of a platform, consumers learn that the platform is soon to introduce online gaming), the value of these anticipated developments will be part of the unobserved quality of the platform in the month they are learned. This provides an additional reason to instrument for these variables.
diminishing returns to additional games by including both a linear and quadratic measure of the number of titles available on a platform. However, because we know that there is, in fact, significant heterogeneity in the quality of games (and, indeed, only a small number of titles actually become “hits”), our second set of demand specifications includes alternative software measures that attempt to capture game quality. Specifically, we exploit the fact that we have actual software sales data for a portion of our sample period and use this data to construct a measure of the number of “hits” available on a platform where “hit” status is defined based on a game reaching certain sales thresholds. The precise construction of these variables is described in Section 4.

In addition to software measures, the one other time-varying console characteristic that we explicitly include in the model is platform-age. While platform-age may not directly affect a consumer’s utility from a console, we expect that it may proxy for certain perceived and actual changes in quality. For example, consumers may use console age as a signal for how soon they expect that platform to be supplanted by a new technological generation. Alternatively, other types of complementary products may emerge for a console (such as, gaming websites or magazines that offer “tips”) as it ages. We measure platform-age using either age-in-years or age-in-months fixed effects. We also include dummy variables to control for the month of the year. These capture the fact that the perceived quality of all consoles may be higher in some months such as November and December, when parents are purchasing gifts for children. Finally, in some specifications we replace the separate platform and age effects with platform-age (in years) fixed effects. This is our most flexible specification in the sense that these fixed effects control for the unobserved quality of each platform in each year of its “life”. However, it is also clearly the most “demanding” of our data in the sense that our coefficients are only identified off of variation during a 12 month period.

Before turning to software supply, it is worthwhile to highlight the way in which non-exclusive software affects the demand analysis. Exclusive and non-exclusive software titles do not need to be distinguished from one another in the data or in Eq. (2). This is because the right-hand side of Eq. (2) is literally the mean utility that a consumer receives from purchasing product \( j \). As is evident from the utility function in Eq. (1), whether or not a particular game is exclusive to console \( j \) has no impact on the utility that a consumer derives from console \( j \). That is, when evaluating his utility from a given console, a consumer will consider the games that can be played on that console, but not whether or not those same titles are also available on other consoles. However, whether or not software is exclusive to a console will affect the relative utilities of the different alternatives and therefore which console a consumer ultimately chooses. Games that are available on multiple platforms will increase the utility the consumer gets from each of those platforms and will, in turn, have little effect on the probability that the consumer chooses one of those platforms over another. On the other hand, an exclusive title will increase the likelihood that a consumer chooses a particular platform.

Given this, one way to illustrate the different effects that exclusive and non-exclusive software have is to calculate separate derivatives of demand with respect to exclusive and non-exclusive software. When calculating the change in demand in response to a change in exclusive software, the change in software will affect only the attributes and utility of platform \( j \). In contrast, when calculating the change in demand with respect to non-exclusive software, the change in software will affect both the attributes of platform \( j \) as well as the attributes of all other platform on which this game is available. In a logit model, the demand for any product depends on the character-istics of all products in the market; therefore, an increase in exclusive software will clearly have a larger effect on demand than an increase in non-exclusive software.

3.2. Software supply

Our supply equation generally follows the previous literature, with modifications that allow us to estimate whether and how the scope of indirect network effects in this industry has changed. In particular, we modify the software supply equation so that we can explicitly estimate whether, in successive generations, the installed base of competing platforms generates a positive spillover in the production of software for platform \( j \). This would provide evidence of a change in the scope of indirect network effects from being platform-specific to being generation-specific. To provide support for our argument that this is due to an increase in the prevalence of non-exclusive games, we then estimate this relationship separately for exclusive and non-exclusive games.

We estimate a reduced-form relationship between the variety of software available on a platform and that platform’s installed base of hardware. Specifically, we estimate the following equation,

\[
SW^j_t = \alpha_0 + \gamma_1 IB^j_t + \gamma_2 IB_{2t}^j + \eta_t
\]

where, \( \alpha_0 \) is a platform fixed effect, \( IB^j_t \) is the installed base of console \( j \) in month \( t \), \( IB_{2t}^j \) is the installed base of all other consoles in the same technological generation as platform \( j \), and \( \eta_t \) is an error term. Because we believe that there have been several important technological changes in the software side of this industry over the period we study, we also control for time effects with either (i) year and calendar month dummies, or (ii) year-month dummies.

\( \gamma_1 \) captures the relationship between the supply of software for platform \( j \) and its installed base, while \( \gamma_2 \) captures the relationship between the supply of software for platform \( j \) and its competitors’ (combined) installed base. We expect \( \gamma_1 > 0 \), meaning that increases in a platform’s installed base stimulate the provision of software for that platform. This is the source of the traditional platform-level indirect network effect. The sign of \( \gamma_2 \) depends on the nature of the technology of software provision. At one extreme, imagine that all development costs were completely specific to a platform, so that writing a version of the same game to run on a second platform required replication of all the same steps and the same costs. If the supply of inputs to this process was perfectly elastic, then the decision about whether to write a game for each potential platform would be a completely independent decision. A software firm would simply calculate the required potential market size and then develop the game for all platforms whose installed base (or projected installed base) exceeded that threshold. This would imply \( \gamma_2 = 0 \). Alternatively, if some inputs to the software development process were scarce, or less than perfectly elastically supplied, then such a model could imply \( \gamma_2 < 0 \) — that is, some crowding out of software development based on the growth of rival platforms. In this case, the growth of a rival platform is bad news for the focal platform because the consequent increase in software development for that rival platform diverts resources away from software development for the focal platform. Under another interpretation, software providers might use the installed base of providers employed by Clements and Ohashi (2005) and Prieger and Hu (2006) in estimating the supply of software. The primary difference is that we allow dependence of one platform’s software supply on all platforms’ installed base of hardware.

While we model this as a static relationship, in reality, software releases in one month will reflect publishers’ investments which were made in prior months, based on expectations of future installed base. However, some release decisions (such as, porting a game from one platform to another) will happen on a shorter time horizon. To the extent that a platform’s current installed base is anticipated by publishers and/or can be reacted to quickly, it should have a meaningful impact on current software availability.

\[14\] This is basically the same reduced form model employed by Clements and Ohashi (2005) and Prieger and Hu (2006) in estimating the supply of software. The primary difference is that we allow dependence of one platform’s software supply on all platforms’ installed base of hardware.

\[15\] While we model this as a static relationship, in reality, software releases in one month will reflect publishers’ investments which were made in prior months, based on expectations of future installed base. However, some release decisions (such as, porting a game from one platform to another) will happen on a shorter time horizon. To the extent that a platform’s current installed base is anticipated by publishers and/or can be reacted to quickly, it should have a meaningful impact on current software availability.

---

13 The inclusion of platform and age fixed effects prevents us from also including year fixed effects. This is the common age/year/cohort problem—see Hall et al. (2007) for example.
competitors as a signal about the likely evolution of platform \( j \); for example, if platform \( j \)'s competitors have large installed bases, software providers may infer that the market is likely to tip away from platform \( j \) and avoid writing software for the platform. This would again generate a negative spillover on software development from the growth of the installed base of rival platforms.

Now imagine instead that the portion of the fixed development costs that must be replicated to "port" the game to another platform falls. This increases the attractiveness of multi-platform releases and introduces the potential for a positive relationship between the supply of games for platform \( j \) and the installed base of platform \( k \)—that is, \( \gamma_2 > 0 \). Specifically, for a software provider who is contemplating writing a game for platform \( j \), the installed base of platform \( k \) represents an additional set of customers over which the fixed costs of this game can be spread. As the costs of porting games to additional platforms fall and/or as the development costs of games increase, games that may not be profitable if developed only for platform \( j \) might become profitable if developed for platforms \( j \) and \( k \). If so, the supply of games for platform \( j \) will be directly affected by the installed base of platform \( k \). This would give rise to generation-wide indirect network effects.

Given the evolution of the gaming industry described in Section 2, we specifically expect \( \gamma_2 \) to become more positive (or less negative) over time. We test this hypothesis by allowing the coefficient \( \gamma_2 \) to vary by generation so that we estimate how the relationship between competitors' installed base and the supply of games for platform \( j \) changes over our sample period. Note that because the Nintendo NES is the only generation three platform selling during our sample period, we cannot estimate this effect for this generation.

We can then combine the parameters of the demand and supply equations to establish the existence and scope of indirect network effects in this industry. In particular, a finding of a positive effect of software availability on utility in the demand equation (i.e., \( \lambda > 0 \) if the software measure is a scalar) and a positive effect of one's own installed base on software availability in the supply equation (i.e., \( \gamma_1 > 0 \)) establishes the presence of a platform-level indirect network effect. A finding of a positive effect of software availability on utility in the demand equation and a positive effect of other platforms' installed base on software availability in the supply equation (i.e., \( \gamma_2 > 0 \)) establishes the presence of a generation-level indirect network effect. Moreover, a finding that \( \gamma_2 \) increases over successive generations in our sample would indicate that the scope of indirect network effects has changed from users of the same platform to users of all platforms in the same generation.

3.3. Endogeneity

3.3.1. Hardware demand

With the inclusion of platform fixed effects, the error term in the demand equation can be interpreted as the difference between the unobserved quality of platform \( j \) in month \( t \) and its average unobserved quality (i.e., the error term captures components of unobserved quality that are not constant over a platform's lifecycle). Although changes to the physical or technical characteristics of a platform are rarely made in the middle of a console's lifecycle, there may nonetheless be several other sources of variation in unobserved quality. First, there may be changes in perceived quality that result from advertising campaigns or from the release of information about the platform—for example through positive or negative product reviews. Second, there may be changes in quality that result from the emergence of complementary products (other than software) that enhance the value of the platform—for example, a website or magazine that provides tips on how to solve games. Finally, although we attempt to capture the relationship between software and platform quality as well as possible, the error term may capture remaining measurement error in our software variables. Because these various types of changes in unobserved quality will be considered by firms when setting prices and by consumers when making choices, they will likely be correlated with both the price and within-group share variables.

In addition, if changes in unobserved quality are persistent over time (i.e., if there is serial correlation in the error term in the demand equation), then our software measures may be endogenous as well. This logic is perhaps best illustrated with an example. Suppose that in month \( t \), platform \( j \) has a "high" level of unobserved quality because the platform received positive reviews from gaming websites. This increase in quality will lead to higher demand for platform \( j \) that month. This, in turn, will increase the installed base of platform \( j \) in month \( t \) and, through the software supply relationship, increase the supply of games for platform \( j \) in month \( t+1 \). If the increase in the unobserved quality of platform \( j \) persists (i.e., the positive reviews in month \( t \) increase the perceived quality of platform \( j \) in month \( t+1 \) as well), then there will be correlation between the number of titles available on platform \( j \) in month \( t+1 \) and the error term in the demand equation in month \( t+1 \). Because we expect that at least some of the time-varying components of unobserved quality may persist for more than one month, we treat our software variables as endogenous.

The endogeneity problems that we confront here also plague much of the other empirical work in this area. Finding plausibly exogenous instruments has been difficult in most settings and has required some set of strong identifying assumptions. Our setting is no exception. Before explaining the instruments that we construct and the assumptions that we require, we first briefly summarize the different types of instruments that have been used in the literature. Gandal et al. (2000) requires the fewest assumptions for identification since they are able to treat price as exogenous (because CD players were competitively supplied) and because they have data on the fixed costs of pressing CDs which can be used to instrument for software availability. Nair et al. (2004)—who study the PDA market—use higher order and interaction terms of a firm's own observable product characteristics as well measures of its competitors' product characteristics as instruments for price, software and within-group share. Their first set of instruments is only justified by functional form assumptions (since a firm's own product characteristics are explicitly included in their demand model) while their second set of instruments requires the standard Berry et al. (1995) assumption that a firm's price is correlated with its competitors' characteristics but its (unobserved) characteristics are not. Clements and Ohashi (2005) exploit the fact that all of the consoles in their data launched in Japan before the U.S. and therefore use the exchange rate between the U.S. and Japan as well as the lagged console price in Japan as instruments for a console's U.S. price. The price of a console in Japan is a valid instrument for its price in the U.S. under the assumption that there are no common demand shocks—i.e., assuming there are no components to unobserved quality that are common across the U.S. and Japanese markets. Clements and Ohashi (2005) instrument for within-group share using the average age of software, which requires the assumption that firms' decisions to supply software to a platform are independent of that platform's unobserved quality. Finally, they use linear and higher order terms of platform-age as instruments for software availability. These instruments are available to them because they assume that these variables are excluded from the demand model. In contrast, we explicitly include platform-age in our demand model.

We construct two types of instruments for use in our hardware demand models. First, we instrument for the price of a console using the exchange rate between the U.S. dollar and the country in which that console was manufactured. While many of the early consoles in our data were manufactured in Japan, several of the later consoles

---

10 We collected information on the location of the manufacturing plants for each console. This data is presented in the final column of Table 1.
were manufactured in other places. Thus, we have time series variation in this instrument that is distinct across platforms (and obviously not captured by the platform fixed effects). We expect that as the U.S. dollar strengthens relative to the currency of the location of manufacture, the cost—and, in turn, price of the console in the U.S. market—should decrease.

Second, in the spirit of Berry et al. (1995), we construct measures of the extent of competition faced by a platform as well as the extent to which that competition comes from other platforms owned by the same “parent”. We use these measures to instrument for the within-group share and software variables. They can also serve as additional instruments for price. Specifically, our instruments include the sum of each hardware characteristic (processor speed, memory, and processor word length in bits) over the competing products in the market, the total number of competing platforms in the market, the number of competing platforms from the same generation, and the number of competing platforms from the same manufacturer. We expect that these variables will be correlated with platform j’s within-group share (because they affect the relative utilities of the different options), with platform j’s software (because, as we show in our modified supply equation, they affect software providers’ incentives to supply games to platform j), and with platform j’s price (because they affect platform j’s ability to raise prices).

Because our model includes platform fixed effects, variation in these instruments comes from changes in the set of platforms in the market at any given time due to entry and exit. Thus, for our instruments to be valid, we must assume that there is no correlation between a platform’s unobserved quality and the entry and exit decisions of competing platforms. Put differently, console manufacturers’ entry and exit decisions must not be made in response to changes in the unobserved quality and the entry and exit decisions of competing platforms. Put differently, console manufacturers’ entry and exit decisions must not be made in response to changes in the unobserved quality of existing platforms and changes in the unobserved quality of existing platforms (for example, through advertising) must not be made in response to the entry and exit of other platforms. The question becomes: how plausible are these assumptions? With respect to entry decisions, given the degree of technological uncertainty in this industry, the actual launch date of a console often ends up being random and, typically, later than the initial target date (see, for example, “Sony Delays PS3 Launch to November”). Thus, even if console manufacturers had intended to correlate their launch dates with unobserved aspects of existing platforms’ quality, the likelihood that this is actually achieved is quite low. With respect to exit decisions, we define a platform’s month of exit to be the first month in which its sales fall below 1000 units. While this cutoff may seem somewhat arbitrary, the arbitrariness of this cutoff makes it more likely that exit decisions are not specifically made in response to some change in the unobserved quality of existing platforms. For example, even if a platform experiences a permanent reduction in sales as a result of some change in the unobserved quality of a competing platform, there is no reason to expect that this change in unobserved quality took place in precisely that month in which we define the platform as exiting the market. Thus, it seems reasonable to assume that entry and exit decisions do not respond to the unobserved characteristics of platforms in the market.

The second part of the assumption requires that changes in unobserved elements of platform quality are not made in response to entry and exit. Note that changes in unobserved quality could result from the actions of the console manufacturer or software publishers. Since changes to physical or technical characteristics rarely take place in mid-lifecycle, the concern with respect to console manufacturers is whether they may change their advertising behavior or their supply of other complementary products in response to entry and exit. With respect to exit, to the extent that platforms typically exit the market at the end of their generation's lifecycle, it seems unlikely that other platforms from the same generation (who have not yet exited) will change their advertising or introduce new complementary products since they are themselves likely to exit soon and their console manufacturer is busy investing in its next-generation console. Furthermore, consoles in the next generation seem unlikely to make changes in investments in response to the exit of a platform from the previous generation since this platform is only an indirect competitor and their investment decisions are largely being made in response to the competition they face from within their generation.

With respect to entry, we must assume that console manufacturers do not change unobserved components of quality in response to new entry. This assumption is probably reasonable when considering the response of console manufacturers to entry by a next-generation platform. As explained above, the incumbent platforms are at the end of their lifecycle and are therefore not likely to make any major changes in investment decisions given that the new generation is already beginning. However, when an incumbent platform faces entry by a competing platform in the same generation, this assumption could fail. For example, when Sony (who launched its PlayStation2 in October 2000) faced the entry of Nintendo’s GameCube and Microsoft’s Xbox in November 2001, it might have responded with changes in elements of quality that we cannot observe. Similarly, when an incumbent platform faces entry, software publishers who had been providing games to the platform may reduce the resources they devote to that platform's games thereby changing unobserved elements of software availability. It is worth pointing out, however, that the endogeneity concerns discussed here are likely to be considerably less severe in a specification that includes platform-age fixed effects since these allow the unobserved quality of a platform to change in each year of its “life”. We include such a specification as a robustness check for our demand model and obtain results consistent with those of our other specifications.

### 3.3.2. Software supply

The installed base variables may be endogenous in the software supply equation. As above, this endogeneity problem will result if there is serial correlation in the error term. Specifically, a high value of $\eta_j^t$ will stimulate additional games for platform $j$ in month $t$. This will increase the utility and sales of platform $j$ that month which will increase the installed base of platform $j$ in month $t+1$. If $\eta_j^t$ is correlated with $\eta_j^{t+1}$, then Platform IB$^{t+1}$ will be endogenous. Because the software available on platform $j$ also affects the demand for competing platforms (and hence their installed base), the Rivals’ IB$^{j,t}$ terms can be endogenous as well.

We follow the literature in using functions of the age of a platform as an instrument for its installed base. Specifically, we use linear and quadratic terms of age in months to instrument for Platform IB. Similarly, we instrument for Rivals’ IB with functions of the age of the other platforms in the generation. We use the cumulative age in months of other consoles in the generation, as well as the maximum age in months of any console in the generation.

Because our software supply equation includes platform fixed effects, the error term in that equation captures unobserved changes in software suppliers’ incentives to supply games for a particular

---

17 One might be concerned that our various instruments might be highly correlated with each other. This is really only an issue with respect to the instruments that are based on the three hardware characteristics of rival platforms. Since our models are overidentified, we can estimate them using only one of these hardware characteristics and we find that the results are very similar. The reason why the three measures of competing products are not that highly correlated is because platforms from multiple generations compete against each other. This is what creates distinct variation in these three measures. For example, from PlayStation's perspective, when GameCube enters, this changes the number of competing platforms in the market but not the number of competing platforms from the same generation since PlayStation is a generation five console and GameCube is a generation six console. Similarly, from PlayStation2’s perspective, when PlayStation finally exits the market, this reduces the total number of competing platforms and the number of competing platforms from the same manufacturer but not the number of competing platforms from the same generation.


19 In calculating age in months for this purpose, we treat platforms that have exited the market as maintaining their age at the date of exit. This prevents the rival's age instruments from having discrete jumps due to platform exit.
console (their average incentives to supply games for a particular console are captured by the platform fixed effects). Thus, for our instruments to be valid, they must be correlated with a platform’s own (and competitors’) installed base but not otherwise correlated with software providers’ incentives to supply games for that platform. We have to assume that a platform’s age in a given month has no effect on software providers’ incentives to supply games for that platform that month other than through its effect on installed base. This would be violated if, for example, game development were characterized by significant learning curve effects within a platform. Given that a significant portion of game development is independent of operating system details (story development, art and music creation, animation and motion-capture performed through platform-independent tools) and given that platform manufacturers have strong incentives to provide good information, training, and developer tools prior to platform launch, the assumption that age has no direct effect on software provision seems plausible if perhaps not ideal. In fact, age-based instruments are the only instruments used elsewhere in the video games network effects literature. Both Clements and Ohashi (2005) and Prieger and Hu (2006) use platform-age and/or higher order terms to instrument for installed base.20

4. Data

4.1. Sources of data

Our empirical analysis combines several sources of data. Our data on hardware prices and quantities were obtained from the NPD Group, a market research firm. The NPD Group collects data from approximately two dozen of the largest game retailers in the United States.21 These retailers account for about 65% of the U.S. market. From this data, NPD formulates estimates of figures for the entire U.S. market. The NPD data provides monthly unit and dollar sales of each console with positive sales. Dollar sales are divided by unit sales to obtain an average monthly price for each console. Our hardware data covers the period 1995–2005, inclusive. We supplement these data with information on the technological characteristics and release dates of each console, which we collected from a variety of sources including analyst reports, company websites, and trade publications. Technological characteristics include processor speed, processor word length (8-bit, 16-bit, etc—the basis of most groupings of video game systems), console memory capacity, and whether the system was CD-ROM based. While these characteristics are not separately identified from the platform fixed effects that we include in our specifications, they are important because we use them to construct some of our instruments.

Our main source of data on software is www.mobygames.com, a website that seeks to “catalog all relevant information about electronic games (computer, console, and arcade) on a game-by-game basis.”22 MobyGames provides a database of software titles that includes the release date of each title, for each platform on which it appears and for each country in which it is released. The data also contain the name of the publisher, the genre of the game, and an indication of whether it uses content licensed from another party (like a movie studio). This database in principle goes back indefinitely in time, and is intended to capture all releases from the beginning of each platform’s life.23

4.2. Variables24

4.2.1. Market shares and installed base measures

Estimation of the hardware demand equation requires that we construct measures of each platform’s share of the total potential market for video game consoles as well as its share of the market captured by all of the “inside goods” combined. Following Clements and Ohashi (2005), we define the potential market for video games consoles in any month to be the number of households with a television (taken from the U.S. Census website) less the combined installed base of all active platforms (i.e., we want to subtract a measure of the number of consumers who are not in the market because they already own consoles). The simplest such measure would be the sum of each console’s past sales. We start by constructing this type of platform-level installed base measure (Platform IB). Since six of the ten platforms that we study launch within the period of our data, we can construct their past sales simply using the sales figures from our data.25 For the other four platforms, we obtained data on their year-end 1994 installed base from other sources. The year-end installed base figures for the platforms in generations three and four come from Shankar and Bayus (2003) while the figure for 3DO comes from an analyst report. For these four platforms, we construct Platform IB by combining these figures with the NPD sales data. Note that for each platform, we truncate the time series in the month that its sales first fall under 1000, defining this as a platform’s “exit” from the market.

20 Clements and Ohashi also use the average age of software titles with positive sales in that month, a variable that is unavailable to us because we have software sales data for only a relatively small subset of the period we study.

21 This set of retailers unfortunately does not include Wal-Mart but does include other large video game retailers like Amazon.com.

22 See http://www.mobygames.com/info/faq1#1. The information contained in MobyGames’ database is provided by the website’s creators as well as from voluntary contributions. All information submitted to MobyGames is checked by the website’s creators.

23 Its earliest entries are in 1972, for which there are six releases for the Odyssey game system.

24 Variable names and definitions appear in Table 3A. Table 3B presents summary statistics.

25 While Table 2 gives a launch date for Jaguar that predates our sample, its national launch was not until the end of 1994, so we ignore sales prior to our data which begins in January 1995.
While we could use this Platform IB measure in our construction of the total potential market, it presents a problem. In particular, this approach does not allow users of an old platform to gradually re-enter the market; rather, when an old platform “exits” the market by having its sales fall below 1000 in a month, it yields a discrete (and potentially huge) influx of new customers into the “potential market.” We solve this problem by modifying the installed base variable based on a depreciation rate. Specifically, we calculate each month’s installed base as a fraction of the previous month’s installed base plus the previous month’s hardware sales, where the depreciation rate varies with the age of the platform.\(^{26}\) We call this variable Depreciated IB. We focus on this particular formulation of the depreciation rate because, practically speaking, it yields declines in installed base that roughly coincides for most platforms with their exit from the market defined by current sales.

Having calculated each platform’s depreciated installed base, we then calculate the total potential market for video game systems in a month as the number of U.S. households with a television minus the sum of Depreciated IB over all active platforms. We construct a platform’s market share in a month as its hardware sales divided by the total potential market and call this Market Share. We construct a platform’s within-group share (Within Group Share) as its hardware sales that month divided by the total hardware sales of all active platforms that month. The share of the outside good (which is needed to construct the dependent variable for the demand equation) is calculated as one minus the combined market shares of all of the active consoles in a month.

Our software supply equation includes measures of both a platform’s own installed base and the installed base of the other platforms in its generation. We use Platform IB in the software supply equation and calculate Rivals’ IB as the sum of Platform IB over the competing platform’s in a generation.

### 4.2.2. Console characteristics

Because our hardware demand equation includes console fixed effects, non-time varying console characteristics (such as technical specifications) are not separately identified. The three time-varying characteristics that we include in the demand model are price, age and measures of software availability and quality. We construct the average price of each console in each month (Price) by dividing the console’s dollar sales by its unit sales. We measure the age of a platform as the number of months that have passed since the platform’s U.S. launch month. We call this variable Platform-Age. In some specifications, we measure a platform’s age in years since its U.S. launch.

We construct several software measures. We use the MobyGames data to calculate the total number of titles that have been released for a platform in the U.S. (starting from the platform’s release date and ending with the present month). We call this variable \# of Titles and use it as a measure of the cumulative amount of software available on a console in a month.\(^{27}\) We use this variable as our main software measure in our hardware demand equations and as the dependent variable in our software supply equations. In addition, we exploit the fact that, for a portion of our sample (1995 through February 2002), we have actual software sales data.\(^{28}\) We use this data for two purposes. First, we re-calculate \# of Titles using only releases that appear in the software sales data and defining their release date as the first month in which they appear on a platform with positive sales. We call this variable \# of Titles (NPD data). This variable both provides a check on the release dates supplied in the MobyGames data and also eliminates counting of any games that never experience positive sales.

Second, we use the software sales data to construct a title-level measure of game quality based on a game’s actual sales. Specifically, for each title (technically, title-platform) that launches between January 1995 and March 2001, we calculate its total dollar sales during the first 12 months after its release.\(^{29}\) We divide this by the platform’s installed base at the end of that 12 month period to obtain a measure of the average number of dollars spent on that game by platform owners. We then take the distribution of this variable over all generation five and six titles for which it is available and identify the 50th, 75th and 95th percentiles of this distribution. Using each of these cutoffs, we define three levels of “hit” games based on whether the game’s dollars per platform exceeded the 50th, 75th, or 95th percentile. We then calculate three corresponding measures of the number of “hits” available on a platform as the total number of titles that achieve a given level of “hit” status that have been released for a platform in the U.S. We call these three variables \# of Hits (50th Percentile), \# of Hits (75th Percentile), \# of Hits (95th Percentile). Note that because these measures as well as \# of Titles (NPD data) require us

---

\(^{26}\) The monthly depreciation rate (or rate of reentry into the potential market) is 0.00065 times the age of the platform in years. This yields about a 1% re-entry rate over the course of the first year and about a 7.5% re-entry rate over the course of a platform’s tenth year.

\(^{27}\) It is worth pointing out how exclusive and non-exclusive games are treated in this measure. \# of Titles counts the number of games that have previously been released for a platform. Therefore, a game that is exclusive to a platform will get counted in that platform’s monthly total once it is released. A game that is always available on multiple platforms will still get counted in each of those platforms’ monthly totals. A game that is first released on one platform and later released on another will get counted in each platform’s monthly total once it has been released for that platform. We believe that this approach is consistent with how consumers derive utility from software in the sense that they care about whether a game is available on a platform at a point in time and not whether or not it is also available on other platforms.

\(^{28}\) This data is also from the NPD Group. Software sales are by platform. Thus, if a given title is available on multiple hardware platforms, we know the sales on each platform.

\(^{29}\) Because we need to observe the first 12 months of sales, we cannot use titles that launch after March 2001 because their 12 months would not end by February 2002 (when our software sales data ends).
to observe the full history of software releases for a platform, they are only defined for the five platforms in our sample (PlayStation, Nintendo 64, Saturn, Dreamcast and PlayStation2) that launch after 1995 and have at least 12 months of sales before February 2002. This results in a greatly reduced sample size for specifications that use these measures.

5. Results

5.1. Hardware demand results

Table 4 and 5 present the results of the hardware demand estimation. All of our models include platform fixed effects (except for the final column of Table 4 which includes platform-age (in years) fixed effects) and calendar month fixed effects. All specifications treat the price, software and within-group share variables as endogenous. Recall that the right-hand side of our model is literally the mean utility of product \( j \) in month \( t \) and therefore the magnitudes of the coefficients on the right-hand side variables are not, on their own, informative. However, where relevant, we calculate either ratios of the coefficients or marginal effects (on market share) to illustrate and compare magnitudes.

In the first column of Table 4, the only software measure we include is the linear \# of Titles variable. The results from this specification indicate that higher prices lower consumers' utility from a platform while greater software availability increases their utility. The relative magnitudes of the coefficients in (4-1) suggest that decreasing the hardware price by $10 has the same effect on utility as introducing about 12 new games. In the second column of the table (and in all remaining columns), we allow for a quadratic relationship between utility and software. The estimates on \# of Titles and its square indicate a concave quadratic relationship. For example, the estimates from this specification imply that when the total number of games on a platform is held at its sample mean of 323, decreasing the hardware price by $10 has the same effect on utility as introducing 7.6 new exclusive games. However, for a platform with a stock of 100 titles, decreasing the hardware price by $10 has the same effect as introducing 4.4 new exclusive games. Thus, marginal games have a greater impact on utility when a platform has fewer games. As expected, the coefficient on log(Within Group Share) lies between zero and one for all specifications. This suggests that the “inside products” are indeed closer substitutes for each other than for the outside good.

As described in Section 3, if there are unobserved shocks to the quality of a platform that persist for more than one month, then the software variable may be endogenous. A second implication of unobserved changes in quality that persist over time is that the error terms in the hardware demand equation will be serially correlated. To

---

**Table 4**

<table>
<thead>
<tr>
<th>Fixed effects</th>
<th>log(s_j/s_n)</th>
<th>Platform, calendar month, age (in years)</th>
<th>Platform, calendar month, age (in months)</th>
<th>Platform-age (in years), calendar month</th>
</tr>
</thead>
<tbody>
<tr>
<td>Price</td>
<td>-0.001</td>
<td>(0.0001)**</td>
<td>-0.0038</td>
<td>-0.0046</td>
</tr>
<tr>
<td># of Titles (00s)</td>
<td>0.1215</td>
<td>(0.0066)**</td>
<td>1.0365</td>
<td>1.6076</td>
</tr>
<tr>
<td># of Titles (00s)^2</td>
<td>-0.0832</td>
<td>(0.0244)**</td>
<td>-0.1173</td>
<td>-0.1347</td>
</tr>
<tr>
<td>log(within_group_share)</td>
<td>0.5645</td>
<td>(0.0642)**</td>
<td>0.7081</td>
<td>0.7901</td>
</tr>
<tr>
<td>Observations</td>
<td>716</td>
<td>716</td>
<td>716</td>
<td>716</td>
</tr>
</tbody>
</table>

**Table 5**

<table>
<thead>
<tr>
<th>Fixed effects</th>
<th>log(s_j/s_n)</th>
<th>Platform, calendar month, age (in years)</th>
<th>Platform, calendar month, age (in months)</th>
<th>Platform-age (in years), calendar month</th>
</tr>
</thead>
<tbody>
<tr>
<td>Price</td>
<td>-0.0001</td>
<td>(0.0016)</td>
<td>0.0018</td>
<td>0.0013</td>
</tr>
<tr>
<td># of Titles (00s)</td>
<td>0.9634</td>
<td>(0.3027)**</td>
<td>0.0002</td>
<td>0.0003</td>
</tr>
<tr>
<td># of Titles (00s)^2</td>
<td>-0.0644</td>
<td>(0.0249)**</td>
<td>0.7090</td>
<td>0.7901</td>
</tr>
<tr>
<td># of Titles (NPD data)</td>
<td>0.3014</td>
<td>(0.2546)**</td>
<td>0.0404</td>
<td>0.1312</td>
</tr>
<tr>
<td># of Titles (NPD data)</td>
<td>0.3014</td>
<td>(0.2546)**</td>
<td>0.0404</td>
<td>0.1312</td>
</tr>
<tr>
<td># of Hits (50th Percentile)</td>
<td>1.3738</td>
<td>(0.5228)**</td>
<td>1.3738</td>
<td>1.3738</td>
</tr>
<tr>
<td># of Hits (75th Percentile)</td>
<td>2.8476</td>
<td>(1.4643)**</td>
<td>2.8476</td>
<td>2.8476</td>
</tr>
<tr>
<td># of Hits (95th Percentile)</td>
<td>6.1597</td>
<td>(3.0257)**</td>
<td>6.1597</td>
<td>6.1597</td>
</tr>
<tr>
<td>log(within_group_share)</td>
<td>0.3014</td>
<td>(0.2546)**</td>
<td>0.0404</td>
<td>0.1312</td>
</tr>
<tr>
<td>Observations</td>
<td>716</td>
<td>716</td>
<td>716</td>
<td>716</td>
</tr>
</tbody>
</table>

**Notes:** *significant at 10%; **significant at 5%; ***significant at 1%. All specifications report 2SLS estimates. Coefficients on the fixed effects are not reported. All standard errors are robust to arbitrary heteroskedasticity and arbitrary autocorrelation. All specifications include data from May 1995 to March 2001 for the following platforms: Nintendo 64, PlayStation, Saturn, Dreamcast, and PlayStation2.
account for this, we do two things. First, in the third column of Table 4, we re-estimate (4-2) using standard errors that are robust to arbitrary autocorrelation. The results are unchanged when we do this. Second, we experiment with two stronger sets of fixed effects. In (4-4), we replace the age (in years) fixed effects with age (in months) fixed effects. If some of the persistent changes in unobserved quality are common across platforms and occur at the same point in the lifecycle, these fixed effects will more carefully control for them. The results from this specification are very similar to those in (4-2). For example, the coefficients from (4-4) imply that for a platform with a stock of 100 titles, decreasing the hardware price by $10 has the same effect on utility as introducing 3.9 new exclusive games (as compared to 4.4 games using the numbers from (4-2)).

In the final column of the table, we replace the platform and age (in years) fixed effects with platform-age (in years) effects. Intuitively, these estimate the unobserved quality of each platform in each year of its life (i.e., rather than estimate one level of unobserved quality for the Sony PlayStation, we now estimate ten different levels). These fixed effects will clearly do a better job of capturing unobserved and potentially persistent changes in quality that differ across platforms and occur at different points in a platform’s life. Moreover, the error term in the hardware demand equation is much less likely to exhibit serial correlation with their platform’s life. Moreover, the error term in the hardware demand equation is much less likely to exhibit serial correlation with their platform’s life. Moreover, the error term in the hardware demand equation is much less likely to exhibit serial correlation with their platform’s life. Indeed, these estimates are not reported. All standard errors are robust to concurrent serial correlation.

The average share of the total market for all platform-month observations is 0.00217; given the roughly 100 million households in the total potential market, this amounts to about 217,000 units. The calculation described above implies that, on average, the introduction of one exclusive game increases a platform’s share of the potential market by 0.0000269, or about 2690 units per month. We can compare this with the estimated effect of a game that is compatible with all available hardware platforms. Such an addition increases each firm’s predicted share of the potential market by only 0.0000129, or about 1290 units. That is, the effect on hardware demand of an exclusive release is on average a little over two times as large as the effect of a title available on all platforms.

In Table 5, we estimate specification (4-3) using alternate software measures. Recall that these measures are only defined for five of the platforms in our data and, for two of these (those in generation six), only for a small number of months. As a result, for all of the specifications in Table 5, our sample size is substantially reduced (from 716 to 194). In the first column of the table, we replicate (4-3) using the smaller sample. In the second column, we replace # of Titles with # of Titles (NPD data). In both specifications, we again find a concave quadratic relationship between software availability and utility. However, probably due to the reduction in sample size, the coefficient on the price variable in all specifications in this table is estimated with a larger standard error. As a result, we illustrate the magnitudes of the software estimates by calculating the marginal effect on sales of one additional game. The estimates in (5-1) imply that, on average, one additional title increases hardware sales by about 1020 units while those in (5-2) imply that, on average, one additional title increases hardware sales by about 3200 units. The consistency between the MobyGames and NPD versions of the software variable (albeit only in this reduced sample) provides a check on the robustness of our results in Table 4.

In the third through fifth columns of the table, we attempt to capture game quality by using our various “hits” measures. In (5-3), we measure software availability and quality using the # of Hits (50th Percentile). In (5-4), we replace this # of Hits (75th Percentile). Finally, in (5-5), we include with the # of Titles (NPD data) variable—to control for total software available—as well as the # of Hits (95th Percentile)—to capture so-called “blockbuster” games. We find positive and statistically significant coefficients on all of these software measures. Moreover, the marginal effects that we calculate imply that higher quality games have a greater impact on utility. Specifically, we find that one additional title that is a 50th percentile “hit” increases hardware sales by 6020 units while on additional title that is a 75th percentile “hit” increases sales by over 16,000 units. An additional

<table>
<thead>
<tr>
<th>Dependent supply estimates</th>
</tr>
</thead>
<tbody>
<tr>
<td># of Titles</td>
</tr>
<tr>
<td>Platform, year-month</td>
</tr>
<tr>
<td>(6-1)</td>
</tr>
<tr>
<td>(6-2)</td>
</tr>
<tr>
<td>(6-3)</td>
</tr>
<tr>
<td>Platform IB (000s)</td>
</tr>
<tr>
<td>0.0351</td>
</tr>
<tr>
<td>(0.0015)**</td>
</tr>
<tr>
<td>Rivals’ IB (000s)</td>
</tr>
<tr>
<td>0.0382</td>
</tr>
<tr>
<td>(0.0008)**</td>
</tr>
<tr>
<td>Rivals’ IB*Generation = 4</td>
</tr>
<tr>
<td>-0.0076</td>
</tr>
<tr>
<td>(0.0025)**</td>
</tr>
<tr>
<td>Rivals’ IB*Generation = 5</td>
</tr>
<tr>
<td>0.0020</td>
</tr>
<tr>
<td>(0.0008)*</td>
</tr>
<tr>
<td>Rivals’ IB*Generation = 6</td>
</tr>
<tr>
<td>0.0041</td>
</tr>
<tr>
<td>(0.0004)**</td>
</tr>
<tr>
<td>Observations</td>
</tr>
<tr>
<td>716</td>
</tr>
</tbody>
</table>

+ significant at 10%; * significant at 5%; ** significant at 1%. All specifications report 2SLS estimates. Coefficients on the fixed effects are not reported. All standard errors are robust to arbitrary heteroskedasticity. Standard errors in (6-5) are also robust to arbitrary autocorrelation.
95th percentile “hit” increases sales by almost 44,000. We interpret these results as indicating that hardware demand is driven by both the quantity and quality of software available and that, consistent with anecdotal evidence, “blockbuster” games are critical drivers of market share in this industry.

5.2. Software supply results

Table 6 presents our main software supply equation. In all specifications, the dependent variable is # of Titles. All specifications include platform fixed effects. The first three columns include month-of-year fixed effects to control for seasonality and year fixed effects to control for common market-wide time effects. The installed base variables are treated as endogenous in all specifications.

In the first column of the table, we estimate the relationship between the supply of software for a video game platform and that platform’s installed base of hardware. As expected, we find a positive relationship. The combination of this positive relationship and the positive effect of software on hardware demand in Tables 4 and 5 establishes the presence of platform-specific indirect network effects.

The most interesting and novel part of our software supply results is the relationship between the supply of games for a platform and the installed base of its competitors. In the remaining columns of Table 6 we add variations of Rivals’ IB, the variable that measures the combined installed base of competing platforms in a generation. In column two, we do not allow the coefficient on this variable to change by generation, and we find that, averaging across generations, Rivals’ IB has a positive and significant effect on # of Titles. That is, increasing the installed base of a platform’s competitors increases the supply of games for that platform. As discussed in Section 3, this positive coefficient is consistent with cross-platform spillovers due to non-exclusive games, whereas a negative coefficient would be consistent with a competitive effect by which other platforms with increased sales lure away software publishers from a platform.

In columns three through six, we interact Rivals’ IB with dummies for generations four through six.\(^{30}\) Since we have data on only one platform in generation three (the NES), we cannot estimate a coefficient on Rivals’ IB for that generation. In (6-3), the coefficients on all three Rivals’ IB terms are significantly different from zero. Moreover, they are monotonically increasing over successive generations. The coefficients imply that, in generation four, increases in the installed base of its competitors lower the supply of games to a platform. In generation five, the coefficient is positive, meaning that increases in its competitors’ installed base stimulate the supply of software for a platform. Finally, in generation six, the coefficient is again positive and twice as large as for generation five. The magnitude of this effect is reasonably large with the point estimates in (6-3) implying that an increase in rivals’ combined installed base is worth, strictly in terms of software availability for a platform, about 15% of what an increase in one’s own installed base is worth.

In columns four and five, we replace the calendar month and year fixed effects with month-year fixed effects to provide more flexible controls for time effects. Column five also allows for arbitrary autocorrelation in the error terms. The results are quite robust to both of these specification changes. While the generation five rivals’ installed base effect becomes insignificant, the point estimates remain monotonically increasing from negative to positive over the three generations. Moreover, the generation four and generation six rivals’ installed base effects are significant and remain negative and positive respectively.

The positive coefficient on Rivals’ IB*Generation = 6 establishes the existence of indirect network effects that are operating at the generation level. Combining this with the demand results suggests that a consumer’s utility of purchasing a Microsoft Xbox is not only increasing in the number of other Xbox users but also in the number of users of the PlayStation2 because, with high costs of developing common content and low enough porting costs, their combined size will induce software providers to write games that are released on both platforms. Thus, the scope of indirect network effects in this industry has changed.

If, as we argue, this positive cross-platform spillover in later generations truly reflects the rise of non-exclusive games, then the effect should be most pronounced for precisely those games and when the porting costs and other conditions of the software publishing market make porting games across platforms attractive (i.e., generation six). The number of exclusive games on a particular platform should not rise in response to rivals’ installed base increases. In fact, if there is any cross-platform effect for exclusive games, it should be negative, reflecting a competitive effect by which software publishers are enticed away from a platform to write for its better-selling rivals.

We test these hypotheses in Table 7 by looking separately at the effect on exclusive and non-exclusive third-party games. We exclude in-house games because the incentives behind their publication are subtle. The first column of the table replicates specification (6-5) for easy comparison with the following columns. In (7-2), we define the dependent variable as the number of third-party non-exclusive titles. The results are consistent with those using the pooled software measure. We find that the rivals’ installed base coefficients are monotonically increasing and the coefficient for generation six is positive and significant. In (7-3), we define the dependent variable as the number of exclusive third-party titles available on a platform. We now find that the coefficients on the Rivals’ IB terms are negative in all three generations and significant in generations four and six. Thus, exclusive third-party games do not exhibit positive cross-platform

\(^{30}\) To instrument for these interactions with Rivals’ IB, we interact our instruments (described in Section 3.3) with generation dummies.
spillovers at any point in our sample period. This suggests that the positive coefficients we estimate on Rivals’ IB in our main specifications are driven by increased incentives for multi-platform releases and not by some other source of positive correlation between one platform’s installed base and competing platforms’ software availability.

5.3. Implications of the estimates

We can illustrate the implications of the change in the scope of indirect network effects in this industry by tracing out the effect on hardware sales of an exogenous change in software availability. In markets with indirect network effects, an exogenous increase in software for a platform has both a direct effect on hardware demand (increasing utility and sales) and an indirect effect (since this direct effect stimulates the provision of additional software, which further increases utility and sales). Tracing through these effects illustrates how the “strong get stronger” and the “weak get weaker” due to what it sometimes called a “virtuous cycle” (see, for example, Shapiro and Varian, 1999). However, once network effects exist across users of competing platforms, this cycle is weakened. In particular, an increase in (exclusive or non-exclusive) software for a platform still increases the installed base of that platform, which still increases the supply of games for that platform, but now it also increases the supply of games for competing platforms. Thus, in this scenario, one could say that the strong get stronger but the weak do not get as weak as they would have absent this cross-platform positive spillover.

Although we do not estimate a dynamic model, we can use the results of our demand and supply equations to illustrate one iteration of this cycle under the assumption that network effects do not exist across platforms (i.e., using our generation four estimate of the coefficient on Rivals’ IB) and under the assumption that network effects do exist across platforms (i.e., using our generation six coefficient on Rivals’ IB). We do this by calculating the effects of a hypothetical addition of 10 exclusive titles for PlayStation 2. We do this separately (not cumulatively) for each month from January 2004 onward, which is the period in which PS2, GameCube, and Xbox are the three competing platforms in generation six, and then present the average effects over these months.

We calculate these effects as follows. We begin with the same calculation employed in the illustration of the demand estimates, in which we back out implied mean utilities and calculate predicted shares of the total market. We then increase PlayStation2’s mean utility to reflect the addition of ten new games. We make no changes to software measures of the other platforms, consistent with the interpretation that these software titles are exclusive to PlayStation2. We then re-calculate the firms’ predicted market shares based on these new mean utilities, calculate the difference from the original implied share, and multiply this by the size of the potential market. We refer to this increase in market share as the direct effect of the increase in software for the PlayStation2, and we report it in the first row of Table 8. This row demonstrates two consequences of an increase in software for PS2. First, the increase in software diverts sales from both of the other platforms to PS2. Second, total sales grow since the sum of the numbers in this row is positive; that is, PS2 gains more sales than Xbox and GameCube lose.

We then use the estimates from our software equation to predict how the number of titles provided on each platform changes as a result of the installed base changes that result from the sales changes reported in the first row. This takes into account both the change in each platform’s own installed base and the change in other platforms’ installed base. Thus, the implied effect on software provision will vary depending on whether we use the generation four or generation six estimates. Finally, we use this implied change in software availability to calculate the indirect effect on hardware sales of the initial hypothetical increase in exclusive titles for PS2. The second and third rows of Table 8 follow this exercise through using the generation four estimates—that is, the estimates that show a negative cross-platform spillover. The fourth and fifth rows of Table 8 follow the exercise through with the generation six estimates, which show a positive cross-platform spillover.

The contrast in these calculations using generation four and generation six estimates helps make concrete the implications of the change in the scope of indirect network effects that we find. Note that in both scenarios (rows two and four) the effect of the diversion of hardware sales to PS2 is to stimulate additional software provision for PS2 and to reduce software availability for the other two platforms. This is one sense in which the indirect network effects lead the strong to get stronger and the weak to get weaker—an exogenous increase in software for one platform leads to both additional software for that platform and a decrease in software for other platforms. However, the magnitudes of the effects are quite different between the two scenarios. The broadening of the network effect in moving from generation four estimates to generation six estimates cuts the implied follow-on increase in software for PS2 slightly, from 1.6 to 1.4 titles. However, the negative effect on the other platforms is cut dramatically in both cases—by roughly an order of magnitude (−0.6 to −0.06 and −0.7 to −0.09). In this sense, the broadening of the network effects leads to a scenario in which the strong still get stronger, but the weak do not suffer nearly the same penalty from further increases in the leader’s sales. This is, of course, precisely because the leader’s increase in sales creates a positive spillover in software provision for the smaller platforms.

The third and fifth rows of Table 8 show how this effect on follow-on software provision translates into an indirect effect on hardware sales. The numbers there exhibit a similar pattern: the broadening of network effects in generation six somewhat weakens the indirect benefit to the leader (by about 30%) but dramatically reduces the penalty to the weaker firms (by more than 50%).

6. Conclusion

In this paper we investigate the scope of indirect network effects in the home video game industry and argue that, despite the fact that all home video game systems are incompatible, indirect network effects increasingly exist between users of rival systems. This is because multi-platform releases (or non-exclusive software titles) allow software publishers to spread the fixed costs of game development over users of competing platforms. Over time, software publishers’ incentives for multi-platform releases have increased as the fixed development costs of common content have soared, while the fixed costs of porting a game to additional platforms have fallen. Both of these increase the relative profitability of non-exclusive software releases, which in turn lead to the presence of cross-platform indirect network effects.

Our empirical results support this interpretation. We estimate a model of hardware demand and software supply that indicates that while platform-specific indirect network effects exist, in recent years generation-wide indirect network effects have also come to exist. Thus, the scope of indirect network effects has changed. We also carry out a simple illustrative exercise that shows why the

<table>
<thead>
<tr>
<th>Table 8</th>
<th>Direct and indirect effects of 10 additional exclusive titles for PlayStation2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>PS2</td>
</tr>
<tr>
<td>Direct effect on hardware sales</td>
<td>+56,000</td>
</tr>
<tr>
<td>Generation 4 estimates</td>
<td></td>
</tr>
<tr>
<td>Resulting effect on software titles</td>
<td>+1.6</td>
</tr>
<tr>
<td>Indirect effect on hardware sales</td>
<td>+11,200</td>
</tr>
<tr>
<td>Generation 6 estimates</td>
<td></td>
</tr>
<tr>
<td>Resulting effect on software titles</td>
<td>+1.4</td>
</tr>
<tr>
<td>Indirect effect on hardware sales</td>
<td>+7900</td>
</tr>
</tbody>
</table>
The presence of positive cross-platform spillovers weakens the benefit that a platform gets from a strategy aimed at stimulating demand, such as the introduction of new software. While we do not estimate a dynamic model, we believe that our findings are suggestive of why, in recent generations, this industry has not been dominated by any single console. Furthermore, our results indicate that hardware compatibility is not the only factor affecting the scope of network effects in an industry. Rather, non-exclusive software creates another avenue by which cross-platform spillovers can arise, as changes in software development and porting technology increase the attractiveness of expanding the potential market through multi-platform releases.

Appendix A. OLS regression of specification (4-3)

<table>
<thead>
<tr>
<th>Dependent variable</th>
<th>log(s_ij)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fixed effects</td>
<td>Platform, calendar month, age (in years)</td>
</tr>
<tr>
<td>Price</td>
<td>-0.0017 (0.0005)**</td>
</tr>
<tr>
<td># of Titles (00s)</td>
<td>0.2344 (0.0813)**</td>
</tr>
<tr>
<td>(# of Titles (00s))^2</td>
<td>-0.0205 (0.0071)**</td>
</tr>
<tr>
<td>log(within_group_sharei)</td>
<td>0.9238 (0.0301)**</td>
</tr>
<tr>
<td>Observations</td>
<td>716</td>
</tr>
</tbody>
</table>

** significant at 1%. Coefficients on the fixed effects are not reported. Standard errors are robust to arbitrary heteroskedasticity and to arbitrary autocorrelation.

Appendix B. OLS regression of specification (6-5)

<table>
<thead>
<tr>
<th>Dependent variable</th>
<th># of Titles</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fixed effects</td>
<td>Platform, year–month</td>
</tr>
<tr>
<td>Platform IB (000s)</td>
<td>0.0248 (0.0007)**</td>
</tr>
<tr>
<td>Rivals’ IB*Generation=4</td>
<td>-0.0163 (0.0043)**</td>
</tr>
<tr>
<td>Rivals’ IB*Generation=5</td>
<td>-0.0011 (0.0011)</td>
</tr>
<tr>
<td>Rivals’ IB*Generation=6</td>
<td>0.0044 (0.0006)**</td>
</tr>
<tr>
<td>Observations</td>
<td>716</td>
</tr>
</tbody>
</table>

** significant at 1%. Coefficients on the fixed effects are not reported. Standard errors are robust to arbitrary heteroskedasticity and to arbitrary autocorrelation.

References