THE CONTROL OF PORTING IN TWO-SIDED MARKETS

RUFUS POLLOCK

CAMBRIDGE UNIVERSITY

2007-01-24

Abstract. A sizable literature has grown up in recent years focusing on two-sided markets in which economies of scale combined with complementarities between a platform and its associated 'software' or 'services' can generate indirect network effects (that is positive feedback between the number of consumers using that platform and the utility of an individual consumer). In this paper we introduce a model of 'porting' in such markets where porting denotes the conversion of 'software' or 'services' developed for one platform to run on another. Focusing on the case where a dominant platform exists we investigate the impact on equilibrium and the consequences for welfare of the ability to control porting. Specifically, we show that the welfare costs associated with the 'control of porting' may be more significant than those arising from pricing alone. This model and its associated results are of particular relevance because of the light they shed on debates about the motivations and effects of actions by a dominant platform owner. Recent examples of such debates include those about Microsoft's behaviour both in relation to its operating system and its media player, Apple's behaviour in relation to its DRM and iTunes platform, and Ebay's use of the cyber-trespass doctrine to prevent access to its site.

Keywords: Network Effects, Two-Sided Markets, Porting, Antitrust, Competition

JEL Classification: L15 L12 L13

1. Introduction

Several recent cases, which we discuss in more detail below, have focused economists' attention on the motivations and effects of the behaviour of a dominant firm in two-sided markets. We believe that much of this activity can usefully be interpreted in terms of efforts to control (and prevent) 'porting' – where porting denotes the conversion of a 'software' or 'service' associated with one platform to run on another platform. Building

Faculty of Economics, Cambridge University, Sidgwick Avenue, Cambridge, CB3 9DD. Email: rp240@cam.ac.uk or rufus@rufuspollock.org. First version July 2005, this version January 2007.

on the existing literature on two-sided markets we develop a formal model of 'porting' and, focusing on the case where a dominant platform exists, we use this model to investigate the impact on equilibrium and the consequences for welfare of the ability to control porting. Specifically, we show that the welfare costs associated with the 'control of porting' may be more significant than those arising from pricing alone. These results are of particular importance for two reasons. First, in their general implications for the evaluation of dominant firm behaviour. Second, for the insights gained into strategic behaviour in two-sided markets and their consequences for welfare. Such insights are particularly germane given the growing importance of markets exhibiting network effects.

For example, much of the 1998 case of *US vs. Microsoft* as well as more recent antitrust disputes in Europe over Microsoft's media player can be seen as related to efforts to control porting. In the 1998 case there was the alleged tying of Internet Explorer browser as well as efforts to undermine compatibility with other systems, for example, by subtly changing the Windows version of the Java Virtual Machine (Jackson, 1999). Similarly, the media player dispute concerned the bundling of Microsoft's own Media Player 'for free' with the operating system. In both cases there has been considerable debate[1] over the motivations for, and consequences of, Microsoft's behaviour, especially as to whether these sorts of activities could be described as 'tying'.[2] To our mind much of this behaviour is best seen light of efforts to control porting and thereby preserve the market power associated with the 'Applications Barrier to Entry' (as the indirect network effects were termed in that anti-trust action). Unlike with traditional tying, Microsoft's actions, though obviously directly *affecting* competing applications (Netscape's Browser, Real Networks Audioplayer etc), were not *directed* at them. Rather, they were motivated by fear that losing control of key Application Programming Interfaces (APIs) and user services would made it easier for end-user applications and services to port between operating system platforms, which would, in turn, make it easier for consumers to switch between different platforms and thereby reduce Microsoft's market power.[3]

[1]See, for example, Hall and Hall (2000); Davis and Murphy (2000); Fisher (2000); Bresnahan (1989); Liebowitz and Margolis (1999); Klein (2001); Gilbert and Katz (2001).

[2]See the works previously cited and, specifically on the tying issues, Whinston (1990, 2001).

[3]This also explains why Microsoft only 'integrates/ties' certain applications and is happy for most software to be produced by third-party vendors. The need to tie only arises when that application or service will itself be the site of significant third-party development. This is clearly the case with web-browsers, as Bill Gates presciently saw in his 'Internet Tidal Wave' memorandum: "A new competitor 'born' on the Internet is Netscape [Netscape was launched 15th Dec 1994]. Their browser is dominant with 70% usage share,

Another example is provided by the 2000 case of eBay vs. Bidder's Edge.[4] Here, eBay, an online auction site, successfully sued Bidder's Edge, a firm which collected together prices from different auction site for consumers to compare, for cyber-trespass, ostensibly on the grounds that Bidder's Edge spidering activities caused excessive load on their servers. However, as various commentators pointed out the ability to exclude a firm such as Bidder's Edge could also have serious anti-competitive effects[5]. EBay is a classic example of a platform in a two-sided market with sellers taking the role of 'software' or 'service' and buyers that of consumers. If a third-party were easily able to transfer (port) sellers from one auction platform to another then eBay's market power would be greatly diminished. A firm such as Bidder's Edge would greatly facilitate such 'porting' by ensuring that a given seller (and their associated 'reputation') would be visible to consumers no matter what auction platform they were on. By preventing Bidder's Edge (and any other similar firm) from being able to extract data from the eBay site without permission eBay obtained very substantial control of porting from its platform.

A final example comes from the ongoing debate in Europe around interoperability of TPMs/DRMs (Technological Protection Measures/Digital Rights Management systems) systems, particularly in relation to Apple's dominant position with its iPod and iTunes systems both of which use Apple's proprietary 'FairPlay' DRM. Here the platform is the digital music player and the 'software' is the music with Apple's offerings being the iPod or iTunes software on the platform side and the iTunes Music Store (ITMS) on the 'software' (music) side of the market. If DRM were interoperable then you could play a song from any given digital music store on any given digital music player. However with no interoperability if someone buys all their songs from the iTunes Music Store (currently with 70-80% of the digital downloads market) then they can only play them on an iPod (and if they change music player they may lose all their purchased music). Without this obstacle it would be substantially easier for consumers to switch platforms (i.e. digital music players). Thus, by having a closed DRM and integrating backwards into the 'software' (music) market (analogously to the previous Microsoft examples) Apple are

allowing them to determine what network extensions will catch on. They are pursuing a multi-platform strategy where they *move the key API into the client to commoditize the underlying operating system ...*" (emphasis added).

[4]EBAY, Inc vs. BIDDER'S EDGE Inc, http://pub.bna.com/lw/21200.htm.

[5]See, for example, the amicus curiae brief filed by a collection of 28 law professors available online at http://www.gseis.ucla.edu/iclp/ebay-ml.

able to effectively control porting and thereby increase their market power in the platform market.[6]

The paper builds several strands in the existing literature. In general, these relationships will be highlighted as appropriate in the main section of the paper. However, it is worth mentioning two particular items here. First, there is a sizable and growing body of work on two-sided markets and indirect network effects in general which is surveyed in Rochet and Tirole (2005, 2003). Second, there is existing work on 'converters' in network markets (converters being devices that allow a user on one network to gain access to a separate network). As porting can be seen as the analogous activity in a two-sided market with 'indirect network effects' to converters in the original one-sided models there is a close relation between our work and these existing papers. For example, Farrell and Saloner (1992) examine the provision and purchase of imperfect converters in a network effects model, as well as the incentive for a dominant firm to make conversion costly.[7]

2. The Model

There are two platforms/networks: $X = A, B$ and a mass of consumers modelled by the interval $[0, 1]$ with the index, $t \in [0, 1]$, used to label them.

Two types of product are provided for each platform/network: the 'hardware' platform itself and associated 'software/services'. The terms 'hardware' and 'software' should not be construed literally but rather to indicate the complementary nature of two types of good. The terms will continue to be used very frequently in the rest of the paper, often without the cautionary quotes, so the reader should keep this clearly in mind.

Consumers must purchase one unit of a network's hardware to be able to use the associated software and consumers derive utility only from from the consumption of software (the hardware itself has no value). If a consumer has already purchased hardware and software from one network she gains no extra utility from purchasing from a second network so a consumer will purchase from at most one network (there is no multi-homing). We also make the traditional assumption that all consumers join one or other network.

[6]It is important to note for this analysis that it is well-known that Apple make their profits on the hardware (the iPod) and make very little from the iTunes Music Store.

[7]See also Choi (1997) for another converter model, albeit a dynamic one related to the transition from an old to a new technology.

The measure of consumers on network X is denoted by n_X and the number of software firms on network X is s_X.

Consumers have the following utility function:

$$u_X(t, p_X, s_X, p^s_X) = \phi - p_X - h_X(t) + u^s_X(s_X, p^s_X)$$

Where

- ϕ is a positive constant introduced so that reservation utility can be normalized to 0 (alternatively one could remove ϕ from utility function and set reservation utility to $-\phi$)
- p_X is the price of hardware on network X
- $h_X(t)$ models consumer heterogeneity. It is assumed that heterogeneity is symmetric across networks that is, $h_B(1-t) = h_A(t)$. This allows one to write $h_A(t) = h(t) = h_B(1-t)$. We shall assume the standard 'orderability' of consumers by heterogeneity, i.e. $h'(t) > 0$. Thus we have a standard linear city model with platform A at 0 and platform B at 1 and consumers preferring, all other things being equal, a closer platform.
- u^s_X is utility from software purchases with s_X the amount of software available on network X and p^s_X the price (or vector of prices) of software. This is discussed further below.

Hardware on network A is controlled by a single firm, the monopolist (M). Hardware on network B is provided competitively. Hardware fixed costs are assumed to be sunk and therefore may be taken without loss of generality to be zero. Marginal costs, c, are constant and the same for each type of Hardware. Since network B's hardware market is perfectly competitive its price equals marginal cost: $p_B = c$. Since the marginal cost is common across the two networks we may, without loss of generality, set $c = 0$.

2.1. **Software Production and Porting.** We take a general approach in which we assume only that software production on platform X involves (a) some form of fixed costs (f_X) (b) that the amount and price of software on platform X may be expressed solely in terms of these fixed costs, f_X and the number of consumers on the platform, n_X. Taken together these mean that the consumer software utility function has a reduced form of the following kind:

$$u^s_X(s_X, p^s_X) = u^s_X(s_X(f_X, n_X), p^s_X(f_X, n_X) \equiv \nu_X(n_X) = \nu(f_X, n_X)$$

We shall term ν_X the 'indirect network effects' function on platform X.[8] By proceeding in this manner the results are kept as general as possible. Furthermore, the two basic models of imperfect competition with fixed costs (monopolistic competition and product differentiation) can both be shown to give rise to this reduced form (the appendix on software production provides an explicit derivation for the case of a standard circular city model of product differentiation).

The software that is produced may be created by two methods. Either it can be created directly for network X at fixed cost f^d_X or it can be ported from the other network at fixed cost f^p (note that this only relates to the fixed cost, the marginal cost is the same whether the software is ported or created directly).

In our model we will suppose that a monopolist may increase the cost of porting from its platform to a competitor's – though at the cost of some expenditure on its own part. Formally, if e is expenditure then $f^p = f^p(e)$. It will be convenient in what follows to have the porting cost, f^p, being the choice variable rather than expenditure, e. This simply involves using the inverse function (the expenditure to prevent porting), $e = e(f^p)$. Efforts to prevent porting display diminishing returns so $e'(f^p) > 0, e''(f^p) > 0$.

Thus the fixed cost of software production on a network, f_X, will be either: f^d_X if all software is produced directly (none is ported); a mixture of f^d_X and f^p if some software is ported and some produced directly; or f^p if all software is ported.

2.2. **Sequence of Actions.**

(1) The monopolist, M, chooses values for control variables: p_A, f^p.
(2) Software producers for each network form expectations of network size. Based on these expectations they decide whether to engage in software production (be it via porting or direct production).

[8]Note that we implicitly assume some symmetry across platforms in that the function ν is the same across the two platforms.

(3) Taking the resulting level of software provision as given consumers solve their utility maximization problem and decide from which network to purchase.
(4) The resulting network sizes should be consistent with rational expectations. That is: actual and expected network sizes are equal and actual and expected software levels are equal.
(5) M's profits, $\Pi = p_A \cdot n_A(p_A, f^p) - e(f^p)$, are determined.

Remark: because of the imposition of rational expectations the order in which software firms and consumers move does not affect the outcome of the model. Thus we could as easily have software firms taking their decisions after consumers or even simultaneously.

3. Solving the Model

As presented we now have a standard two-sided model with utility functions:

$$u_X(t, p_X, f_X, n_X) = \phi - p_X - h_X(t) + \nu(f_X, n_X)$$

We can solve this in the usual manner to obtain platform sizes as a function of the monopolist's choice variables: $n_A = n_A(p_A, f^p)$. The monopolist then solves: $\max_{p_A, f^p} p_A n_A(p_A, f^p) - e(f^p)$.

3.1. Solving for Network Equilibrium.

Solving for Network Equilibrium. To solve for equilibrium network size we proceed by the usual method based on finding the marginal consumer indifferent between the two platforms.

First, recall that we have assumed that consumers gain no extra utility from purchasing from more than one network. Thus, we may assume that consumers purchase at most one set of compatible hardware and software. We further assume that all consumers do purchase from one or other network. Thus we have $n_B = 1 - n_A$ and we need only consider n_A in what follows.

Define: the *conditional utility advantage* of network A over network B for consumer t when network size is n_A:

$$\hat{A}(t, n_A) = u_A(t, n_A) - u_B(t, 1 - n_A)$$

and the *utility advantage (function)*, which gives the utility advantage of network A over B if t is the marginal consumer (so $t = n_X$):

$$A(t) = \hat{A}(t, t)$$

Lemma 1. *The set of equilibria of the model as presented above are given by $E = E_0 \cup E_{-0}$ where E_0 is the set of interior equilibrium, $E_0 = \{t : A(t) = 0\}$, and E_{-0} is the set of extremal or 'standardization' equilibrium in which all consumers join one or other network, $E_{-0} = \{0 : A(0) < 0\} \cup \{1 : A(1) > 0\}$. An equilibrium $t_e \in E_0$ is stable if $A'(t_e) < 0$. All $t_e \in E_{-0}$ are stable.*

Proof. See appendix. □

Using the expression for the utility function we have that:

$$A(t) = -p_A - h_A(t) + h_B(t) + \nu(f_A, t) - \nu(f_B, 1 - t)$$

3.2. **Porting.** In this section we shall determine the amount of software produced for each network of the various possible types (produced directly, ported or produced by a mixture of those methods). In doing so, we will also have determined the 'actual' fixed cost of software production for each network f_A, f_B in terms of the fixed cost of directly producing software for that network and the (common) porting cost (f_X^d, f^p).

Lemma 2. *In equilibrium only one network has software produced directly for it. All the software on the other network derives from porting. Let us denote the first network for which software is produced directly by X and the other by X'.*

Then the amount of software on X' will be equal to the smaller of 1) the amount of software on X (in the case where all software is ported) or 2) the 'unconstrained' level software production, i.e. that which would be produced with $f_{X'} = f^p$. If the first case obtains, i.e. all possible software is ported, the porting constraint will be said to bind.

Finally we have $f_X = f_X^d$ and, if the porting constraint does not *bind, $f_{X'} = f^p$.*

Proof. See appendix. □

We now make two assumptions. These assumptions are weak and are here to ensure that the situation we analyze is both realistic and interesting.

Assumption: There exists an asymmetric stable equilibrium where network A is larger than B.

Justification: in most real world situations one network is larger than the other. Furthermore, in any situation with antitrust considerations this will necessarily be the case.

Assumption: In the case of asymmetry it is the network with larger (expected) size for which software is produced directly.

Justification: In previous section on porting it was shown that it will always be the case (in this model) that software on one network has all software produced directly and one has all software ported. Since the amount of software on the 'porting' network must always be less than or equal to that on the 'direct-production' network it is natural to assume that it is the network with larger (expected) size for which software is produced directly.[9]

Combining these assumptions with the results of the previous section we may set $f_A = f_A^d$ and $f_B = f^p$ (though we will also need to check that the porting constraint does not bind). As already discussed the monopolist may control the cost of porting from its network so the profit maximization problem becomes:

$$\max_{p_A, f^p} p_A \cdot t_e(p_A, f^p) - e(f^p)$$

Remark: The result that, for any given platform, all software is either produced directly or ported may seem a little implausible. After all, in reality, we usually see software produced directly for all platforms. It also usual for there to substantial porting, with the same piece of software available on multiple platforms (one could see this as multi-homing on the software side). Such results would be obtained by a simple extension to our model in which there is heterogeneity in the fixed costs of direct production for a given platform (for example, one could postulate that costs follow a uniform distribution of measure N over $[0, f_X]$). However, all that is necessary for the results in the rest of this paper is that the *marginal* piece of software on the platform competing with the monopolist is ported – and that, as a result, altering the porting costs affects the amount of software on that platform. Thus, while the model as given may seem to be overly restrictive in its

[9]In fact if networks displayed symmetry, i.e. direct production costs are equal and heterogeneity functions on the two networks are the same, this is a result rather than an assumption.

implications the necessary result, that is that the porting constraint binds, will still hold in the more general case.

3.3. **Example I: Equilibrium and Demand.** Below we will derive various general results about the existence, stability, and welfare properties of equilibria. However, first it will be worthwhile to examine a specific case graphically in order to aid intuition. The situation we shall consider is one in which the two networks are equivalent, that is the fixed costs of software production on the two networks are equal and heterogeneity is symmetric ($h_B(1-t) = h_A(t)$). For the 'network effects' function we use the reduced form derived from a circular city model (see appendix), that is $\nu(f,t) = C - \sqrt{\frac{f}{t}}$. We set the heterogeneity function to be $h_A(t) = 10t^{10}$. This corresponds to a situation where there is a large middle ground of consumers who are fairly indifferent between the two platforms (h(t) is small until t is close to 1) but two 'extreme' groups at either end who have strong preferences for their nearest platform. Set fixed costs as follows $f_B = f_A = 1.5$. These values are chosen so as to generate a stable asymmetric equilibrium as shown in Figure 1. Note that in its general shape (i.e. number of equilibria, location of maxima/minima) this graph is the simplest possible that gives rise to a stable asymmetric equilibrium.[10]

3.3.1. *Discontinuity of demand:* since price enters $A(t)$ linearly the diagram above also implicitly defines the demand function in the neighbourhood of an equilibrium (an increase in the p_A shifts the $A(t)$ curve down by that amount). A maximum of $A(t)$ therefore corresponds to a point at which demand is discontinuous (as price rises above the maximum value demand jumps down as the market tips to the neighbourhood of next lowest stable equilibrium).

An illustration of this is provided in Figure 2, which plots the demand function derived from Figure 1 in the neighbourhood of the stable equilibrium at 0.84. Here demand is discontinuous at a price just below 0.5 (i.e. at the left edge of the diagram – the discontinuity itself is not shown as it distorts the scale). At the discontinuity demand will suddenly jump down to approximately 0.14 which is the next place the line y=0.5

[10]To have an interior stable equilibrium $A(t)$ must intersect the line $y = 0$ from above. If heterogeneity is symmetric, $h_A(t) = h_B(1-t) = h(t)$ then when fixed costs are equal and prices are zero, $A(t)$ must be anti-symmetric about 0.5, i.e. $A(t) = -A(1-t)$. This implies $A(0.5) = 0$ so 0.5 is an equilibrium. Thus with symmetry in the network function and assuming that standardization equilibria exist (i.e. 0 and 1 are equilibrium) the fewest crossings (i.e. interior equilibria) that lead to the existence of a stable asymmetric equilibrium is five and we must have a situation similar to that shown.

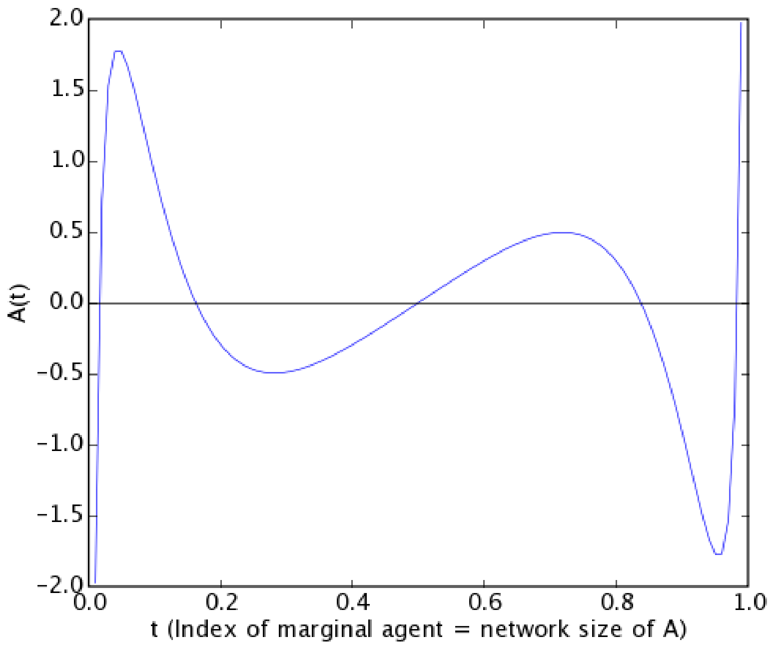

FIGURE 1. The utility advantage function, $A(t)$ in the symmetric case when the access prices for the two networks are the same. There are stable equilibria at 0 and 1 (the 'standardization' equilibria) and 0.16 and 0.84 (asymmetric stable equilibria). There are unstable equilibria at 0.5 and 0.02 and 0.98.

would intersect $A(t)$ (see Figure 1). Note how this diagram is just the relevant portion of Figure 1 between 0.73 and 0.84 'blown up'.

In all cases where there is symmetry and a stable asymmetric equilibrium $A(t)$ must have a bounded maximum just like it does in Figure 1. A bounded maximum in turn implies a discontinuity in the demand function of the monopolist. Thus, in all such cases, **a monopolist will face a discontinuous demand function**. This discontinuity in demand does not exist in the traditional linear network effects models and it functions here to place a sharp upper bound on the price the monopolist can charge without a sudden jump downwards in market share.

3.3.2. *Other Comparative Statics:* we can evaluate the effect of changing production and porting costs by considering how it shifts $A(t)$. In particular, increasing fixed costs of software production for A f_A will shift $A(t)$ down and increasing f_B will have the opposite effect (note that f_B is equal to the porting costs, f^p if the porting constraint does not

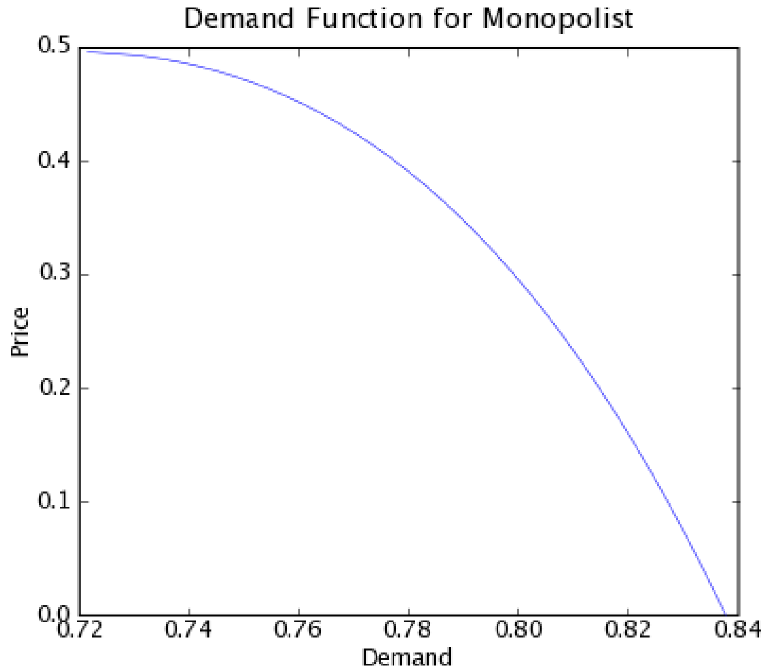

FIGURE 2. The Demand function for the monopolist in the neighbourhood of the stable equilibrium at 0.84. Demand is discontinuous at a price just below 0.5 (i.e. at the left edge of the diagram – the discontinuity itself is not shown as it distorts the scale).

bind). Note that unlike price, fixed costs do not enter linearly so they will also change the shape of $A(t)$ and not just its level.

3.4. **Properties of Equilibrium.** The insights gained in relation to this special case can be distilled into a general result:

Lemma 3. *Noting that the advantage function implicitly depends on all of our exogenous and choice variables: $A(t) = A(t, p_A, f_A, f_B)$ (and therefore so does the set of equilibria $E = E(p_A, ...)$), then, having picked a stable equilibrium $t_e^0 \in E_0(p_A^0, ...)$ we have associated to it a well-defined, continuous and differentiable 'equilibrium function' $t_e(p_A, f_A, f_B)$ defined in a neighbourhood of t_e^0. In particular, restricting to changes in p_A we have a demand function:*

$$q(p_A) = t_e(p_A) = A^{-1}(p_A)$$

Differentiating we have:

(1) *Downward sloping demand schedule:* $\frac{dq}{dp_A} = \frac{-1}{A'(t_e(p_A))t_e'(p_A)} < 0$
(2) $\frac{dt_e}{df_A} < 0$

(3) $\frac{dt_e}{df_B} > 0$

Finally, though demand is discontinuous, there exists locally a unique profit maximizing price.

Proof. See appendix. □

4. Welfare

Having established the various properties of equilibrium in this section we come of the central questions of this paper: how does the monopolist's control of prices and the cost of porting affect consumer and social welfare? Giving equal weight to monopoly profits and consumer welfare we have that total welfare, $W = \Pi_A + W^C$ where W^C is consumer welfare and Π_A are the monopolist's profits.

Lemma 4. *When network's A size is x the marginal change in consumer welfare as a function of network A's size is:*

$$\frac{dW^C}{dx} = A(x) + \mu(x)$$

where $A(x)$ is the utility advantage of A over B defined previously

$$\mu(x) \equiv x\nu'_A(x) - (1-x)\nu'_B(1-x)$$

At an interior equilibrium $t_e \in (0,1), A(t_e) = 0$, and this reduces to:

$$\frac{dW^C}{dx} = \mu(t_e)$$

Proof. See appendix. □

Remark: A first point to emphasize is that this result (and Lemma 5 below) are entirely general and will hold in *any* model in which consumer utility incorporates a 'network effects' function (whether arising directly, or, indirectly as a reduced form derived from a more complex model). That is, there is nothing that depends on the specifics of the porting framework as presented in this paper. In particular, these resultss would apply both traditional direct network effects models of communication networks and some of the more recent models arising from a two-sided market structure.

Remark: This result means that, when at an interior equilibrium ($x = t_e$), the marginal change in consumer welfare with respect to network size is a function of 'network effects' alone (encapsulated in μ). The two basic possibilities, namely that consumer welfare is increasing ($\mu(t_e) > 0$) or that it is decreasing ($\mu(t_e) < 0$) with the size of network A have a simple interpretation. In the first case we have a situation in which more standardization (that is more consumers on network A) is preferable. In the second case we have a situation in which more symmetrical network shares are preferable. (There is, also the third possibility that the change in consumer welfare is zero).

We can in turn relate the value of $\mu(t_e)$ and therefore whether standardization or symmetry is preferable to the rate of diminishing returns to network size displayed by the network effects function. To illustrate consider the case where $\nu_A = \nu_B = \nu$ and we have an equilibrium $x = t_e > 0.5$.[11] Furthermore let ν takes a simple polynomial form $\nu(x) = C + k^{-1}x^k$, then:

$$\mu(x) = \begin{cases} > 0, & \nu(x) = C + x^k, k > 0 \text{(Diminishing returns are relatively weak)} \\ 0, & \nu(x) = C + ln(x) \\ < 0, & \nu(x) = C - \frac{1}{x^k}, k > 0 \text{(Diminishing are relatively strong)} \end{cases}$$

Seen in this light, $\mu > 0$ corresponds to 'network effects' which only diminish gradually with network size while $\mu < 0$ corresponds to a situation in which diminishing returns to network size are relatively strong. The natural logarithm here is the dividing line between the two cases. The classic form studied in the literature is of course where $k = 1$ and ν is linear (so $\mu > 0$) while the circular city model of indirect network effects studied in the appendix gives rise to the case where $k = -0.5$ and so $\mu < 0$.[12]

Thus, network effects which display weakly diminishing returns imply that a standardization-type network configuration (everyone on one network) will be preferable. Conversely, if

[11] Of course we are dealing with the 'asymmetric' case where production costs on the two networks are not equal and so $\nu_A(x) = \nu(f_A, x) \neq \nu(f_B, x) = \nu_B(x)$ (the fixed cost of porting – which is the cost of production of software on B – is less than the cost of production on A). However, similar results will still hold with some minor modifications in the 'asymmetric' case.

[12] Odlyzko and Tilly (2005) provide a thoughtful critique of existing assumptions regarding the form of the network effects function such as that embodied in Metcalfe's law (Metcalfe's law corresponds to the linear case $\nu(x) = x$). Interestingly, as a replacement they propose using the logarithmic form, $\nu(x) = ln(x)$. As we have just shown this is a very special case in which at a network equilibrium we have $\mu = 0$ and therefore consumer welfare is neither increasing or decreasing in network size. Clearly, one would like to determine the exact form of the (indirect) network effects function empirically. However, at least to our knowledge, there are no *economic* papers which deal with this issue.

network effects show strongly diminishing returns, a more symmetric network configuration is preferable.

Lemma 5. *At a network equilibrium, t_e, the effect on consumer welfare of an increase in the price charged by the monopolist is negative if $\mu(t_e) \geq 0$ and is ambiguous otherwise depending on the relative magnitudes of the monopoly pricing effect (-ve) and the network externality (+ve). Furthermore, at a full equilibrium (i.e. a network equilibrium where the monopolist is profit-maximizing) the change in total welfare equals that in consumer welfare and therefore has the same properties.*

Proof. See appendix. □

Remarks: Monopoly pricing does not result in traditional deadweight losses since total demand is fixed and does not change (consumers who leave one network join the other).[13] However, it does shift consumers away from the monopolist's platform (an effect exacerbated by the feedback from the indirect network effects). In market's with 'externalities' such as these this will have consequences for welfare.

The effect of an increase in the monopolist's price depends on two distinct factors. The first factor is the simple one that higher prices reduce consumer welfare because consumers pay more. The second factor is more subtle. An increase in M's price moves consumers off A onto B. This effect may either be negative or positive depending, respectively, on whether a more standardization-type or a more symmetric network configuration is better for welfare. As shown in Lemma 4 this second condition is equivalent to asking whether $\mu(t_e)$ is positive (standardization-type better) or negative (symmetric better). Thus, if $\mu(t_e)$ is positive, an increase in the monopoly price by reducing the size of network A acts to reduce welfare. Conversely when more symmetric network sizes are preferred then an increase in the monopoly price by reducing the size of network A actually acts to increase welfare.

If we combine the two factors then we only get an unambiguous prediction (increase in prices reduces welfare) in the first case, that is when a more standardization-type network configuration is preferable. In the second case, where a more symmetric network

[13]This explains why at full equilibrium marginal consumer welfare and total welfare are equal.

configuration is preferable, the effect will be ambiguous and welfare could actually rise due to an increase in the monopolist's prices.

Lemma 6. *At a network equilibrium, t_e, if $\mu(t_e)$ is negative (strongly diminishing network effects) then an increase in porting costs will unambiguously reduce consumer welfare. If $\mu(t_e)$ is positive (weakly diminishing network effects) then the effect on consumer welfare is ambiguous in general and will depend on the relative magnitudes of the welfare loss from a direct reduction in software provision on platform B and the welfare gain from an increase in A's market share. Furthermore, at a full equilibrium (i.e. where the monopolist is profit-maximizing) the marginal effect on total welfare again equals the marginal effect on consumer welfare.*

Proof. See appendix. □

Remarks: Again we have two distinct effects of higher porting costs. The first, and the direct one, is that higher porting costs result in a reduction in availability of software for those on platform B (and probably higher prices too – though this may depend on the specifics of the model for software provision). This unambiguously reduces welfare because higher porting costs mean less software for B users (holding network B's share constant).

The second effect arises from the fact that, as a result of the change in software availability on B, some consumers move from platform B to platform A. This change is an exactly similar one to that already analyzed above when discussing the effect of a price rise (except here an increase porting cost increases the size of network A while an increase in price reduces the size of network A). In particular the effect will be negative if, and only if, $\mu(t_e)$ is negative (more symmetric network configuration preferred). In this case, both effects operate in the same direction and an increase in porting cost is unambiguously harmful to consumer welfare. On the other hand if a more standardization-type network is preferable ($\mu(t_e) > 0$) then this effect is positive and the overall impact on welfare will depend on the relative magnitude of the two effects. In this second ambiguous case, it is more likely that the welfare effect is negative:

- The larger is platform B's market share (more consumers to suffer from the reduction on software provision on B)

- The larger is the direct impact of higher porting costs on the provision of software for B (greater reduction on software provision on B).
- The smaller is the impact of changes in porting cost on A's market share.[14]
- The smaller is the increase in consumer welfare of an increase in A's market share.

5. Example II: Welfare

We now return to our previous specific example, this time in order to illustrate the welfare analysis. Using it, among other results, we demonstrate that it is possible for the welfare costs (consumer or societal) of the control of porting to be significantly greater than the costs of monopoly pricing.

We first choose specific functional forms and values for constants. The heterogeneity function is chosen to ensure that there exists an asymmetric stable equilibrium and is the same as that used for figure 2 above: $h(t) = 10t^{10}$.

The direct costs of software production are set to $f_A = 1.5$ and the initial porting cost is set to two-thirds of that value, so $f^p = 1.0$. The monopolist's expenditure function is: $e(f^p) = 2 \cdot (f^p - 1)^4$ and the initial value of f^p when there are no efforts by the monopolist is set to 1. The expenditure function displays diminishing returns and while initial efforts to prevent porting are relatively cheap the cost then escalates rapidly.

The exact parameters for the functional form of the expenditure function are chosen so that an interior 'porting cost' solution exists i.e. the value of porting cost obtained is such that $f_A > f^p$ and expenditure to prevent porting is non-zero and non-infinite. Using these values we can now proceed to solve the monopolist's problem by numerical means and have the following results.

We find the values chosen for the two control variables are 1.419 for porting costs and 0.43 for the price of hardware on A. We also calculate the profit-maximizing price M would charge when unable to influence porting costs: 0.079. Our main interest is in the significance of M's choices for welfare and welfare outcomes. These, along with the values of other significant variables, are presented in Table 1 (NB: since ϕ is an arbitrary constant it has been set so that initial welfare values are normalized to zero. This value has no

[14]For example, if the main effect of changes in porting cost were to soften competition rather than to directly increase A's market share. That is, in terms of A's demand curve, increasing porting costs steepened the demand curve or shifted it up but did not shift it out.

	Porting Cost	**Price of A Hardware**	**A's market share**	**Net Profits for M**	**Consumer Welfare**	**Total Welfare**
Initial porting cost, competitive prices	1.0	0	0.758	0	0.0	0.0
Initial porting cost, monopoly price on A	1.0	0.079	0.704	0.056	-0.046	0.010
Monopolist chosen porting cost, monopoly price on A	1.419	0.43	0.729	0.252	-0.406	-0.154

Table 1. Welfare Results at Various Prices and Porting Costs

significance since, as already explained, welfare can be changed by a fixed constant (ϕ). Thus only the sizes of welfare *changes* can be meaningfully compared.)

The first line is there to show the baseline case, when the control parameters are at their 'default' values (that is without intervention by the monopolist). In this case, M's market share, with its own price at zero and the fixed costs of porting at 1, is still 75%. Total welfare and consumer welfare are the same – since prices are zero – and has been normalized to zero.

The next line shows the situation if the monopolist can only set prices and is not able to influence porting costs. This helps us benchmark the relative gain to a monopolist of being able to influence porting costs in addition to setting prices. In line with theory the welfare change is slightly positive, reflecting the reduction in the size of Network A.

The final line shows the actual outcome with the porting cost and price at the level chosen by M to maximize its profits. Porting costs increase by almost a half to 1.42, nearly reaching the same levels as the cost of direct production (1.5). Prices rise by over five times compared to the situation when porting costs can not be altered demonstrating the large impact of the Monopolist's control of porting. Despite the far higher price, market share for the monopolist rises though it is still lower than in the situation where neither price nor porting cost can be set.

5.1. **The Monopolist's Profits.** M gains dramatically from the ability to manipulate porting costs, the percentage increase in profits being approximately 400% over what is obtained when porting costs are fixed. Moreover this is net of the costs incurred to prevent porting, $e(f^p) = 0.0616$, which are equal to a fifth of gross profits. The main effect of raising porting costs is not to increase market share but to soften competition between the two platforms and therefore permit a much higher profit-maximizing price to be charged. Market share at the monopoly price in the two cases when porting cost is and is not manipulatable are quite close (0.704 vs. 0.729).

5.2. **Consumer welfare.** The change in consumer welfare from monopoly pricing, $\Delta W_c^M = -0.046$. The change resulting from higher pricing and higher porting costs is $\Delta W_c^{Mf} = -0.406$. Thus consumer welfare losses arising from the combination of higher porting costs and higher prices are almost **nine times** as large as those arising from higher prices alone.[15]

5.3. **Total welfare.** For total welfare increasing M's price will actually increase welfare: with porting cost at 1, $\Delta W^M = 0.01$. However the welfare change due to the combination of monopoly pricing and higher porting costs is decidedly negative $\Delta W^{Mf} = -0.156$. Thus for this case welfare costs go from barely positive to significantly negative.

6. Conclusion

In this paper, we introduced 'porting' into a standard, two-sided, indirect network effects model, with 'porting' playing a role analogous to 'converters' in the simpler direct network effects models. With 'porting', software developed for one can be converted to run on another network (usually at a cost lower than that of direct production). We examined general properties of this model, looking, in particular, at what occurs when one (dominant) network is controlled by a single firm, the Monopolist, who is able to control the cost of porting to a competitor network (at the cost of some expenditure on the Monopolist's part). We demonstrated the existence of a network (and porting)

[15] As already stated, as welfare is only defined up to a constant we can only compare changes in welfare and not levels. Nevertheless, utility is money metric (prices enter linearly) and profits are well-defined so it *is* possible to convert of welfare *changes* into monetary terms. As a very simple 'back-of-the-envelope' calculation consider applying this analysis to the Microsoft case. Profits in 2000 (around the time of the antitrust settlement in the US) were approximately \$9.5 billion and in our model profits equal 0.252. Thus, in dollar terms the change in consumer welfare from monopoly pricing alone equals approximately \$1.7 billion ($0.046/0.252 \cdot 9.5$), while the change in consumer welfare with both higher prices and higher porting costs equals \$15.3 billion ($0.406/0.252 \cdot 9.5$).

equilibrium and examined various associated properties, such as the discontinuity in the monopolist's demand function.

Next we turned to the question of consumer and social welfare. It was shown that, the effect on welfare both of monopoly pricing and higher porting costs depended crucially on the degree of diminishing returns to network size in the indirect network effects function (ν). If diminishing returns were weak then monopoly pricing had a negative effect on welfare but the effect of the higher porting costs was ambiguous, while with strongly diminishing returns the converse held, that is the effect of monopoly pricing was ambiguous but higher porting costs had a negative effect.

Finally, we provided an illustrative example using a specific case of our model. We showed that, in this example, the social and consumer welfare losses arising from the control of porting combined with monopoly pricing dwarfed the welfare effects stemming from monopoly pricing alone. In particular, consumer welfare losses from the combination of higher porting costs and higher prices were over nine times higher than those arising from higher prices alone. For total welfare, there was almost no effect of monopoly pricing alone but a significant reduction when the monopolist controlled both prices and porting costs (in this second case the welfare loss was equal to approximately three fifths of the monopolist's profits). Of course this is a single example and without either calibrating from empirical data or extensive robustness-checking one would not wish to use the results for policy-making. Nevertheless, it does provide a useful example that helps put flesh on the dry bones of the general model.

These results, taken together, have important consequences for competition policy. They demonstrate how, in a two-sided market environment, anti-competitive behaviour may manifest indirectly through actions taken to control porting rather than through direct tying or pricing behaviour. Furthermore, for the monopolist the benefits of controlling porting may also accrue indirectly: that is, by increasing the prices that can be charged at a given level of demand rather than increasing demand. Returning to the examples discussed in the introduction, we would suggest that an analysis based on the control of porting provides a better way of understanding the effects and motivations of a dominant firm than alternative approaches, such as those based on traditional theories of tying or even switching costs.[16]

Specifically our model suggests that policy-makers should endeavour to take steps to reduce the control of porting by a dominant firm. One simple way to do this is to promote 'open standards' at the interface between the 'software/service' and the 'hardware' platform. For example, in the case of TPMs/DRMs (Technological Protection Measures/Digital Rights Management systems) a policy-maker could promote (or require) interoperability between different TPM/DRM systems so that the music ('software' in our terminology) purchased from any given vendor will work on any given digital music player (the 'hardware' platform).[17] Similarly, in the case of the EU dispute with Microsoft over Microsoft's Windows Media Player, rather than requiring unbundling the authorities could simply require that any audio formats specific to Windows Media Player must be 'open' and freely licensable so as to ensure that it is easy to port music and complementary services to a media player on another platform such as Linux. The same approach would also apply to web browsers where there already exist an extensive set of open standards developed by the W3C. Again, rather than requiring Microsoft to unbundle Internet Explorer the authorities could simply press for 'standards-compatibility'. In this way developers of websites and other forms of web-services would be able to develop in a platform-neutral way (essentially the cost of porting to a different platform such as Linux+Firefox would then be zero) with all the associated long-run benefits for competition and consumer choice.

Finally, we mention some of potential avenues for future work. One of the most obvious improvements that could be made would be to replace the simple monopoly model with an oligopoly in which each platform has a profit-maximizing owner. Porting, and the manner in which it may be controlled, have been modelled in a fairly simple manner. One might improve this in various ways. For example, one could change from a 'black box' cost function e to a setup where f_A increases with f^p – this would correspond to an 'obfuscation'

[16]Though, of course, in one sense the control of porting can be seen as a special case of tying (or the creation of switching cost) in which the 'tie' is not aimed at competing providers of the tied good but at the owners of competing *platforms*.

[17]At the present time this very issue of DRM interoperability is being debated both at the EU level and in various individual European countries in relation to Apple's FairPlay DRM.

situation where increasing porting costs to competitor platforms also increases the cost of producing software on one's own platform.

One could also add dynamics to the model (though this would also greatly increase complexity). For example, rather than having a fixed static demand one could allow consumers to arrive over time.[18] Alternatively consumers could make repeat purchases but with a switching cost if a different network were chosen in a subsequent period.

Finally, it would be interesting to explore the consequences of allowing for innovation in software provision perhaps via the introduction of a quality ladder. Such an approach would raise additional thorny questions about the welfare impact of monopolist behaviour if innovation were not barrier to entry neutral. For example, if innovations while increasing quality also made it easier to port from one platform to another (consider the case of Java or the emergence of the web and web browsers as a fully-fledged application development platform).[19] In this case, efforts to obstruct porting would also hinder innovation, with all the attendant consequences for welfare.

Appendix A. Proofs

A.1. **Proof of Lemma 1.** Recall that the *conditional utility advantage* of network A over network B for consumer t when network size is n_A:

$$\hat{A}(t, n_A) = u_A(t, n_A) - u_B(t, 1 - n_A)$$

and the *utility advantage (function)*, which gives the utility advantage of network A over B if t is the marginal consumer (so $t = n_A$):

$$A(t) = \hat{A}(t, t)$$

Suppressing n_A for the time being we shall simply write $\hat{A}(t)$.

Since 'heterogeneity cost' for a consumer is increasing in the distance of the consumer from the chosen network we have that $\forall t, \hat{A}'(t) < 0$. Then $\hat{A}(t_m) > 0$ implies $\hat{A}(t) > 0, \forall t \leq t_m$. Conversely if $\hat{A}(t_m) < 0$ then $\hat{A}(t) < 0 \forall t \geq t_m$.

[18] This might result in limit-pricing behaviour by the monopolist similar to that in Fudenberg and Tirole (2000).

[19] See e.g. Farrell and Katz (2000) on network monopolies and downstream innovation.

Now a consumer (with expectations of network A size equal to n_A) chooses network A over B iff $\hat{A}(t) \geq 0$. Thus if a consumer with index t_m chooses network A then all consumers with index $t \in [0, t_m]$ choose network A. Similarly if a consumer with index t_m chooses network B then all consumers with index $t \in (t_m, 1]$ choose network B.

In particular this immediately implies that if there exists $t_m \in [0, 1], \hat{A}(t_m) = 0$ (and there is at most one such solution since $\hat{A}' < 0$) then this is the marginal consumer and the resulting network size of A is t_m. This is because for $t \in [0, t_m], \hat{A}(t) > 0$ so these consumers choose network A while for $t \in (t_m, 1], \hat{A}(t) < 0$ so these consumers choose network B.

For the extremal cases by the same arguments if $\hat{A}(0) < 0$ then all consumers choose network B and if $\hat{A}(1) > 0$ then all consumer's choose network A.

Furthermore, only one of these alternatives is possible so there is a unique implied network size for any given assumed n_A. Thus one may define a function $f : [0, 1] \to [0, 1]$ where for a given assumed network size, $n, f(n)$ is the resulting implied network size.

Imposing rational expectations then implies that n_A is an equilibrium if and only if n_A is a fixed point of f. But n_A is a solution of $f(n) = n \Leftrightarrow n_A \in E$. QED

Remark: Equilibria $t \in E_{-0}$ are often termed standardization or tipping equilibria as they involve all consumers joining a single network.

Remark: This result sets up an implicit equivalence between network size and the marginal consumer (where the term marginal is broadened to include the tipping situations where $t_m = 0$ or 1 and $A(t_m) \neq 0$

Stability of Equilibria: Suppose we have equilibrium $t_m \in E_0$ with $A'(t_m) < 0$. Suppose that there is a perturbation in expectations so that a network size of $t_m + \epsilon$ is expected instead of t_m (where $\epsilon > 0$). Since $A' < 0$ we must have $\hat{A}(t_m + \epsilon, t_m + \epsilon) = A(t_m + \epsilon) < 0$. Now in the interior all functions are continuous so $\hat{A}$ is continuous. Thus δ in the region $t_m + \epsilon$ we have that $\hat{A}(x, t_m + \epsilon) < 0$ for $x \in (t_m + \epsilon - \delta, t_m + \epsilon]$. But then all consumers with indices in that range wish to leave network A and go to network B. Repeating this process we converge back to the equilibrium t_m. The analogous argument for negative ϵ shows the equilibrium is stable to perturbation downwards in expectations. Thus the equilibrium is stable.

The exact same form of argument applied to an equilibrium $t_m \in E_{-0}$ shows that it too is stable. QED.

A.2. **Proof of Lemma 3. Proof of existence:** Fix an equilibrium $t_e^0 \in E_0(p_A^0, ...)$ then we can define $t_e(p_A, ...)$ by picking $t_e \in E(p_A, ...)$ consistent with t_e^0. Since $A(t)$ is continuously differentiable so too will be $t_e(p_A, ...)$ (at least almost everywhere – see below). For notational convenience whenever a parameter is fixed we shall drop it from the list of arguments to $t, A, ...$

Differentials: implicitly differentiate the equation $A(t) = 0$ with respect to the relevant variable (p_A, f_A, f_B). Since increasing A's price by dp shifts the $A(t)$ curve down by dp reducing t_e the sign of the differential is as stated. Similarly increasing f_A shifts the network advantage curve down and therefore the advantage curve down reducing t_e and therefore the differential with respect to f_A must be negative (and conversely for f_B).

Remarks on discontinuity and profit maximization: Fix f_A, f_B, then $t_e(p_A) = A^{-1}(p_A)$ is the demand function faced by M. From the previous result we know this is downward sloping. Now take a stable equilibrium t^0 when $p_A = 0$ and assume there exists an adjacent non-extremal equilibrium $t^{0'} \leq t^0$ (which must be unstable). Then there must exist a maximum of $A(t)$ at $t^1 \in (t^{0'}, t^0)$ with $A'(t^1) = 0$ and the demand function $t_e(p_A)(t_e(0) = t^0)$ is discontinuous at t^1 with $p_A^d = A(t^1)$.

Despite this there will still exist a profit maximizing price $p_A^d > p_A^m$ since

$$\lim_{t \to t^1_+} A^{-1}(t) = -\infty$$

A.3. **Proof of Lemma 2 (Porting Lemma).**

Proposition 7. *Suppose that a network has a piece of software produced directly for it. Then s_X, p_X^s are determined by f_X^d (the direct cost of software production) alone. We may therefore take $f_X = f_X^d$ in all the formulas obtained above (it is immaterial for the purposes of calculating all equilibrium values whether software is ported or produced directly for this network).*

Proof. The cost of porting is less than the cost of direct production. Thus as long as one software firm enters directly it must be the profit condition of that firm that binds (i.e.

is zero). This condition alone determines the total number of software firms and software prices. □

Clearly if no firm produces directly there can be no porting as there would be nothing to port.

Proposition 8. *If porting is possible in both directions and both hardware platforms have some software produced directly then both platforms have the same amount of software produced for them.*

Proof. If software is produced directly then all software that could have ported must have been (since it is cheaper to port). Let d, p (d', p') be the amount of directly produced software and ported software respectively on A (B). Then $s_A = d + p$ but $p' = d, p = d'$ so $s_A = s_B$. □

If this is the case it requires $f_A^d n_B = n_A f_B^d$ since $s_X^2 f_X^d = n_X$. This is a strong condition which is unlikely to be satisfied. Thus we assume:

Assumption: $f_A^d n_B \neq n_A f_B^d$

This assumption immediately implies the converse of the previous proposition, namely that that software is produced directly for at most one network.

A.4. **Proof of Welfare-Related Propositions.** Consumer welfare as a function of network A's size (t) is given by (for simplicity ϕ is omitted):

$$W^C(x) = -x \cdot p_A + x\nu_A(x) + (1-x)\nu_B(1-x) - \int_0^x h_A(t)dt - \int_x^1 h_B(t)dt$$

Moving to total welfare we need only add in the relevant expression for $\Pi_A = x \cdot p_A - e(f^p)$. Thus:

$$W = x \cdot p_A - e(f^p) - x \cdot p_A + x\nu_A(x) + (1-x)\nu_b(1-x) - \int_0^x h_A(t)dt - \int_x^1 h_B(t)dt$$

A.4.1. *Proof of Lemma 4.* Differentiating consumer welfare with respect to x yields:

$$\frac{dW^C}{dx} = -p_A + \nu_A(x) - \nu_B(1-x) - h_A(x) + h_B(1-x) + x\nu_A'(x) - (1-x)\nu_A'(1-x)$$

This simplifies to ($A(x)$ is the utility advantage of A over B defined previously):

$$\frac{dW^C}{dx} = A(x) + x\nu_A'(x) - (1-x)\nu_B'(1-x) = A(x) + \mu(x)$$

where we have defined:

$$\mu(x) = x\nu'_A(x) - (1-x)\nu'_B(1-x)$$

At an equilibrium $t_e, A(t_e) = 0$, so this reduces to:

$$\frac{dW^C}{dx} = \mu(t_e)$$

QED.

A.4.2. *Proof of Lemma 5.*

$$\frac{dW^C}{dp_A} = -x + \frac{dx}{dp_A}\frac{dW^C}{dx}$$

Considered at an asymmetric equilibrium the second term will be greater than or less than zero depending on whether μ is less than or greater than zero. If μ is non-negative then the second term is negative and total sum will be negative. If μ is negative the total sum will be ambiguous (depending on the relative magnitudes of the two terms). Thus, if network effects do not show very strong diminishing returns (and so μ is non-negative) welfare changes negatively with increasing price. If μ is negative (as it would in the circular city model) then the effect on consumer welfare depends on the relative size of the monopoly pricing costs (first term) versus the network externality (second term).

Turning to total welfare we have:

$$\frac{dW}{dp_A} = \frac{d\Pi_A}{dp_A} + \frac{dW^C}{dp_A} = \frac{dx}{dp_A}(p_A + \frac{dW^C}{dx})$$

The term outside the brackets is negative but again here the second term can have either positive or negative sign in general. NB: when the monopolist is profit maximizing the differential of monopolist profits with respect to price is zero. Thus, the differential of total welfare equals the differential of consumer welfare.

A.4.3. *Proof of Lemma 6.* The change in consumer welfare as a consequence of an increase in the cost of porting is:

$$\frac{dW^C}{df^p} = (1-x)\frac{d\nu_B}{df^p} + \frac{dx}{df^p}\frac{dW^C}{dx}$$

The first term is clearly negative since software provision on network B declines as porting costs go up. The analysis of the second term is similar to the case of a change in price. As network A's market share increases as porting costs increase the second term will be greater than or less than zero depending on whether μ is greater than or less than zero. Thus, if μ is less than zero (strongly diminishing marginal network effects) the total will be unambiguously negative and consumer welfare declines with increases in porting costs. If μ is positive then the total has ambiguous sign in general, and will depend on relative sizes of the two terms.

For total welfare we have:

$$\frac{dW}{df^p} = \frac{d\Pi_A}{df^p} + \frac{dW^C}{df^p}$$

When profit-maximizing the first term is zero and the differential of total welfare equals that of consumer welfare. When not at a profit-maximizing level of porting costs the first term is positive. In this case whether the total is positive or negative will depend on the specific circumstances.

Appendix B. Software Production

There are two main methods of modelling product variety in the literature. One based on monopolistic competition and one based on locational models. The monopolistic competition approach has already been extensively used to demonstrate indirect network effects in hardware/software systems (see e.g. Church and Gandal (1992); Church, Gandal, and Krause (2003)). One can also use an approach based on locational differentiation and that is the approach we adopt here.

Software firms on platform X have fixed costs f_X and marginal costs c^s_X. Marginal costs are assumed to be constant across the two networks but fixed costs are not. Because software production involves a fixed cost it cannot be provided competitively. Instead we introduce a locational model of product differentiation and imperfect competition

For each network model software 'space' is represented as a circle (of circumference 1). Software firms are assumed to locate symmetrically (and therefore equidistantly) in this space.[20] while consumers are distributed uniformly over it (so total demand for software on

[20]Firms' location decisions could be endogenized and this outcome derived as an equilibrium configuration – see Economides (1989) However we choose to take this as an assumption for the sake of simplicity.

network X is the total number of consumers on that network: n_X). Following the standard circular city model[21] we have consumer's (expected) utility from software consumption is:

$$u^s_X(s_X, p^s_X) = -E[d(x(s_X))] - p^s_X$$

Where d is a 'travel' cost function of all locational models, $x(s_X)$ is the distance a consumer is from the nearest software, and E is the expectation operator. Average travel cost is used because it is assumed that consumers make their decision when they do not yet know their exact position in software space relative to software producers. Thus they base their decisions on expected costs (which will be common across consumers). We shall assume a linear travel cost, $d(x) = kx$.

B.1. **Solving.** The main result can be stated in the form of a lemma:

Lemma 9. *Given expected network sizes n^e_X the equilibrium level of software production, associated prices, and software utility are:*

$$s_X = \sqrt{\frac{kn^e_X}{f_X}}$$

$$p_X = c^s_X + \sqrt{\frac{kf_X}{n^e_X}}$$

$$u^s_X(s_X, p^s_X) = -c^s_X - \frac{5}{4}\sqrt{\frac{kf_X}{n^e_X}}$$

Proof. The setup is exactly the same as the textbook circular city model (see e.g. Tirole 1988) except that demand rather than being 1 is equal to the expected market size of that network: n^e_X. This leaves prices unchanged (since the shape of demand curve is unchanged), so in equilibrium: $p_X = c^s_X + \frac{k}{s_X}$ where k is the cost of travel ($d(x) = kx$). Firms locate equidistantly and each face the same level of demand equal to total demand divided by the number of software firms. To determine the number of software firms we use the free entry condition which means that in equilibrium firms earn zero net profits – i.e. they cover fixed costs:

[21]See e.g. Tirole (1988) for details.

$$(p_X - c_X^s)\frac{n_X^e}{s_X} - f = 0 \Rightarrow \frac{kn_X^e}{s_X^2} - f = 0 \Rightarrow s_X = \sqrt{\frac{kn_X^e}{f}}$$

This in turn gives:

$$p_X = c_X^s + \sqrt{\frac{kf}{n_X^e}}$$

The form of the software utility functions in our particular case? Consumers do not know the exact location of firms in advance so they base their decisions on the expected distance from a software producer. Software firms locate randomly but equidistantly on the circle and consumers are uniformly distributed thus expected distance between a consumer and the nearest software is a quarter of the distance between firms. Distance between firms is the inverse of the number of firms, s_X. We therefore have:

$$u_X^s(s_X, p_X^s) = -p_X^s - k(\frac{1}{4s_X})$$

Substituting the values for p_X, s_X we have[22]:

$$u_X^s(s_X, p_X^s) = -c_X^s - \frac{5}{4}\sqrt{\frac{kf}{n_X^e}}$$

□

Remark: Since the constant $\frac{5\sqrt{k}}{4}$ can be absorbed into fixed cost f_X this variable will be omitted in future and we have:

$$u_X^s(s_X, p_X^s) = -c_X^s - \sqrt{\frac{f}{n_X^e}}$$

We can now substitute this expression for u_X^s to obtain:

Corollary 10. *The reduced form of the utility function is:*

$$u_X(t) = \phi - p - h_X(t) - c_X^s - \sqrt{\frac{f_X}{n_X^e}}$$

Remark: Note how this shows that the model displays indirect network effects as the reduced form expression for utility displays positive feedback between the *total* number of

[22]The result for the quadratic distance case would be:
$u_X^s(s_X, p_X^s) = -c_X^s - \sqrt{\frac{kf}{n_X^e}} - \frac{f}{16n_X^e}$

consumers on X and the utility of an *individual* on X: $u^{s'}_X > 0$ (differentiating with respect to n^e_X).

References

T. Bresnahan. The economics of the microsoft case, 1989. Unpublished discussion paper.

Jay-Pil Choi. The provision of (two-way) converters in the transition process to a new incompatible technology. *Journal of Industrial Economics*, 45(2):139–153, 1997.

Jeffrey Church and Neil Gandal. Network effects, software provision and standardization. *Journal of Industrial Economics*, 40(1):85–103, 1992.

Jeffrey Church, Neil Gandal, and David Krause. Indirect network effects and adoption externalities, 2003. mimeo.

Stephen Davis and Kevin Murphy. A competitive perspective on internet explorer. *AER Papers and Proceedings*, 90(2):184–187, 2000.

Nicholas Economides. Symmetric equilibrium existence and optimality in differentiated products markets. *Journal of Economic Theory*, 47(1):178–194, 1989.

Joseph Farrell and Michael Katz. Innovation, rent extraction, and integration in systems markets. *Journal of Industrial Economics*, 48(4):413–432, 2000.

Joseph Farrell and G. Saloner. Converters, compatibility, and the control of interfaces. *Journal of Industrial Economics*, 40(1):9–35, 1992.

Franklin Fisher. The ibm and microsoft cases: What's the difference. *AER Papers and Proceedings*, 90(2):180–183, 5 2000.

D. Fudenberg and J. Tirole. Pricing a network good to deter entry. *Journal of Industrial Economics*, 48(4):373–90, 2000.

R. Gilbert and Michael Katz. An economist's guide to u.s. v. microsoft. *JEP*, 15(2):25–44, 2001.

C. Hall and R. Hall. Toward a quantification of the effects of microsoft's conduct. *AER Papers and Proceedings*, 90(2):188–191, 5 2000.

R. Jackson. Findings of fact in the case of united states vs. microsoft, 11 1999.

Benjamin Klein. The microsoft case: What can a dominant firm do to defend its market position? *JEP*, 15(2):45–62, 2001.

S. Liebowitz and S. Margolis. *Winner, Losers and Microsoft: Competition And Antitrust in High Technology*. Independent Institute, 1999.

Andrew Odlyzko and Benjamin Tilly. A refutation of metcalfes law and a better estimate for the value of networks and network interconnections, 2005. URL `http://www.dtc.umn.edu/ odlyzko/doc/metcalfe.pdf`.

Jean-Charles Rochet and Jean Tirole. Two-sided markets : A progress report. IDEI Working Papers 275, Institut d'conomie Industrielle (IDEI), Toulouse, November 2005. Forthcoming in the RAND Journal of Economics.

Jean-Charles Rochet and Jean Tirole. Platform competition in two-sided markets. *Journal of the European Economic Association*, 1(4):990–1029, 06 2003.

M. Whinston. Tying, foreclosure, and exclusion. *American Economic Review*, 80(4): 837–859, 1990.

M. Whinston. Exclusivity and tying in u.s. v. microsoft: What we know, and don't know. *JEP*, 15(2):63–80, 2001.