The Control of Porting

Rufus Pollock

Cambridge University

2006-09-25

Abstract

Keywords: Network Effects, Converter, Porting, Microsoft, Welfare

The recent antitrust case of *US vs. Microsoft* has focused attention on the welfare consequences of a dominant firm's activities – be they related to pricing or to maintaining barriers to entry. In this paper we introduce a model tailored to two-layered 'hardware/software' network markets in which there is 'porting' – the conversion of 'software' developed for one network to run on another. Using this model we show that the social and consumer welfare losses arising from efforts by a dominant firm to maintain barriers to entry – interpreted as obstructing porting – can be significantly larger that the losses associated with higher prices.

Introduction

It is well known that social welfare losses associated with monopoly may arise from activities to maintain or erect barriers to entry as well as from higher prices. However little work has been done to compare the welfare losses from such activities to the traditional deadweight losses from higher prices. In this paper we address this question in the context of a hardware/software[1] model of indirect network effects where the barrier is the cost of 'porting' – porting being the conversion of 'software' running on one 'hardware' to run on the hardware of another network.

We demonstrate that a dominant player's activities to maintain or erect barriers to entry may result in significantly larger costs to total and consumer welfare than its pricing activities. In particular, using a specific example, consumer welfare losses when a monopolist can manipulate price *and* porting cost are **nine** times higher than when it can manipulate price alone (for welfare as a whole the change is from a very slight positive increase in welfare when the monopolist can set price to a significant decrease in welfare when the monopolist can manipulate price and porting costs). Furthermore, the ability to manipulate porting costs proves very lucrative for the monopolist: profits increase by 400% when porting costs can be manipulated – this even though expenditure to prevent porting is equal to a fifth of gross profits.

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 network markets and their social welfare consequences. Such insights are particularly germane given the growing importance of markets exhibiting network effects – the recent antitrust case, *US v. Microsoft* demonstrates the considerable uncertainty over the motivations for and consequences of dominant firm's behaviour[2].

At another level the paper is important for introducing a new, and micro-founded, model of indirect network effects. Existing approaches such as that of Church and Gandal (1992) and Church, Gandal Krause (2003) have been based on monopolistic competition. In this paper we introduce a new approach based on a locational model of imperfect competition. This has the added benefit over a monopolistic competition approach that the functional form for indirect network effects is analytically derivable.

The paper also builds on existing work on 'converters' in network markets (converters are devices that allow a user on one network to gain access to a separate network). 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[3]. We introduce 'porting' which denotes the conversion of software from one network to another (i.e. getting software that 'runs on' the hardware of one network to run on the hardware of another network). Thus Porting is an analogous activity in a hardware/software model of indirect network effects to a converter in a model of direct network effects.

Given the centrality to our model of porting and the manipulation of its cost it is worth providing a few concrete examples taken from the Microsoft case of what this means in reality. For while the cost of porting is something that is readily acknowledged[4] the effort expended to maintain or increase porting costs is not often as well appreciated.

A first example is provided by Microsoft's behaviour in the area of web browsers. Microsoft not only developed their own independent browser (Internet Explorer) rather than adopt an existing product (such as Netscape) but also engaged in substantial engineering (and sales) efforts to make Explorer the default browser on Windows (for example by hard-wiring it into the core operating system).

While there may have been some technological reasons for this choice in the main it appears to have been motivated by a desire to control a key interface of future application development. This policy required substantial expenditure for Microsoft. According to the Judge Jackson's finding of fact (Jackson 1999) paragraph 135: From 1995 onward, Microsoft spent more than $100 million each year developing Internet Explorer. The firm's management gradually increased the number of developers working on Internet Explorer from five or six in early 1995 to more than one thousand in 1999. It also had substantial costs for other players in the form of inhibiting web-based innovation and increasing the difficulties of developing websites (Internet Explorer continues to remain one of the least standards-compatible browsers available).

A second example is provided by Microsoft's behaviour with respect to core data formats. In general, and unless pressured by external events, Microsoft has consistently refused to publish any details. The 'IP' locked up in those in terms of useful technological information about the programs is likely to be minimal[5], but by keeping these formats closed Microsoft make it very hard to port productivity applications to other platforms[6]. Specific examples include the file formats of the applications in Microsoft Office and the specification for the NTFS file system. This is an example where the expenditure by Microsoft is more indirect. It consists of the extra effort required to write and support their own system and, more importantly, the reduction in quality and variety of software applications available for its own platform.

A final example is the expenditure of time and effort to undermine the use of

Java as a cross platform programming language. In the late 1990s Microsoft wrote a non-conformant Java (VM) (virtual machine) including a set of Windows specific APIs (Application Programming Interfaces) of dubious technical value but which would have severely undermined the cross-platform nature of Java and hence reduced the ease of porting of Java programs. Since 1999 Microsoft has also engaged in enormous expenditure in developing the .NET framework to compete directly with Java[7]. .NET is extremely similar to Java in concept and implementation but is not cross-platform (so programs written using the .NET framework cannot be ported without great difficulty to other systems such as Linux). Microsoft have gone to great lengths to encourage programmers to write against the .NET APIs (rather than Java) and currently intend to use .NET as the core OS API in Longhorn, the next version of Windows. While any estimate for the cost of such efforts must be speculative it is probably not an underestimate to suggest that well over $1 billion have been spent on the project. Given that .NET provides few advantages over Java but instead replicates most of the core features - except for the crucial one of cross-platform compatibility - this expenditure is a prime example of an investment whose sole aim is to raise or maintain the cost of porting.

The Model

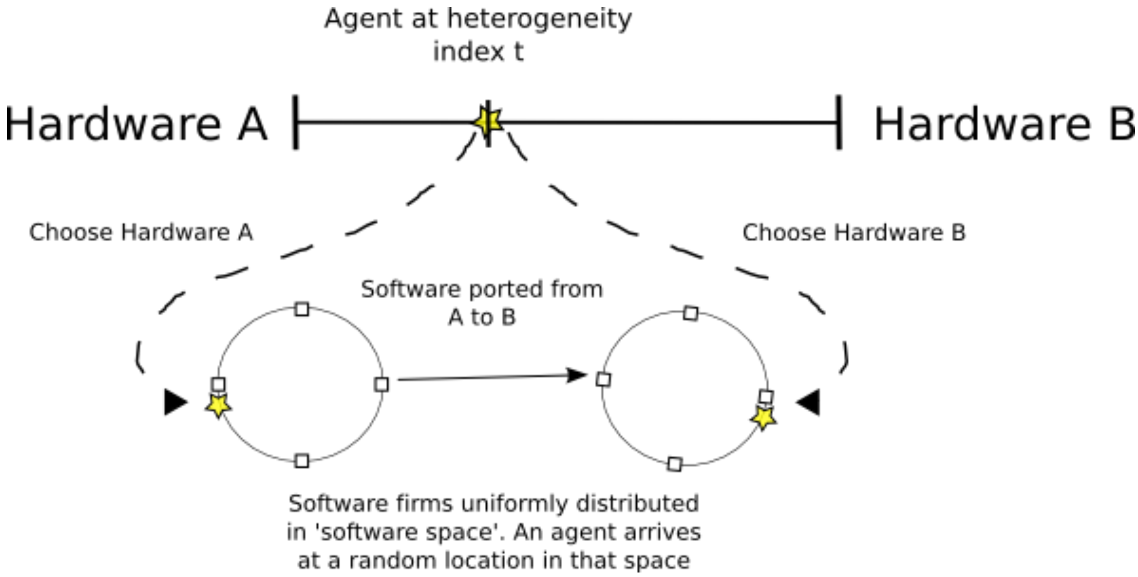

Figure 1: A visual summary of the model. There is heterogeneity between agents in two respects: first via their location relative to 'hardware' (linear city type) but also in relation to 'software' (circular city type). (NB: that though porting has been shown only from A to B porting from B to A is not proscribed).

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

Two types of product are provided for each network: hardware (H) and software (S). Agents must purchase one unit each of a network's hardware and the associated software to obtain utility from that network. The terms hardware and software should not be construed literally but rather to indicate the complementary nature of two types of good. If an agent has already purchased hardware and software from one network she gains no extra utility from purchasing from a second network (so an agent will purchase from at most one network).

The measure of agents 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:

$$u_X\left(t,\ p_X,\ s_X,\ p_X^s\right) = \varphi - p_X - h_X(t) + u_X^s\left(s_X,\ p_X^s\right)$$

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 -φ)
- p_X is the price of hardware on network X
- $h_X(t)$ models agent 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 agents by heterogeneity, i.e. $h^{'}(t) > 0$
- u_X^s is utility from software purchase with s_X the amount of software available on network X and p_X^s the price of a piece of software (constant across software firms). 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 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 $c = 0$.

Software Production

Software firms on platform X have fixed costs f_X and marginal costs c_X^s. 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[8].

For each network model software 'space' as a circle (of circumference 1). Software firms are assumed to locate symmetrically (and therefore equidistantly) in this space[9] while agents are distributed uniformly over it (so total demand for software on network X is the total number of agents on that network: n_X). Following the standard circular city model[10] we have agent's (expected) utility from software consumption is:

$$u_X^s\left(s_X,\ p_X^s\right) = -E[d(x(s_X))] - p_X^s$$

where d is a 'travel' cost function of all locational models, $x(s_X)$ is the distance an agent is from the nearest software, and E is the expectation operator. Average travel cost is used because it is assumed that agents 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 agents). We shall assume a linear travel cost, $d(x) = kx$.

Porting

Software may be created by two methods. Either it can be created directly for network X at fixed cost f_X^d 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 competitors though at the cost of some expenditure on its own part.

Take f^p as the fixed cost of such porting and define $e = e(f^p)$ be the expenditure to prevent porting. 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_X^d if all software is produced directly (none is ported); a mixture of f_X^d and f^p if some software is ported and some produced directly; or f^p if all software is ported.

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)
3. Taking the resulting level of software provision as given agents 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, $\Pi = p_A \cdot n_A(p_A,\ f^p)$, are determined.

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

Solving the Model

We shall solve the model in the following way: first, solve software producers problem (on the basis of a common expectation by software producers of the network sizes that will occur). This yields the number of software firms and the software price in terms of the expected network size. Substitute these values into u_X^s to obtain a reduced form of agents' utility function on network X. This utility function will now display 'network effects', that is positive feedback between an individual's utility and the number of other agents on the same network. We now have a standard network effects model which may me solve in the usual manner to obtain network sizes as a function of the monopolist's choice variables: $n_A = n_A(p_A,\ f^p)$. The monopolist then solves:

$$\max_{p_A,\, f^p} p_A n_A(p_A,\ f^p) - e(f^p).$$

Software Production

Lemma 1: Given expected network sizes n_X^e the equilibrium level of software production, associated prices, and software utility are:

$$s_X = \sqrt{\frac{k n_X^e}{f_X}}$$

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

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

Proof: see appendix

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: The reduced form of the utility function is:

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

Remark: Note how this shows that the the model displays indirect network effects as the reduced form expression for utility displays positive feedback between the *total* number of agents on X and the utility of an *individual* on X: $u_X^{s'} > 0$ (differentiating with respect to n_X^e).

Solving for Network Equilibrium

To solve for equilibrium network size we proceed by the usual method based on finding the marginal agent indifferent between the two platforms. This is standard in the literature so the details will be omitted and results stated directly.

First observe that agents gain no extra utility from purchasing from more than one network. Thus we may assume that agent's purchase at most one set of compatible hardware and software. We further assume that all agents do purchase from one or other network[11]. 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 agent 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 agent:

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

Lemma 2: The set of equilibria of the model as presented above are given by $E = E_0 \cup E_{-0}$ where $E_0 = \{t : A(t) = 0\}$ and $E_{-0} = \{0 : A(0) < 0\} \cup \{1 : A(1) > 0\}$. An equilibrium $t_e \in E_0$ is stable if $A^{'}(t) < 0$. All $t_e \in E_{-0}$ are stable.

Proof: see appendix.

Using the expression for the utility function from the corollary above we have that:

$$A(t) = -p_A - h_A(t) + h_B(t) - \sqrt{\frac{f_A}{t}} + \sqrt{\frac{f_B}{1 - t}}$$

While this characterizes the equilibria implicitly we should like to have an explicit expression for equilibrium network size $t_e = n_A$. Unfortunately given the functional form of $A(t)$ there is no way to analytically solve the equation $A(t) = 0$ to obtain t_e. Of course we may still solve this equation numerically – as we shall do below – as well to characterise the behaviour of the solution(s) in various ways. First however let us examine one specific case graphically in order to aid our intuition.

An Example

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 ($h_B(1 - t) = h_A(t)$). Let us set heterogeneity to be $h_A(t) = 10t^{10}$. This corresponds to a situation where there is a large middle ground of agents who are fairly indifferent between the two platforms (h(t) is small) 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:

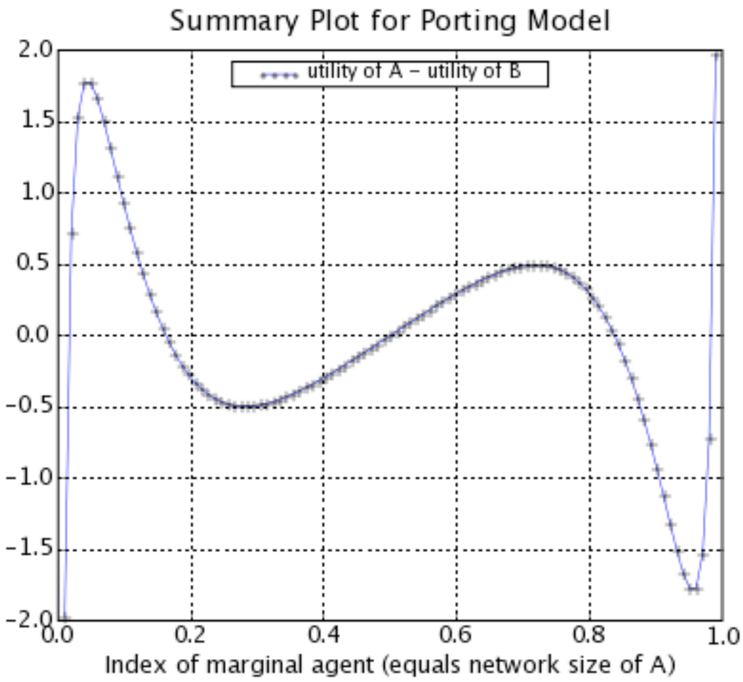

Figure 2: A Plot showing advantage function, $A(t)$ in the symmetric case when the access prices for the two networks are the same. There are stable equilibria at 0 and 1 (the 'standardization' equilibria) and 0.16 and 0.84 (asymmetric stable equilibria). There are unstable equilibria at 0.5 and 0.02 and 0.98.

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[12].

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). For example we can plot the demand function derived from Figure 1 in the neighbourhood of the stable equilibrium at 0.84:

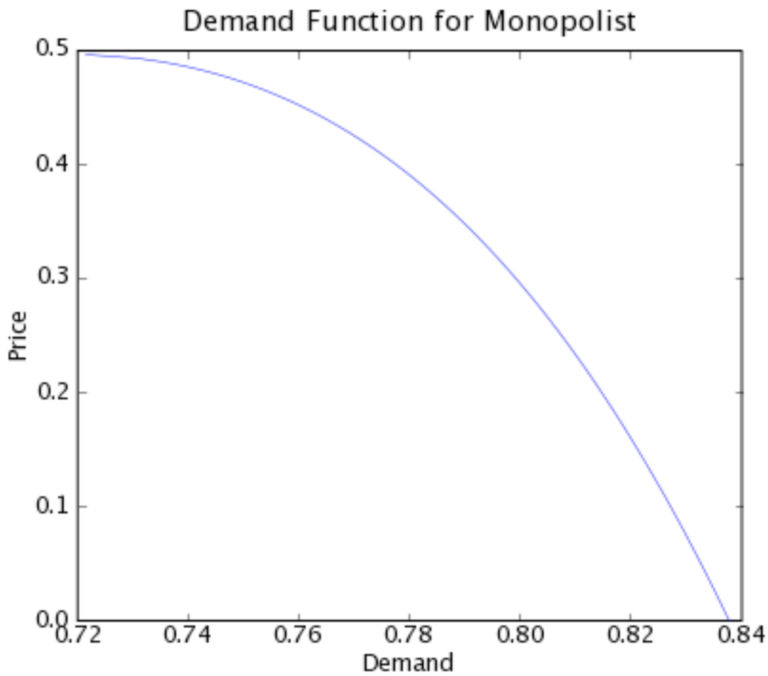

Figure 3: Demand function for monopolist in neighbourhood of stable equilibrium at 0.84. Demand is discontinuous at a price just below 0.5 (i.e. at the left edge of the diagram – the discontinuity itself is not shown as it distorts the scale). At the discontinuity demand will suddenly jump down to approximately 0.14 which is the next place the line y=0.5 would intersect $A(t)$ (see Figure 2). Note that this diagram is just the relevant portion of Figure 2 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 2. 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. Furthermore suppose that the monopoly price is 'close' to the price at which the discontinuity occurs (as is the case above) and that the monopolist is subject to some degree of uncertainty. Then the monopolist would want to keep a safe distance from the discontinuity price to avoid sudden tipping (the market above is subject to hysteresis – once you have jumped to another equilibria it is hard to get back). If the profit-maximizing price is close to the discontinuity price the result would be that the monopolist would probably not charge the profit maximizing price but rather something lower in order to have a margin for error.

Other Comparative Statics: Just as we can evaluate the effect of changing prices in the above diagram by considering how it shifts $A(t)$. In particular considering fixed costs we can see that 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 unlike price, fixed costs do not enter linearly so they will also change the 'shape' of $A(t)$.

Properties of Equilibrium

We can distill the insights gained in relation to the special case above into a general result:

Lemma 3: Noting that the advantage function 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 $E = E(p_A,\ ...)$) after picking a stable equilibrium $t_e^0 \in E_0\left(p_A^0,\ ...\right)$ we have associated 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 an 'demand function':

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

Differentiating we have:

1. Downward sloping demand curve: $\dfrac{dq}{dp_A} = \dfrac{-1}{A'(t_e(p_A))t_e'(p_A)} < 0$
2. $\dfrac{dt_e}{df_A} < 0$
3. $\dfrac{dt_e}{df_B} > 0$

Finally though 'demand' is discontinuous at some point locally there exists a unique profit maximizing price.

Proof: see appendix.

Porting

Let us determine the level of differents kinds of software production for each network (produced directly, ported or produced by a mixture of those methods). In doing so we will have determined the fixed cost 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 4: 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.

The Monopolist's Profit Maximization Problem

We first 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 be the case by definition!

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 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[13].

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 doesn't bind). As already discussed the monopolist may control the cost of porting from its network so the profit maximization problem becomes:

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

Since t_e does not have an analytical form the monopolist's maximization problem will have to be solved numerically.

Welfare

What are the welfare effects in this model of changes in prices and fixed costs? Total welfare, $W = \Pi_A + W^C$ where W^C is consumer welfare and Π_A are the monopolist's profits.

Lemma 5: When at an interior equilibrium, t_e, the marginal change in welfare of a change with respect to network size is:

$$\frac{dW^C}{dt} = -0.5 \cdot (\text{A Network Effects} - \text{B Network Effects}) = 0.5 \cdot \left(\sqrt{\frac{f_A}{t_e}} - \sqrt{\frac{f_B}{1 - t_e}} \right)$$

Proof: see appendix.

Remark: this is negative when there is as much or more software on network A than on network B – exactly the situation we are focusing on (recall that the f_A/f_B terms in the square root are the amount of software on A/B times a constant). Thus in this model, *at asymmetric equilibria* consumer welfare is decreasing as network A gets larger.

Lemma 6: The effect on consumer and total welfare of an increase in the price charged by the monopolist is ambiguous both in general and when the monopolist is profit maximizing.

Proof: see appendix.

Remarks: Monopoly pricing does not result in traditional deadweight losses since total demand is fixed and does not change (agents who leave one network join the other). 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.

These consequences may be either positive or negative depending on whether welfare changes positively or negatively with network's A share. In our situation the network effects function derived from software consumption means that for asymmetric equilibria welfare decreases as the share of network A increases. Thus higher prices, by decreasing network A's share, may in fact increase consumer welfare even though consumers on A are having to pay more. The same is true with total welfare though here of course the monopolist's profits are added back into the equation. In both cases it is uncertain whether an increase in M's price will have a positive or negative effect.

Lemma 7: The effect on consumer welfare of an increase in the cost of porting is negative at all times. For total welfare when the monopolist is profit maximizing the effect is the same as for consumer welfare (i.e. negative).

Proof: see appendix.

Remarks: Higher porting costs result in a reduction in availability of software for those on platform B as well as higher software prices. These losses are further compounded by the network effects as initially marginal agents move from platform B to platform A (further reducing software availability on network B). Both of these changes have negative welfare consequences. The first because higher porting costs mean less software for B users (holding network B's share constant). The second because reducing network B's market share reduces welfare.

Finally higher porting costs that result from M's efforts result in costs to A that from a welfare point of view are deadweight losses (of course for A such activities may be highly profitable).

An Example

We now turn to a specific example to illustrate the previous analysis. We also demonstrate one of the key quantitative claims of the paper, namely that the welfare costs (consumer or societal) of efforts to maintain barriers to entry (increase porting costs) can 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 inital porting cost is set to two-thirds of that value, so $f^p = 1.0$. The monopolist's expenditure function is: $e\left(f^p\right) = 2 \cdot \left(f^p - 1\right)^4$ 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 functional form of the expenditure function is 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 the following table (NB: since φ is an arbitrary constant it has been set so that initial welfare values are normalized to zero).

	Porting Cost	Price of A Hardware	Demand for A (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

Welfare Results at Various Prices and Porting Costs

The first line is there to show the baseline case when the values of various key variables when the control parameters are at their default values (that is without intervention by the monopolist). M's default market share (with its own price at zero and the fixed costs of porting at 1) is 75%. With zero prices, though, profits for M are also zero. Total welfare and consumer welfare are the same – since prices are zero – and has been normalized to zero. This value has no 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 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 changes 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, almost reaching the same levels as the cost (1.5) of direct production. 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. Thus 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.

Remarks

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, $e\left(f^{p}\right)=0.0616$, 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).

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.

Total welfare: For total welfare as discussed above 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.

Conclusion

In this paper we have introduced a new model of indirect network effects in hardware/software markets based on spatial (circular city) production differentiation in the software market. Solving the model we have obtained closed form solutions for the network utility function and characterised the set of equilibria.

Into this model we then introduced 'porting' which plays a role analogous to 'converters' in the simpler direct network effects models. With 'porting' software developed for one network 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 the cost of porting to a competitor network. It was shown that while the effect of pricing effect on welfare were ambiguous the effect of efforts to increase barriers to entry by raising porting costs was negative.

In order to derive a quantitative estimate we examined a specific case of our model and found that the social and consumer welfare losses arising from efforts to manipulate porting dwarfed the welfare effects stemming from monopoly pricing. In particular, consumer welfare losses from the combination of higher porting costs and higher prices were over nine times higher that those arising from higher prices alone. Meanwhile total welfare was barely changed with monopoly pricing alone but showed significant losses with the combination of higher porting costs and monopoly pricing (welfare loss went from zero to approximatley three fifths of the monopolist's profits).

This result has important consequences for antitrust policy which has traditionally focused on the welfare impact of higher prices (on consumers) as it suggests that such an approach may greatly underestimate the implications for welfare of a dominant's firm activities (in network markets).

Finally there are several obvious avenues for future research. Porting and the manner in which a dominant firm may prevent it have been modelled in a fairly simple manner. One might improve this, for example, by changing from a 'black box' cost function E to a setup where f_A increases with f^p. This would correspond to an 'obfuscation' situation where increasing porting costs to competitor platforms also increases the cost of producing software on one's own platform.

Another improvement would be to add dynamics to the model (though this would also greatly increase complexity). For example rather than having a fixed static demand one could allow agents to arrive over time[14]. Alternatively agents 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 also raise interesting questions about 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)[15].

Appendix

Proof of Lemma 1

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_X^e. This leaves prices unchanged (since the shape of demand curve is unchanged), so in equilibrium:

$$p_X = c_X^s + \frac{k}{s_X}$$

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:

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

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?
Agents 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 agents are uniformly distributed thus expected distance between an agent 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:

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

Substituting the values for p_X, s_X we have[16]:

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

Proof of Lemma 2

Recall that the *conditional utility advantage* of network A over network B for agent 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 agent:

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

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

Since 'heterogeneity cost' for an agent is increasing in the distance of the agent 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 $\hat{A}(t) < 0 \forall\ t \geq t_m$.

Now an agent (with expectations of network A size equal to n_A) chooses network A over B iff $\hat{A}(t) \geq 0$. Thus if an agent with index t_m chooses network A then all agents with index $t \in [0, t_m]$ choose network A. Similarly if an agent with index t_m chooses network B then all agents 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 agent 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 agents choose network A while for $t \in (t_m, 1], \hat{A}(t) < 0$ so these agents choose network B.

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

Furtheremore 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 $f : [0, 1] \rightarrow [0, 1]$ 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 agents joining a single network.

Remark: This result sets up an implicit equivalence between network size and the marginal agent (where the term marginal is broadened to include the tipping situations where $t_m = 0 \text{or} 1$ 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 $x \in (t_m + \epsilon - \delta,\ tm_m + \epsilon]$. But then all agents 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 $t_m \in E_{\text{-}0}$ shows that it too is stable. QED.

Proof of Lemma 3

Proof of existence: Fix an equilibrium $t_e^0 \in E_0\left(p_A^0,\ ...\right)$ 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 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 $t^1 \in \left(t^{0'},\ t^0\right)$ with $A'\left(t^1\right) = 0$ and the demand function $t_e(p_A)\left(t_e(0) = t^0\right)$ is discontinuous at t^1 with $p_A^d = A\left(t^1\right)$.

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

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

Proof of Lemma 4 (Porting Lemma)

Prop: 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. QED.

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

Prop: 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 produced directly then all software that could have ported must have been (since 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 $s_A = s_B$ QED.

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.

Proof of Welfare-Related Propositions

Consumer welfare as a function of network A's size (t) is given by:

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

Where for simplicity φ is omitted and we define $\nu_X(t)$ for network effects on network X, i.e. $-\sqrt{\frac{f_X}{t}}$

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

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

Proof of Lemma 5: Differentiating consumer welfare with respect to t yields:

$$\frac{dW^C}{dt} = -p_A + \nu_A(t) - \nu_B(1-t) - h_A(t) + h_B(1-t) + t\nu_A^{'}(t) - (1-t)\nu_A^{'}(1-t)$$

This simplifies to:

$$\frac{dW^C}{dt} = A(t) + t\nu_A^{'}(t) - (1-t)\nu_B^{'}(1-t)$$

In our case $t\nu_X^{'}(t) = -0.5 \cdot \nu_X(t)$. Thus
$t\nu_A^{'}(t) - (1-t)\nu_B^{'}(1-t) = -0.5 \cdot (\nu_A(t) - \nu_B(1-t))$.

$$\frac{dW^C}{dt} = A(t) - 0.5 \cdot (\nu_A(t) - \nu_B(1-t))$$

At an equilibrium t_e, $A(t_e) = 0$. QED.

Proof of Lemma 6

$$\frac{dW^C}{dp_A} = -t + \frac{dt}{dp_A}\frac{dW^C}{dt}$$

Considered at an asymmetric equilibrium the second term will be positive since both derivatives will be negative. Thus whether welfare changes positively or negatively with increasing price depends on the relative size of the monopoly pricing costs (first term) versus the network externality.

Turning to total welfare we have:

$$\frac{dW}{dp_A} = \frac{d\Pi_A}{dp_A} + \frac{dW^C}{dp_A} = \frac{dt}{dp_A}\left(p_A + \frac{dW^C}{dt}\right) = \frac{dt}{dp_A}(0.5 \cdot (v_A - v_B) - (h_A - h_B))$$

The first term is negative but again here the second term can have either postive or negative sign in general. NB: when the monopolist is profit maximizing the differential of monopolist profits with respect to price and differential of total welfare equals the differential of consumer welfare.

Proof of Lemma 7: The change in consumer welfare as a consequence ofan increase in the cost of porting is:

$$\frac{dW^C}{df^p} = (1 - t)\,\frac{d\nu_B}{df^p} + \frac{dt}{df^p}\frac{dW^C}{dt}$$

The first term is clearly negative since software provision on network B declines as porting costs go up. The second term is also negative since network A's market share increases as porting costs increase.

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 which is negative. When not profit-maximizing and porting cost is below the profit-maximizing level the first term is positive. In this case whether the total is positive or negative will depend on the specific circumstances.

Endnotes

1. This nomenclature is conceptual referring to the existence of two types of good which are complements – the products need not actually be hardware or software. For example in credit card networks hardware represents the credit card type and software the number of shops supporting that credit card. ↩
2. See, for example, Hall and Hall (2000), Fisher (2000), Bresnahan (2001), Liebowitz and Margolis (1999), Klein (2001) and Gilbert and Katz (2001)↩
3. See also Choi (1997) for another converter model albeit a dynamic one related to the transition from an old to a new technology.↩
4. Often demonstrated by the revealed preference of development firms who rarely port, or only after a significant delay, to other platforms. See also the introduction in Farrell and Salonder (1992) and the comments in Church and Gandal (1992).↩
5. The available information only indicates that these formats are fairly opaque (those interested can find leaked copies of various out of date formats such as Word 97 at http://www.wotsit.org/>). Whether this is because of poor design, obfuscation decisions, or because of the requirements of back compatibility is difficult to tell.↩
6. This is also a clear example of the creation of switching costs as a way of locking in existing customers.↩
7. such as a) a new object oriented language C# that reproduced Java in every major respect (almost as far as syntax). b) outputting to an intermediate language (called IL analogous to Java bytecode) that could then be run on any platform (though Microsoft's commitment to this feature of .NET has rapidly waned and a cross platform version of .NET now depends on the efforts of the open source Mono project) c) provision of comprehensive auxiliary class libraries ↩
8. 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. However it is analytically rather intractable, particularly for our purposes. Thus we introduce a model based on locational foundations, that has, to our knowledge, not been previously used. ↩
9. 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. ↩
10. See e.g. Tirole (1988) for details. ↩
11. Such an assumption while imposing some restrictions does not effect

the results but does simplify working greatly. What are these restrictions?

Since agents always have an outside option that yields 0 utility the requirement that all agents choose to purchase from at least one network requires for the marginal agent t_e, $u_X(t_e, n_X) \geq 0$. This requires φ to be large enough that utility on both networks is positive for the *marginal agent*. Clearly for finite φ this is not always possible since $\lim_{n_X \to 0} u_X^s(n_X) = -\infty$. But this situation, where one network's size is so low that utility drops below 0, corresponds to the point at which the whole market tips onto one network (though in this case some agents may leave rather than purchase that network, thus while say $n_B = 0$ we may have $1 > n_A$) ↩

12. To have an interior stable equilibrium $A(t)$ must interect 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 (t') is five and we must always have a situation similar to that shown. ↩
13. In fact if networks displayed symmetry, i.e. direct production costs are equal and heterogeneity funtions on the two networks are the same, this is a result rather than an assumption. ↩
14. This might result in limit-pricing behaviour by the monopolist similar to that in Fudenberg and Tirole (2000).↩
15. The alert reader may also have considered connections with the paper by Farrell and Katz (2000) on network monopolies and downstream innovation.↩
16. The result for the quadratic distance case would be:

$$u_X^s(s_X, p_X^s) = -c_X^s - \sqrt{\frac{kf}{n_X^e}} - \frac{f}{16n_X^e}$$ ↩

Bibliography and References

1. **Aoki, R. and Small, J.** *The Economics of Number Portability: SCs and Two Part Tariffs*, 2001. http://www.crnec.auckland.ac.nz/crnec/research/wp.html
2. **Arthur, B.** *Competing Technologies, Increasing Returns and Lock-in by Historical Events*, Economic Journal pp. 106-131, 1989.
3. **Baker, J.** *The Case for Antitrust Enforcement*, Journal of Economic Perspectives pp. 3-26, 2003.
4. **Bresnahan, T.** *The Economics of The Microsoft Case*, 2001. http://www.stanford.edu/~tbres/Microsoft/The_Economics_of_The_Microsoft_Case
5. **Chakravorti, S.** *Theory of Credit Card Networks: A Survey of the Literature*, Review of Network Economics, pp. 50-68, 2003-06.
6. **Choi, J.P.** *The Provision of (Two-Way) Converters in the Transition Process to a New Incompatible Technology*, Journal of Industrial Economics, 1997.
7. **Church, J. and Gandal, N.** *Network Effects, Software Provision and Standardization*, Journal of Industrial Economics pp. 85-103, 1992.
8. **Church, J.; Gandal, N. and Krause, D.** *Indirect Network Effects and Adoption Externalities*, 2003. http://spirit.tau.ac.il/public/gandal/Research.htm
9. **Crandall, R. and Winston, C.** *Does Antitrust Policy Improve Consumer Welfare? Assessing the Evidence*, Journal of Economic Perspectives pp. 3-26, 2003.
10. **Davis, S. and Murphy, K.** *A Competitive Perspective on Internet Explorer*, American Economic Review Papers and Proceedings pp. 184-187, 2000-05.
11. **Department of Justice Antitrust Division.** *COMPLAINT (For Violations of Sections 1 & 2 of the Sherman Act)*, 1994-07-15. http://www.usdoj.gov/atr/cases/f0000/0046.htm and http://www.usdoj.gov/atr/cases/ms_index_licensing.htm
12. **Economides, N.** *Symmetric Equilibrium Existence and Optimality in Differentiated Products Markets*, Journal of Economic Theory, vol. 47, no.1, pp. 178-194, 1989.
13. **Economides, N. and Salop, S.** *Competition and Integration Among Complements, and Network Market Structure*, Journal of Industrial Economics pp. 105-123, 1992.
14. **Farrell, J.** *Integration and Independent Innovation on a Network*, American Economic Review Papers and Proceedings pp. 420-424, 2003.
15. **Farrell, J.** *A Simple Price-Theory Model of Anticompetitive Exclusive Dealing*, 2004. http://emlab.berkeley.edu/users/farrell/ftp/exclusive2.pdf
16. **Farrell, J. and Saloner, G.** *Standardization, Compatibility and*

Innovation, Rand Journal of Economics pp. 70-83, 1985.
17. **Farrell, J. and Saloner, G.** *Installed Base And Compatibility: Innovation, Product Preannouncements, And Predation*, American Economic Review pp. 940-955, 1986.
18. **Farrell, J. and Saloner, G.** *Converters, Compatibility, and the Control of Interfaces*, Journal of Industrial Economics pp. 9-35, 1992.
19. **Farrell, Joseph and Katz, M.** *Innovation, Rent Extraction, and Integration in Systems Markets*, Journal of Industrial Economics pp. 413-432, 2000.
20. **Farrell, J. and Klemperer, P.** *Competition and Lock-In: Competition with Switching Costs and Network Effects*, 2001. http://www.nuff.ox.ac.uk/users/klemperer/lockinrefswebversion.pdf
21. **Fisher, F.** *The IBM and Microsoft Cases: What's the Difference*, American Economic Review Papers and Proceedings pp. 180-183, 2000.
22. **Fudenberg, D. and Tirole, J.** *Pricing a Network Good to Deter Entry*, Journal of Industrial Economics, pp. 373-390, 2000.
23. **Gilbert, R.** *Symposium on Compatibility: Incentives and Market Structure*, Journal of Industrial Economics pp. 1-8, 1992.
24. **Gilbert, R. and Katz, M.** *An Economist's Guide to U.S. v. Microsoft*, Journal of Economic Perspectives pp. 25-44, 2001.
25. **Hall, C. and Hall, R.** *Toward a Quantification of the Effects of Microsoft's Conduct*, American Economic Review Papers and Proceedings pp. 188-191, 2000.
26. **Jackson, R.** *Findings of Fact in the case of United States vs. Microsoft*, 1999-11-05. http://www.usdoj.gov/atr/cases/f3800/msjudgex.htm
27. **Katz, M. and Shapiro, C.** *Network Externalities, Competition, and Compatibility*, American Economic Review pp. 424-440, 1985.
28. **Klein, B.** *The Microsoft Case: What Can a Dominant Firm Do to Defend Its Market Position?*, Journal of Economic Perspectives pp. 45-62, 2001.
29. **Laffont, J. and Tirole, J.** *Competition in Telecommunications*, 2000.
30. **Liebowitz, S. and Margolis, S.** *Winner, Losers and Microsoft: Competition And Antitrust in High Technology*, [Independent Institute], 1999.
31. **Saloner, G. and Shepard, S.** *Adoption of Technologies with Network Effects: An Empirical Examination of the Adoption of Automated Teller Machines*, The RAND Journal of Economics pp. 479-501, 1995.
32. **Schmalensee, R.** *Antitrust Issues in Schumpeterian Industries*, American Economic Review Papers and Proceedings pp. 192-196, 2000.
33. **Tirole, J.** *The Theory of Industrial Organization*, [MIT Press], 1988.
34. **Whinston, M.** *Tying, Foreclosure, and Exclusion*, American Economic Review pp. 837-859, 1990.

35. **Whinston, M.** *Exclusivity and Tying in U.S. v. Microsoft: What We Know, and Don't Know*, Journal of Economic Perspectives pp. 63-80, 2001.