



## Information Systems Research

Publication details, including instructions for authors and subscription information:  
<http://pubsonline.informs.org>

### Analyzing Software as a Service with Per-Transaction Charges

Dan Ma, Abraham Seidmann

To cite this article:

Dan Ma, Abraham Seidmann (2015) Analyzing Software as a Service with Per-Transaction Charges. Information Systems Research 26(2):360-378. <http://dx.doi.org/10.1287/isre.2015.0571>

Full terms and conditions of use: <http://pubsonline.informs.org/page/terms-and-conditions>

This article may be used only for the purposes of research, teaching, and/or private study. Commercial use or systematic downloading (by robots or other automatic processes) is prohibited without explicit Publisher approval, unless otherwise noted. For more information, contact [permissions@informs.org](mailto:permissions@informs.org).

The Publisher does not warrant or guarantee the article's accuracy, completeness, merchantability, fitness for a particular purpose, or non-infringement. Descriptions of, or references to, products or publications, or inclusion of an advertisement in this article, neither constitutes nor implies a guarantee, endorsement, or support of claims made of that product, publication, or service.

Copyright © 2015, INFORMS

Please scroll down for article—it is on subsequent pages



INFORMS is the largest professional society in the world for professionals in the fields of operations research, management science, and analytics.

For more information on INFORMS, its publications, membership, or meetings visit <http://www.informs.org>

# Analyzing Software as a Service with Per-Transaction Charges

Dan Ma

School of Information Systems, Singapore Management University, Singapore 178902,  
[madan@smu.edu.sg](mailto:madan@smu.edu.sg)

Abraham Seidmann

W. E. Simon Graduate School of Business Administration, University of Rochester, Rochester, New York 14627,  
[avi.seidmann@simon.rochester.edu](mailto:avi.seidmann@simon.rochester.edu)

Software as a Service (SaaS) delivers a bundle of applications and services through the Web. Its on-demand feature allows users to enjoy full scalability and to handle possible demand fluctuations at no risk. In recent years, SaaS has become an appealing alternative to purchasing, installing, and maintaining modifiable off-the-shelf (MOTS) software packages. We present a game-theoretical model to study the competitive dynamics between the SaaS provider, who charges a variable per-transaction fee, and the traditional MOTS provider. We characterize the equilibrium conditions under which the two coexist in a competitive market and those under which each provider will fail and exit the market. Decreasing the lack-of-fit (or the cross-application data integration) costs of SaaS results in four structural regimes in the market. These are MOTS Dominance → Segmented Market → Competitive Market → SaaS Dominance. Based on our findings, we recommend distinct competitive strategies for each provider. We suggest that the SaaS provider should invest in reducing both its lack-of-fit costs and its per-transaction price so that it can offer increasing economies of scale. The MOTS provider, by contrast, should not resort to a price-cutting strategy; rather, it should enhance software functionality and features to deliver superior value. We further examine this problem from the software life-cycle perspective, with multiple stages over which users can depreciate the fixed costs of installing and customizing their MOTS solutions on site. We then present an analysis that characterizes the competitive outcomes when future technological developments could change the relative levels of the lack-of-fit costs. Specifically, we explain why the SaaS provider will always use a forward-looking pricing strategy: When lack-of-fit costs are expected to decrease (increase) in the future, the SaaS provider should reduce (increase) its current price. This is in contrast with the MOTS provider, who will use the forward-looking pricing strategy only when lack-of-fit costs are expected to increase. Surprisingly, when such costs are expected to decrease, the MOTS provider should ignore this expectation and use the same pricing strategy as in the benchmark with invariant lack-of-fit costs.

**Keywords:** software as a service; game theory model; pricing based on transactions; competitive strategies; lack-of-fit costs; economies of scale

**History:** Chris Forman, Senior Editor; Hsing Kenneth Cheng, Associate Editor. This paper was received on February 5, 2014, and was with the authors 2 months for 2 revisions.

## 1. Introduction

The rapid growth of the Internet and progress in telecommunication technologies over the last decade has given birth to Software as a Service (SaaS). With SaaS, software applications and clients' data are stored off-site in a central location operated by the provider. The provider is in charge of all information technology (IT) support services, including daily software maintenance, data backups, software upgrades, and security management. Users neither purchase nor own the software, but instead access it through a network, with payments to the provider based on use. This business setting differs from the traditional software solution in which enterprise software systems are delivered as modifiable off-the-shelf (MOTS)

products.<sup>1</sup> Under MOTS, providers sell the software application to users who customize and install the software, and who also must provide in-house IT infrastructure, hardware, and support services.

The SaaS concept originated with the Application Service Providers (ASPs) of the dot.com era in the late 1990s, which experienced several rounds of success and failure. Circumstances have since changed; bandwidth costs continue to decline, and users have

<sup>1</sup> A MOTS product is "a commercial application package whose code can be customized to meet a customer's particular requirements... [It] is designed to be easily installed and to interoperate with existing system components" (see <http://searchenterprise.linux.techtarget.com/definition/COTS-MOTS-GOTS-and-NOTS> for more information).

become tired of the high cost of system implementation, customization, and maintenance. More important, with the increasing use of Web tools such as Ajax, JavaScript, Flash, and XML, the technology is ready for SaaS (Dubey and Wagle 2007). According to International Data Corporation (IDC) research, SaaS revenue reached \$22.9 billion in 2011, a 30.9% year-to-year growth rate, and is expected to grow to \$67.3 billion by 2016 (Mahowald and Sullivan 2012). SaaS is increasing seven times faster than the traditional software market and will account for \$1 of every \$5 spent on software by 2016 (Sullivan et al. 2012). Clearly, SaaS providers are putting great competitive pressure on traditional MOTS providers.

In our study of the competition between SaaS and MOTS, we seek answers to the following questions: What unique value does SaaS offer to users? How will the two kinds of providers compete and share the market? What factors will determine a provider's competitive power? Finally, what competitive pricing strategies should each use?

There are a number of fundamental differences between SaaS and MOTS. For example, SaaS users expect to reap the benefits of scalability, reliability, system flexibility, ease of deployment, and ease of management (Dunn 2006, Singh et al. 2004). Many believe that SaaS is cheaper (InformationWeek 2012, Koehler et al. 2010, Zwim 2012) because it offers savings on hardware, infrastructure, application implementation and installation, and system maintenance and support. Moreover, its "pay as you go" pricing structure allows users to shift expenses from the capital budget to the operating budget and significantly reduces start-up costs.

Even so, SaaS has drawbacks. SaaS is based on the concept of a large amount of shared computing resources (Mell and Grance 2009). When running one instance of the software in a central location managed by the vendor, SaaS can attain economies of scale, but it also poses the multitenancy problem: Options for customizing SaaS applications are limited (Muller et al. 2009). Users with different backgrounds and various legacy systems inevitably have different requirements to fulfill their unique business needs, but multitenancy makes it impossible for SaaS providers to develop and maintain a version of the application code for each user. As a result, SaaS services are usually designed as standardized functionalities. Users must generally accept the applications as provided,<sup>2</sup> and bear "lack-of-fit" costs for not using

fully customized products. This issue is less severe, however, in the traditional MOTS solution: The MOTS software's source code can be modified to meet a user's requirements. Thus, an in-house MOTS system is usually better customized than a SaaS product.

Moreover, SaaS uses a totally different pricing method than traditional approaches. IDC identifies two main types of SaaS pricing, i.e., a monthly subscription fee or a pay-per-use fee (Outsource2India 2011). Many providers, such as Salesforce.com, use the subscription fee pricing because it is clean and simple and gives them recurring revenues (Brill et al. 2010). Others charge based on transactions. Paying per (income generating) transaction reduces the users financial risk. For example, in the healthcare industry, pay-per-use is the dominant pricing method. It follows from the prevalent "Fee for Service" model. Some of the leading SaaS providers, such as Carestream, MedPACS, SourceRad, and Roentgen Works, which offer Radiology Picture Archiving and Communications Systems (PACS) to clinics and hospitals, charge a fee per transaction, which ranges from \$0.50 to \$3.<sup>3</sup> In this scenario, a transaction refers to one complete request from doctors to deliver, update, store, and manage patients' medical images. One of the authors of this research is very familiar with the 2014 installation of a university-wide students' billing system by a major SaaS vendor. The charges to each university unit are based on the number of students allocated, or billed for, by that software. Similarly, the medical billing SaaS CollaborateMD claimed that such a pricing model could help keep customers costs down without sacrificing convenience.<sup>4</sup> This transaction-based fee also applies to financial and accounting applications. Many small banks in Switzerland are using a similar transaction-pricing method when working with their SaaS providers. For financial applications, a transaction could be one complete loan application, a cashier's transaction from a bank or one result for an outsourced credit-scoring function.<sup>5</sup> For accounting applications, a transaction could be one line on the user's general ledger or a complete journal entry.<sup>6</sup> The transaction-based pricing approach offers the benefit of linking IT costs to transactions and hence is specifically applicable to

<sup>3</sup> MedPACS charges a flat fee of \$3.89 per study; SourceRad charges a fee of \$2 to \$3 per image delivery; and Roentgen Works offers image services at \$0.50 per study and archival services at \$0.86 per study.

<sup>4</sup> <http://www.collaboratemd.com>.

<sup>5</sup> For example, Wall Street System, a leading provider of global treasury, trading, and settlement solutions and services, charges customers per transaction (<http://www.bobsguide.com>).

<sup>6</sup> For example, the SaaS provider PremAPack in Singapore offers an accounting application for which it charges clients according to the number of transaction lines.

<sup>2</sup> Configuration is possible in SaaS, which uses predefined parameters to change software functionalities. For those beyond the predefined scope, however, SaaS cannot offer customization. The main difference between *customization* and *configuration* is that "configuration does not involve source code changes, whereas customization does" (Guo 2009, p. 2).

businesses that pull their earnings out of a single transaction. This paper, unlike most previous studies that focus on subscription-based pricing, investigates the implications of transaction-based pricing of SaaS.

We aim to capture the essence of the competition between SaaS and MOTS providers, which offer differentiated software products, have distinct cost structures, and use distinct pricing strategies and risk management approaches. Based on our visits to multiple sites, including Amazon Web Services, IBM On Demand, and SingTel,<sup>7</sup> as well as a number of small and medium size enterprises that are SaaS users, we can understand the practical pricing and risk issues from both providers' and clients' perspectives. We build a game-theoretical model that includes the multitenancy structure of SaaS, the capacity-hedging challenge of MOTS users, and the cost savings from economies of scale and lack-of-fit costs in the SaaS environment. Our analyses generate interesting results based on which we recommend practical strategies.

First, we show analytically that as the lack-of-fit (or data integration) costs of using SaaS due to its multitenancy nature decrease, the software market will demonstrate an evolution pattern from "SaaS fails," to "SaaS serves only residual users," to "SaaS competes with MOTS effectively," and finally to "SaaS dominates the market." We find that this pattern applies, regardless of the software's lifespan. We hence suggest that SaaS providers should invest heavily in reducing their lack-of-fit costs, thereby allowing users to effectively integrate their SaaS applications' data with the rest of the enterprise data model. Furthermore, we highlight the importance of developing an appropriate pricing strategy for each provider. For the SaaS providers with per-transaction charges, we recommend a higher value but lower price strategy so that it can pass the benefits of economies of scale on to users by steadily reducing prices. By contrast, the MOTS provider should adopt a totally different strategy. Above all, it should not resort to cutting prices. Instead, it should focus on enhancing the product's value by developing feature-rich and full-functionality software, which will allow it to retain monopolistic power in certain market segments.

The rest of the paper is organized as follows. Section 2 provides a review of the related literature. Section 3 describes our model. We analyze the competition and present major results in §4. Section 5 extends our model in two ways. First, we extend it to study the impact of increasing the software lifespan on the competitive outcome. Second, we include future expectations for changes in market conditions and analyze

the providers' pricing decisions in a dynamic marketplace. Section 6 recommends three competitive strategies for SaaS managers and MOTS providers based on our results. Section 7 concludes with a discussion of our study's limitations and options for future research.

## 2. Literature Review

Our work draws on two lines of literature, i.e., studies of selling and renting information goods, and research on the SaaS market.

In the literature on selling and renting information goods, many studies have used a single firm as the setting. Varian (2000), for example, considered a monopoly firm's choice when both selling and renting are feasible. Choudhary et al. (1998) analyzed the possibility of a monopoly firm's simultaneously selling and renting a software package. Sundararajan (2004) compared fixed-fee pricing and usage-based pricing for information goods for a monopoly firm. Fishburn et al. (1997) were among the few who compared selling and renting in a competitive market. In their paper, the two sellers' offerings were homogeneous; the only difference was the pricing method. Balasubramanian et al. (2011) also studied the competition between selling and renting, concentrating on the providers' endogenous choices of pricing strategy rather than on the characteristics of the competition itself. In this work, we focus on the competition between selling (MOTS) and renting (SaaS); SaaS services are priced based on transactions. In addition, we model users' in-house capacity management under the MOTS solution, which (to our knowledge) has not been analyzed in previous competition models.

Research on SaaS began with pioneering studies on ASPs in the 1990s. For example, Susarla et al. (2003, 2009) studied the contract design for ASPs and pointed out the incompleteness and opportunism in the ASP contract. Choudhary (2007) compared software publishers' incentive to invest in software development under both SaaS and a perpetual licensing model and found that in most cases SaaS creates a greater incentive to invest in product development and higher software quality. August et al. (2014) studied the security impact of a software vendor's versioning strategy when it chooses to offer an SaaS variant in addition to the existing on-premise product. Many other studies have focused on optimal strategies for the SaaS subscription-based pricing model. Some have studied the monopoly setting and how on-demand pricing affects monopolistic profits (Choudhary et al. 1998, Gurnani and Karlapalem 2001, Seidmann and Zhang 2010). These authors have concluded that a monopoly software provider could price discriminate and segment the market by simultaneously using subscription and licensing strategies. However, they ignored

<sup>7</sup> SingTel is the largest telecom company in Singapore. It began offering similar SaaS products to users in 2010.



the existence of established MOTS competitors. Others have studied the competition between SaaS (or an ASP) and shrink-wrapped software (Fishburn et al. 1997, Fishburn and Odlyzko 1999, Balasubramanian et al. 2011, Ma and Seidmann 2008, Fan et al. 2009) and found that on-demand software could pose a real competitive threat to traditional software.

We target a similar research question, but we model the competition in a different way. First, we focus on the impact of SaaS multitenancy on the competition by highlighting its ability to offer economies of scale while also imposing lack-of-fit costs on users. Because these two factors have opposite effects on the competitiveness of SaaS, the final outcome is worth examining. Second, unlike most previous studies that only differentiated competitors in terms of their pricing schemes, we also consider users' capacity-management problems under selling. Finally, we extend our model to a multiple stage setting to study the impact of the software lifespan and to analyze the competitive outcome when future technological developments could change the level of multitenancy.

More generally, our study relates to recent research on cloud computing, of which SaaS is typically considered to be a fundamental component.<sup>8</sup> The core concept of cloud computing is its convenient, on-demand network access to a shared pool of computing resources, including networks, servers, storage, applications, and services (Mell and Grance 2009, Katzan 2010). Although a few researchers have started to look at the economic aspects of cloud computing (for example, Retana et al. (2012) examined the economic value of the knowledge transfer mechanism in the cloud), most current research on the cloud continues to focus on related technical challenges. Thus, our work will contribute to cloud-computing research from an economic perspective.

### 3. The Model

We model the MOTS solution in §3.1, the SaaS solution in §3.2, and the users in §3.3. Table 1 summarizes our parameters and variables. Throughout the paper, we use capital letters for one-time costs/utilities/payments, and lowercase letters for per-unit or per-transaction costs/utilities. Subscripts "m" and "s" indicate the MOTS and SaaS provider, respectively.

#### 3.1. The MOTS Solution

The MOTS provider sells software applications to firms that need enterprise software. The provider charges a one-time up-front fee  $P_m$ . We assume that the provider offers only one version of the product and that  $P_m$  is not a function of the users' transaction

**Table 1** Modeling Notation and Definitions

Notation	Definitions
$C_{\text{cust}}$	Customization costs for the in-house MOTS software system
$P'_m; P_m^*$	Optimal one-time price charged by the MOTS provider in the pre-SaaS market; and in the competition
$p_s^*$	Optimal per-transaction price charged by the SaaS provider in the competition
$Q_i^r$	The MOTS user $i$ 's installed infrastructure capacity
$D_i; d_i$	User's actual and expected demand (in terms of the number of transactions)
$d^r; d^*$	The indifferent user in the pre-SaaS market; and in the MOTS-SaaS competition
$d^{**}; d_{\text{switch}}$	The indifferent user and the marginal switcher in the MOTS-SaaS two-stage competition with dynamic lack-of-fit costs
$c$	User's unit infrastructure costs under the MOTS solution
$c_m$	User's per-transaction service costs under the MOTS solution
$c_s$	The SaaS provider's per-transaction service costs
$u$	User's per-transaction value under the MOTS solution when its infrastructure capacity limit is not reached yet
$u_r (u_r < u)$	User's per-transaction value when the actual demand for transaction processing exceeds the installed infrastructure capacity ( $u_r < u$ after considering delay costs)
$\Delta u = u - u_r$	User's per-transaction utility loss when the infrastructure capacity is exceeded
$t$	Lack-of-fit costs per transaction for SaaS users
$u - t$	User's per-transaction value under the SaaS solution
$t_H(t_L)$	The increased (decreased) lack-of-fit costs in the second stage of the MOTS-SaaS two-stage competition
$\theta$	Volatility of users' actual transaction volume (for users with $d_i > \theta$ )

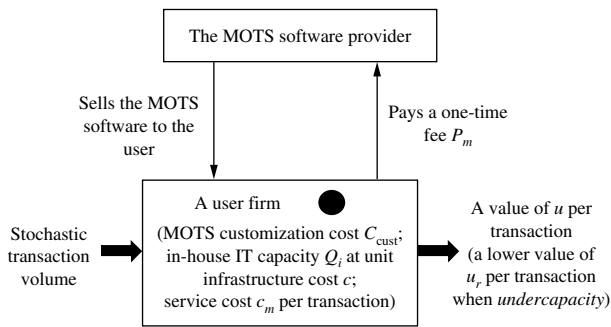
volume.<sup>9</sup> Source code for MOTS software is modifiable, and so it can be customized to fit an individual user's needs. The user will hire its own in-house IT personnel or a third-party professional to perform this customization process, incurring a cost of  $C_{\text{cust}}$ .

Users install the MOTS software in-house and maintain it using their internal IT capability. They must install hardware and create infrastructure, hire IT staff, and organize an internal IT group to provide the ongoing maintenance, security, and capacity management. By keeping these IT resources in-house, the user incurs infrastructure costs. Let  $c$  denote the user's unit infrastructure costs. In addition, when the software system is in use, the user bears operational costs that include data backup and management, monitoring, troubleshooting, and a variety of transaction-related services. These are the user's service costs under the MOTS solution. Here they are denoted by  $c_m$  per transaction. Following Huang and Sundararajan (2011), we distinguish between infrastructure costs

<sup>8</sup> Three components comprise cloud computing: SaaS, Infrastructure as a Service (IaaS), and Platform as a Service (PaaS).

<sup>9</sup> This is because the MOTS provider can monitor the number of users or installations, but it is difficult to monitor the actual transaction volume on the user's side, which varies over time.

Figure 1 The MOTS Solution



Note. The black dot indicates the location of the software system.

and service costs: The former are the fundamental IT investment, maintenance, and management costs, regardless of whether the system is in use; the latter are transaction-based costs, incurred only when a transaction takes place.

Users face stochastic demand in terms of the number of transactions involving software use.<sup>10</sup> Ex ante, a user cannot precisely estimate its actual transaction demand. It must determine and install the proper infrastructure capacity in-house, based on expected demand. We denote a user  $i$ 's infrastructure capacity by  $Q_i$  based on the quantity of transactions that can be supported. When the transaction demand is satisfied using the in-house system, the user gains a utility value of  $u$  per transaction. If, however, the actual demand for transaction processing exceeds the installed capacity, the user faces *insufficient capacity* and may have to incur some level of processing delay and poor service delivery. In this instance, the utility per transaction processing is  $u_r$  and  $u_r < u$ . On the other hand, when actual demand is less than the installed infrastructure capacity, the user faces *overcapacity* and still bears the infrastructure cost of maintaining idle IT power. Figure 1 depicts the MOTS solution.

### 3.2. The SaaS Solution

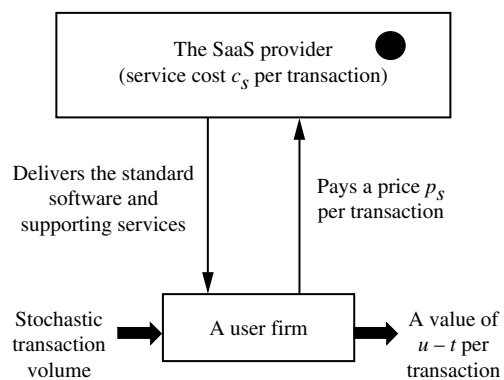
The SaaS provider offers large-scale data centers, a large number of servers, strong network infrastructure, and high-performance computing power. Built on extensive shared computing resources, SaaS providers often face the challenge of capacity overprovision. Their resources are used at a very low rate. For example, the Gartner report shows that typical data-center use varies between 7% and 12% (Glanz 2012). In addition, many SaaS providers now deploy their applications on a platform offered by external PaaS providers and run their back-end servers on-demand from IaaS

providers. An increasing number of SaaS providers, for instance, are using Amazon Web Services (AWS) as an ideal infrastructure platform for their business. The AWS usage-based pricing model and scale-on-demand infrastructure align well with their revenue and operating models. These practices make SaaS resemble an ample-capacity power station. To capture this reality, our model will not consider limited capacity and congestion issues for the SaaS solution, similar to Huang and Sundararajan (2011).

The provider is responsible for all IT services necessary to enable software use and incurs service cost  $c_s$  per transaction. The service cost  $c_s$  might be higher or lower than the user's in-house service cost  $c_m$  under the MOTS solution. When  $c_s < c_m$ , the SaaS provider realizes cost savings from economies of scale when serving a large number of users;  $c_s > c_m$ , by contrast, suggests that the user's in-house IT group has expertise in and specific knowledge of the user's IT requirements and therefore can serve the user more cost-efficiently. As described above, we assume that the provider largely relies on the external PaaS and IaaS to build its infrastructure. In other words, separate ongoing infrastructure maintenance costs for the SaaS provider are not considered in the model. Any such costs could be factored into  $c_s$ .

Under the SaaS solution, users must generally accept the application as provided due to its multi-tenancy nature. Therefore, we assume that for a user the SaaS application is not well customized, especially compared to an in-house software system. Each transaction gives the user a value of  $u - t$ . The parameter  $t$  measures the user's disutility from using a not-fully-customized product. In many cases, it also represents the cost of ensuring that the SaaS product will work smoothly with the user's existing IT components. In the rest of the paper, we call  $t$  a user's lack-of-fit cost. The SaaS provider uses a transaction-based pricing scheme and charges price  $p_s$  per transaction. Figure 2 illustrates the SaaS solution.

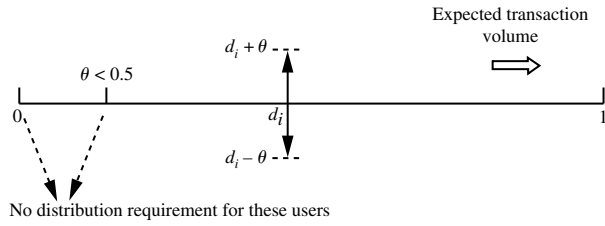
Figure 2 The SaaS Solution



Note. The black dot indicates the location of the software system.

<sup>10</sup> As discussed in §1, we focus on businesses that pull earnings out of single transactions. Examples of transaction-oriented software include, but are not limited to, healthcare software, financial trading systems, and accounting applications.

Figure 3 Distribution of Users' Transaction Volume



### 3.3. The Users

When modeling users, we capture two characteristics. First, users have different IT needs as measured by expected transaction volume. Some firms may expect to use the software application more frequently than others, so these firms have greater IT needs.

Second, each user's actual transaction volume is stochastic. Users are uniformly distributed on a unit-length line normalized from 0 to 1. A user's location on this line represents its expected transaction volume. The actual transaction volume of a user at  $d_i$  is a random variable uniformly distributed on  $[d_i - \theta, d_i + \theta]$ , where  $d_i$  is the user's expected use of the software (i.e., the number of transactions), and  $\theta$  measures the volatility of the transaction volume. However, this uniform distribution applies only to users with  $d_i \geq \theta$ .<sup>11</sup> For users with  $d_i < \theta$ , we do not assume any specific distribution. All we need is that their expected transaction volume is  $d_i$ .<sup>12</sup> In addition, we require that our distribution assumption  $[d_i - \theta, d_i + \theta]$  can be applied to the majority of the users in the market; i.e.,  $\theta < 0.5$ .<sup>13</sup>

We make the following general parameter assumption:  $C_{\text{cust}} \leq (u - c_m - c) - c\theta(\Delta u - c)/\Delta u$ . If  $C_{\text{cust}}$  exceeds this upper bound, it is too costly to customize the MOTS software. No user will opt for the MOTS solution, which is a case of no interest.

## 4. Analysis and Results

### 4.1. The Pre-SaaS Market (The Benchmark)

Before SaaS, users accessed a software system by installing it in-house. We use this pre-SaaS market as

<sup>11</sup> This is to avoid a positive probability on negative transaction volume.

<sup>12</sup> It can be shown that these users are "insignificant" regardless of their demand distribution. With a low transaction volume  $d_i < \theta$ , they will be priced out of the market in the pre-SaaS scenario. Their concrete demand distribution will not affect our analytical results at all as long as their expected demand is  $d_i$ .

<sup>13</sup> Note that the total market size is 1. So  $\theta < 0.5$  means that more than half of the users will satisfy  $d_i \geq \theta$ . This ensures that the assumed distribution could be applied to the majority of the users though not the whole user population. In §4.3 and in the online appendix (available as supplemental material at <http://dx.doi.org/10.1287/isre.2015.0571>), we also study two other types of distributions, i.e., uniform distribution in  $[a, a + 1]$  with  $\theta < a$ , and in  $[(1 - \theta)d_i, (1 + \theta)d_i]$ , which avoid the issue of negative transaction numbers. We show that all major results are robust.

our benchmark. The MOTS provider determines its price  $P_m$ . Given the price, users make two decisions: (1) whether to buy the software and, if so, (2) the proper infrastructure capacity to install in-house.

We solve the problem using backward induction. Consider a user  $i$  at  $d_i$  ( $d_i > \theta$ , see Figure 3). Denote the actual transaction volume of user  $i$  as  $D_i$ , and the user's in-house IT capacity as  $Q_i$ . This user must determine an optimal capacity level  $Q_i^*$  by solving  $\max_{Q_i} E[(u - c_m) \min\{D_i, Q_i\} + (u_r - c_m)(D_i - Q_i)^+ - P_m - C_{\text{cust}} - cQ_i]$ . The first two terms in the expression above represent user  $i$ 's expected utility from using the software: When demand is less than capacity, each transaction involving in-house software use will generate a net utility of  $(u - c_m)$ ; when demand exceeds capacity, the excess demand portion,  $(D_i - Q_i)^+$ , will generate a lower net utility of  $(u_r - c_m)$ . The last three terms in the expression represent the user's total costs under the MOTS solution, which consists of three parts: the software purchase fee paid to the provider, the internal cost of customizing and integrating the software, and the cost of establishing and maintaining the required in-house IT capacity. Following Dewan and Mendelson (1990), we assume a linear function for capacity costs to keep our analysis tractable. In practice, capacity costs could be a step function, as suggested by Huang and Sundararajan (2011). We recognize that a linear function is the best first-order approach for a step function, as shown by Zimmerman (1979).

Solving the optimization problem gives the following:

$$Q_i^* = d_i + \theta \left(1 - \frac{2c}{\Delta u}\right). \quad (1)$$

PROOF. See the appendix.

The optimal in-house IT capacity  $Q_i^*$  deviates from the user's expected transaction volume  $d_i$  by a factor of  $\theta(1 - 2c/\Delta u)$ . The deviation may be positive or negative, suggesting that the user's optimal in-house capacity could be more or less than its expected demand depending on the relative magnitude of  $c$  (the unit infrastructure cost) and  $\Delta u$  (the utility loss per transaction due to *undercapacity*).

The expected utility for user  $i$ , if it uses the MOTS solution, is

$$Eu_{\text{MOTS}}(d_i) = d_i(u - c - c_m) - P_m - C_{\text{cust}} - \frac{c(\Delta u - c)}{\Delta u} \theta. \quad (2)$$

The first term is the expected net value from using the software after considering both service costs and infrastructure costs; the second term is the user's one-time payment to the MOTS provider; and the third term is the user's customization cost. The last term represents the user's utility loss due to demand



uncertainty; its magnitude increases with the transaction volatility level ( $\theta$ ).<sup>14</sup>

The user will choose to purchase the software if its expected utility, Equation (2), is nonnegative. The marginal user, who gets zero expected utility from using the MOTS, is at

$$d' = \frac{P_m + C_{\text{cust}}}{u - c - c_m} + \frac{c(\Delta u - c)}{\Delta u(u - c - c_m)}\theta. \quad (3)$$

The provider's market share is  $1 - d'$ . The price  $P_m$  is chosen to maximize the provider's profit:  $\max_{P_m} \Pi_{\text{MOTS}} = P_m(1 - d')$ .

The optimal price is

$$P'_m = \frac{u - c - c_m}{2} - \frac{1}{2}C_{\text{cust}} - \frac{(\Delta u - c)c}{2\Delta u}\theta. \quad (4)$$

Consequently, the marginal user is at

$$d' = \frac{1}{2} + \frac{C_{\text{cust}}}{2(u - c_m - c)} + \frac{c(\Delta u - c)\theta}{2\Delta u(u - c_m - c)} > \frac{1}{2}.^{15} \quad (5)$$

Equation (5) shows that the majority of users will be excluded from the MOTS solution. This is consistent with the observation that small- and medium-size firms typically have had no access to in-house enterprise software systems because of the high costs of purchasing, customizing, and maintaining software. In addition, Equation (4) indicates that the MOTS provider will reduce its price when the user's transaction volatility ( $\theta$ ) increases. Plugging this optimal price into Equation (2), we find that the reduced price partially compensates for a user's utility loss due to demand uncertainty. Moreover, the marginal user's location ( $d'$ ) moves to the right as transaction volatility increases, implying that the provider's market share decreases even though it lowers its selling price. Consequently, the provider's profit declines. This means that the MOTS provider also incurs losses when users' demand uncertainty increases. Hence, we conclude:

**PROPOSITION 1 (THE PRE-SaaS-MARKET PROPOSITION).** *In a market without the SaaS option, users with low transaction volume are priced out of the market. When the volatility of transaction volume increases, both the buyers and the MOTS provider are worse off.*

The fundamental challenge faced by MOTS users is always capacity hedging. Users must specify service

capacity in advance based on anticipated transaction volume; therefore, they are unable to quickly scale up or down according to changing business needs. Users anticipate such risk associated with demand uncertainty and, to some extent, will shift it to the MOTS provider: When the risk increases, users' willingness to pay for the MOTS software declines. As a result, the software provider must charge a lower price, leading to a lower profit.

#### 4.2. The Market with MOTS-SaaS Competition

We now analyze the competition between the providers. Throughout this work, we will focus on the price competition between the two. We do not, however, evaluate their decisions in terms of either optimal software quality or optimal service capacity. In other words, the MOTS provider's R&D investment decisions as to its software quality and the SaaS provider's IT investment decisions as to its infrastructure capacity are not within the scope of this work.

The MOTS and SaaS providers simultaneously decide their prices,  $P_m$  and  $p_s$ . Given the prices, users decide to use one or the other or to stay out of the market. Consider user  $i$  with expected transaction volume  $d_i$ . It gains an expected utility of  $Eu_{\text{SaaS}}(d_i) = (u - p_s - t)d_i$  from the SaaS option, and  $Eu_{\text{MOTS}}(d_i) = d_i(u - c - c_m) - P_m - C_{\text{cust}} - (c(\Delta u - c)/\Delta u)\theta$  from the MOTS offering (Equation (2)). In equilibrium, users with low transaction volume choose the SaaS and users with high transaction volume choose the MOTS. The indifferent user, denoted as  $d^*$ , gets the same expected utility from using MOTS and SaaS, and thus  $Eu_{\text{SaaS}}(d^*) = Eu_{\text{MOTS}}(d^*)$ :

$$d^* = \frac{P_m + C_{\text{cust}}}{p_s + t - c - c_m} + \frac{c(\Delta u - c)\theta}{\Delta u(p_s + t - c - c_m)}. \quad (6)$$

The MOTS provider serves users in  $[d^*, 1]$ , with a market share of  $1 - d^*$ ; the SaaS provider serves users in  $[0, d^*]$ , with a market share of  $d^*$ . Both maximize profit by choosing prices:  $\max_{P_m} \Pi_{\text{MOTS}} = P_m(1 - d^*)$ ;  $\max_{p_s} \Pi_{\text{SaaS}} = \int_0^{d^*} (p_s - c_s)x dx$ ; s.t.:  $c_s \leq p_s \leq u - t$ ;  $P_m \geq 0$ ;  $0 \leq d^* \leq 1$ . There are some constraints. The SaaS provider's price  $p_s$  has an upper bound  $u - t$  beyond which no user will gain any positive utility from using the SaaS product and hence none will opt for SaaS, and also a lower bound  $c_s$  below which the SaaS provider cannot make nonnegative profit. The MOTS price  $P_m$  must also be nonnegative for the MOTS provider to survive. In addition, the marginal user's position is bounded:  $0 \leq d^* \leq 1$ .

Next, we solve this constrained optimization problem. We begin by stating the conditions under which one provider can price strategically and drive its competitor out of the market. We then present the pricing strategy for each provider when they coexist in the market, and discuss the equilibrium market

<sup>14</sup> The condition  $\Delta u - c > 0$  should hold. Otherwise, users will always choose not to build any in-house capacity.

<sup>15</sup> To see  $d' > 0.5 > \theta$ : This is consistent with our earlier statement that users with low transaction volume  $d_i < \theta$  will be priced out of the market in the pre-SaaS scenario. In addition,  $d' < 1$  is always satisfied under our general parameter assumption  $C_{\text{cust}} \leq (u - c - c_m) - c\theta(\Delta u - c)/\Delta u$ .



outcomes (Propositions 3 and 4). Finally, we draw a complete competitive diagram for different parameter regimes and discuss the key factor that determines each provider's competitive power (Proposition 5).

Under certain circumstances, a single software solution, SaaS or MOTS, can drive its competitor out of the market and therefore dominate the whole market. The *One-Provider Dominance Proposition* (Proposition 2) gives both the condition under which on-demand software providers will completely replace MOTS providers and the condition under which the SaaS business model will ultimately fail and become a historical footnote.

**PROPOSITION 2 (ONE-PROVIDER-DOMINANCE PROPOSITION).** *There are two threshold values for the lack-of-fit costs  $t$ ,  $t_1 = c(\Delta u - c)\theta/(2\Delta u) + C_{\text{cust}}/2 + c + (c_m - c_s)$  and  $t_2 = u - c_s$ . When  $t < t_1$ , the SaaS provider serves the whole market; when  $t > t_2$ , the SaaS provider fails and exits the market.*

PROOF. See the appendix.

When the standardized SaaS application imposes very small lack-of-fit costs on users,  $t < t_1$ , the SaaS provider can price the MOTS provider out of the market. In this case, all users will access the software through the SaaS. We find that  $t_1$  increases in both  $\theta$  and  $(c_m - c_s)$ . This suggests that it will become easier for the SaaS provider to dominate the market as users' transaction volatility increases and as the SaaS solution's cost efficiency increases. On the other hand, when SaaS users must incur large lack-of-fit costs,  $t > t_2$ , the SaaS provider cannot operate profitably. This is because when  $t$  is too big, at any price the SaaS provider might charge, one of the two constraints will be violated, i.e., either (1) users will obtain no utility or (2) the SaaS provider will obtain no profit. In this case, only the MOTS solution exists.

When the lack-of-fit cost is in the middle range,  $t_1 \leq t \leq t_2$ , the providers will coexist in the market. Our analysis further identifies a threshold value:  $t^* = (u - 2c_s + c_m + c)/2$ . Lemma 1 shows the order of these three critical values of  $t$ .

LEMMA 1.  $t_1 \leq t^* \leq t_2$ .

In  $[t_1, t_2]$ , where the two providers coexist, equilibrium prices are different depending on whether  $t$  is greater or smaller than  $t^*$

$$(1) \ t^* \leq t \leq t_2: \left\{ \begin{aligned} p_m^* &= \frac{u - c - c_m}{2} - \frac{1}{2}C_{\text{cust}} \\ &\quad - \frac{(\Delta u - c)c}{2\Delta u}\theta; p_s^* = u - t \end{aligned} \right\}.$$

$$(2) \ t_1 \leq t < t^*: \left\{ \begin{aligned} p_m^* &= (c_s + t - c - c_m) - \frac{1}{2}C_{\text{cust}} \\ &\quad - \frac{(\Delta u - c)c}{2\Delta u}\theta; p_s^* = 2c_s + t - c - c_m \end{aligned} \right\}.$$

PROOF. See the appendix.

As a result, the market outcome will exhibit totally different features as described in Propositions 3 and 4.

**PROPOSITION 3 (THE SEGMENTED-MARKET PROPOSITION).** *When the lack-of-fit costs are high ( $t^* \leq t \leq t_2$ ), the two providers coexist but do not compete with each other. They share the market in an entirely segmented way. The MOTS provider is still a monopoly in its own market segment, and the SaaS provider's entry into the market has no effect on either the MOTS provider or users.*

If users face high lack-of-fit costs for using the SaaS application, the SaaS provider has no competitive effect: It will not affect the MOTS provider's price, market share, or profit. The MOTS provider still behaves as if it were the monopolist in the market, charging a monopolistic price and serving the monopoly market share as in the benchmark case. In this case, the SaaS provider will make its profit by serving residual customers, those who are priced out of the market in the benchmark case (in  $[0, d']$ ).

Only when lack-of-fit costs are reduced below the threshold value  $t^*$  will competition be seen. This is described in Proposition 4.

**PROPOSITION 4 (THE COMPETITIVE-MARKET PROPOSITION).** *When the lack-of-fit costs are low ( $t_1 \leq t < t^*$ ), the two providers directly compete with each other. The competition results in a lower price, smaller market share, and less profit for the MOTS provider (compared to the pre-SaaS market). Each user is better off, and the total consumer surplus increases.*

When  $t < t^*$ , the SaaS provider's optimal price can be rewritten as  $p_s^* = c_s + t - c - (c_m - c_s)$ , which has two counterintuitive features. First, we find that the SaaS provider should use a "higher value, lower price" strategy. Its price decreases with users' lack-of-fit costs  $t$ . In other words, when the SaaS provider can deliver a higher value to users (i.e., when  $t$  decreases), it should reduce its price. Second, we find that the SaaS provider should use a "higher cost efficiency, lower price" strategy. Its price decreases with  $(c_m - c_s)$ , which measures the difference between the user's in-house service costs and the SaaS provider's service costs per transaction. When  $(c_m - c_s)$  increases, the SaaS provider becomes more cost-efficient in providing IT services, and we find that it should reduce its price. Both the "higher value, lower price" and the "higher cost efficiency, lower price" strategies enable the SaaS provider to compete aggressively for large-volume users, who are more profitable than small-volume users due to the SaaS provider's use of a per-transaction pricing scheme. As a result, the increase in market share from attracting large-volume users will offset the reduced profit per transaction and eventually generate a higher total profit for the SaaS.

*The Competitive-Market Proposition* (Proposition 4) states that when  $t_1 \leq t < t^*$ , a common competition outcome is reached: The SaaS provider's entry into the market has negative effects on its competitor and positive effects on users. The MOTS provider must reduce its price, but it still loses market share to the SaaS provider and eventually obtains lower profits compared to the benchmark case. SaaS benefits users in multiple ways. Users with low transaction volume, who would otherwise be excluded from the market, can afford the software through the SaaS provider's pay-as-you-go pricing scheme. Users with medium transaction volume, who would otherwise install an in-house system, use the SaaS application because it offers a cheaper solution. Finally, users with high transaction volume stay with the in-house solution but pay less to the MOTS provider because of the competitive pressure from the SaaS alternative. Therefore, each user is better off, and the total consumer surplus is strictly improved.

Lemma 2 states the impact of transaction volatility ( $\theta$ ) on the competitive outcome.

**LEMMA 2.** *When users' transaction volatility increases, the MOTS provider will reduce its price,  $dP_m^*/d\theta < 0$ , whereas the SaaS provider keeps its price unchanged,  $dp_s^*/d\theta = 0$ . Even so, the MOTS provider loses market share to the SaaS provider,  $dd^*/d\theta < 0$ .*

The SaaS on-demand feature allows users to enjoy full scalability and to handle possible demand fluctuations at no risk. Users are not negatively affected by demand volatility, so the SaaS provider does not need to adjust its price to accommodate a more volatile market. To remain attractive to potential clients, the MOTS provider must share such uncertainty risks with them by reducing its price, as it would have done in the pre-SaaS market. Interestingly, we find that even if the SaaS price remains unchanged and the MOTS price is reduced, more users tend to choose the SaaS alternative. This means that in addition to the cost savings offered by SaaS, users also value its scalability.

We can now draw a complete picture of SaaS-MOTS competition. Depending on the value of the SaaS application's lack-of-fit cost, Figure 4 depicts four regimes. In each regime, the competition and market outcomes will be as follows:

*Regime 1.*  $t < t_1$ : The SaaS provider serves the whole market. The market outcome in this regime is summarized in the *One-Provider-Dominance Proposition* (Proposition 2).

*Regime 2.*  $t \in [t_1, t^*]$ : The SaaS and MOTS providers coexist in the software market, and the competition between them benefits each market user. The competition in this regime is characterized in *The Competitive-Market Proposition* (Proposition 4).

*Regime 3.*  $t \in [t^*, t_2]$ : The SaaS and MOTS providers coexist in the software market. The SaaS provider, however, only serves residual users, and the MOTS provider still acts as a monopolist. The market outcome in this regime is characterized in *The Segmented-Market Proposition* (Proposition 3).

*Regime 4.*  $t > t_2$ : The SaaS provider is unable to make a profit, and the MOTS is the only software solution. The market outcome in this regime is characterized in the *One-Provider-Dominance Proposition* (Proposition 2).

Note that in Regime 3, users in the interval  $[0, d^*]$  use the SaaS offering, but their utility is zero. The SaaS provider is extracting all consumer surpluses at the equilibrium price  $p_s^* = u - t$ . Users are, in fact, indifferent between using the SaaS services or the MOTS. Theoretically we can assume the tie-breaking rule, i.e., that users will choose to join the SaaS provider when indifferent. In practice, the provider needs to adjust its price to  $p_s^* = u - t - \Delta$ , where  $\Delta$  is a discount that is too small to alter the whole market equilibrium but which can give users a positive yet negligible margin to entice them to the SaaS services.<sup>16</sup>

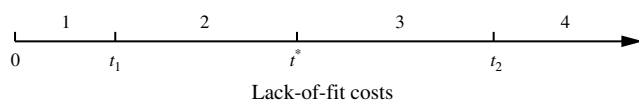
Under different values of  $t$ , the market exhibits different features. When  $t$  is low, the SaaS solution dominates, revealing the possibility of "all software in SaaS." When  $t$  is moderate, the SaaS provider shares the market and competes with the MOTS. When  $t$  gets larger, the SaaS provider loses its competitive advantage and eventually is unable to survive. Thus, we conclude:

**PROPOSITION 5 (LACK-OF-FIT-COSTS PROPOSITION).** *When the lack-of-fit costs of the SaaS application decrease, the SaaS provider's ability to compete increases monotonically.*

What we find here mirrors what Varian (2000) shows in a monopoly setting: When the quality difference between selling and renting a product is small enough, the renting strategy dominates. We show that a similar conclusion holds in the duopoly competition: When the two providers' offerings are close enough (i.e., when the lack-of-fit cost  $t$  is small enough), the SaaS provider, which charges based on use, can drive the MOTS provider, which charges a one-time fee, out of the market.

<sup>16</sup> We thank the anonymous reviewer for pointing this out. This is important in our setting since the indifferent users here are not a single user, but a group of users in an interval.

**Figure 4** A Complete Diagram of the Price Competition



Other factors that affect SaaS performance are the volatility of demand for users ( $\theta$ ) and the SaaS provider's cost efficiency relative to the user's internal IT group ( $c_m - c_s$ ). We see that when  $\theta$  and  $(c_m - c_s)$  increase, the critical value  $t_1$  increases:  $\partial t_1 / \partial \theta > 0$  and  $\partial t_1 / \partial (c_m - c_s) > 0$ , and the position of an indifferent user also increases:  $\partial d^* / \partial \theta > 0$  and  $\partial d^* / \partial (c_m - c_s) > 0$ . This means that, all else being equal, when users' demand volatility increases, and when the SaaS provider can provide supporting services with a higher cost efficiency, the SaaS provider's competitive power increases. It is expected to gain a significant market share and is more likely to take over the whole market.

### 4.3. Model Discussion and Robustness Check

As briefly discussed in §3, our model assumes ample IT capacity for the SaaS provider. There are several reasons for such an assumption. First, ample capacity is the fundamental business concept for SaaS (and also for other types of cloud computing models). SaaS providers build their service systems with ample capacity as a way of reducing the risk of congestion and to guarantee the fulfillment of users' scale-up requests at any time, which users consider to be SaaS's key value. The reality, for most on-demand service providers, is that they typically provide much more capacity than needed.<sup>17</sup> In addition, many SaaS providers choose to obtain their infrastructure capacity from external PaaS and IaaS, so that they are operating with complete scalability. Our model is designed to capture this reality by abstracting out the congestion effects in the SaaS setting.

Second, even with that said, we still agree that at a certain point, SaaS providers will need to consider optimizing their infrastructure capacity. For this study, however, we have chosen to focus on competition between the two providers' pricing decisions, rather than on their investment decisions from a long-term perspective. We consider the competition setting in which the MOTS provider's R&D investment decision on software quality and the SaaS provider's IT investment decision on infrastructure capacity have been made. To study the providers' quality/capacity competition is appealing, but it would shift the paper's focus from the aspects of pricing competition that our model aims to highlight.

We made several other simplifying assumptions without loss of generality. For example, we assume that the MOTS software is highly customized, whereas the

SaaS application tends to come with higher lack-of-fit costs. Such an assumption does not imply that the MOTS product must be a perfectly fit system; in practice, an in-house software system could also impose lack-of-fit costs on users. All that really matters for modeling purposes is the marginal disutility for users of MOTS versus SaaS. In fact, our model addresses a case in which the MOTS software has unusually high lack-of-fit costs, i.e.,  $t < 0$ . This leads to the trivial outcome that the SaaS provider will dominate the marketplace.

We assume that both providers use "basic" pricing strategies. The SaaS provider charges a fee per transaction, and the MOTS provider requires an outright purchase of the system. Here, our research objective is to characterize the nature of the competition between SaaS offerings and the more conventional MOTS software packages. In doing so, we focus on the fundamental operational differences between the two delivery models, trying to capture the salient advantages and limitations of each approach. The SaaS industry is rapidly expanding and evolving, and not all SaaS providers are using the same pricing method discussed above. The practical implications of our research, therefore, are primarily relevant to SaaS systems with per-transaction charges.

Finally, our outcomes are robust to the assumption about the distribution of users' transaction volumes. In the current model, we assume that users' expected transaction volumes are distributed over a unit-length line  $[0, 1]$ , and that the majority of users have their actual transaction volume follow a uniform distribution of  $[d_i - \theta, d_i + \theta]$ . We could make the setting more general by assuming that (1) the users' expected transaction volumes fall within a more general range of  $[a, a + 1]$  for any positive number  $a > \theta$ , or (2) a user's demand volatility is proportional to its expected transaction volume, and it follows a uniform distribution on  $[(1 - \theta)d_i, (1 + \theta)d_i]$  with  $\theta < 1$ . Our major findings and conclusions will still hold under these distributions.<sup>18</sup>

## 5. Model Extension and Analysis

We now study two extensions of the current model. Section 5.1 extends it to a multistage time horizon, and §5.2 extends it to a setting with dynamic lack-of-fit costs.

### 5.1. Competition in a Multistage Time Horizon: The Impact of Software Lifespan

Most software systems have a typical lifespan of about three to five years before they become obsolete. How does the software's lifespan affect our findings? For a

<sup>17</sup> For example, Marc Benioff, CEO of Salesforce.com, in his seminar talk in Singapore in 2010, stated that the company had built more than 3,000 high-performance servers and that a large proportion of their computing power had been kept idle to ensure constant always-on-stand-by service capacity.

<sup>18</sup> The proof is available on request.



MOTS user, the initial purchase cost  $P_m$  and customization cost  $C_{\text{cust}}$  are incurred only once. Intuitively, when the MOTS system is used for a longer time, users will value the MOTS solution more. It is therefore reasonable to expect that the advantage of MOTS becomes more apparent in the competition. Yet to what extent will this change our results? Will it change the competition regimes completely, or only revise the relative competitive power of the two providers but not the nature of their high-level competitive diagram? To address these questions, we extend our analysis to a multistage time horizon. We assume that the MOTS system can be used for  $k$  stages. All other model constructs are the same as in §3. We check the robustness of our results in this extended competition timeline.

MOTS user  $i$  must determine an optimal periodic capacity level  $Q_i^k$  by solving

$$\max_{Q_i^k} E[k(u - c_m) \min\{D_i, Q_i^k\} + k(u_r - c_m)(D_i - Q_i^k)^+ - kcQ_i^k - P_m - C_{\text{cust}}].^{19}$$

The user gains value from using the MOTS system and bears ongoing infrastructure costs for maintenance and management of the in-house IT capacity in  $k$  stages, but it only incurs the initial purchase and customization costs once. The optimal  $Q_i^k$  is derived using the same method as in §4, and we find that  $Q_i^k = Q_i^*$ . It is intuitively correct that  $Q_i^k$  is not a function of  $k$ . Instead, it depends on the mean and volatility of demand in each stage; all one-time costs are canceled out in the optimization process.

We can therefore write user  $i$ 's total utility of using MOTS and SaaS in  $k$  stages as:  $Eu_{\text{MOTS}}(d_i)^k = kd_i(u - c - c_m) - k(c(\Delta u - c)/\Delta u)\theta - P_m - C_{\text{cust}}$  and  $Eu_{\text{SaaS}}(d_i)^k = k(u - p_s - t)d_i$ . The indifferent user is at  $d^{*k} = (P_m + C_{\text{cust}})/(k(p_s + t - c - c_m)) + c(\Delta u - c)\theta/(\Delta u(p_s + t - c - c_m))$ , and the two providers' optimization problems are:  $\max_{P_m^k} \Pi_{\text{MOTS}} = P_m^k(1 - d^{*k})$ ;  $\max_{p_s^k} \Pi_{\text{SaaS}} = k \int_0^{d^{*k}} (p_s^k - c_s)x dx$ ; s.t.:  $c_s \leq p_s^k \leq u - t$ ;  $P_m^k \geq 0$ ;  $0 \leq d^{*k} \leq 1$ . This could be solved in the same way as in §4. In what follows, we only present the results and omit the detailed derivations.

We get three critical values for  $t$ :  $t_1^k = c(\Delta u - c)\theta/(2\Delta u) + C_{\text{cust}}/(2k) + c + (c_m - c_s)$ ,  $t_2^k = u - c_s$ ,  $t^{*k} = (u - 2c_s + c + c_m)/2$ ; and find  $t_1^k \leq t^{*k} \leq t_2^k$  (similar to Lemma 1), and  $t_1^k \leq t_1$ ,  $t^{*k} = t^*$ , and  $t_2^k \leq t_2$ .

<sup>19</sup> IT infrastructure costs include IT investment, maintenance, management, upgrade, and replacement costs. Such costs are ongoing, not one-time. They could be different in each stage, and a non-monotonic change over time ( $k$ ) is likely. Although it is a simplifying expression here that the infrastructure costs are constant over stages ( $c$ ), it could be easily modified to incorporate "varying  $c$  in each stage." If we have different  $c_k$  for each  $k$ , we need only denote  $c = (\sum_k c_k)/k$ , and all derivations and expressions will remain the same.

The equilibrium prices and outcomes depend on  $t$ , as follows:

(1) When  $t < t_1^k$ , the SaaS provider serves the whole market.

(2) When  $t_1^k \leq t < t^{*k}$ , the equilibrium prices are

$$\left\{ P_m^{*k} = k(c_s + t - c - c_m) - \frac{1}{2}C_{\text{cust}} - \frac{k(\Delta u - c)c}{2\Delta u}\theta; \right. \\ \left. p_s^* = 2c_s + t - c - c_m \right\},$$

and the indifferent user is

$$d^{*k} = \frac{1}{2} + \frac{C_{\text{cust}}}{4k(c_s + t - c - c_m)} + \frac{c(\Delta u - c)\theta}{4\Delta u(p_s + t - c - c_m)}.$$

(3) When  $t^{*k} \leq t < t_2^k$ , equilibrium prices are

$$\left\{ P_m^{*k} = \frac{k(u - c - c_m)}{2} - \frac{1}{2}C_{\text{cust}} - \frac{k(\Delta u - c)c}{2\Delta u}\theta; p_s^{*k} = u - t \right\},$$

and the indifferent user is

$$d^{*k} = \frac{1}{2} + \frac{C_{\text{cust}}}{2k(u - c_m - c)} + \frac{c(\Delta u - c)\theta}{2\Delta u(u - c_m - c)}.$$

(4) When  $t > t_2^k$ , the SaaS provider fails and exits the market.

The above cases (1) and (4) correspond to Regimes 1 and 4 in Figure 4, as summarized by Proposition 2. Case (2) is just Regime 2, described by Proposition 4, and Case (3) is just Regime 3, described by Proposition 3. We therefore conclude that in the multistage setting, the major results derived from the previous one-stage setting are all valid; there will still be four structural competitive regimes, and the market features in each regime will practically remain the same as before.

On the other hand, we also find that the MOTS provider will be better off with an extended lifespan for the MOTS software. The benefits to the MOTS provider, as  $k$  increases, can be seen in two ways. First, the critical value  $t_1^k$  decreases in  $k$ , which suggests a shrinking Regime 1. It becomes more difficult, as  $k$  increases, for the SaaS provider to dominate the market. Second, in Regimes 2 and 3, when  $k$  increases, the MOTS provider can charge a higher price as ( $P_m^{*k}$  increases in  $k$ ) and meanwhile get a larger market share as ( $d^{*k}$  decreases in  $k$ ). As a result, the MOTS provider's profit is higher.

To demonstrate the impact of extending the economically useful lifespan of the MOTS package on the two providers' competitive positions, in the following Figures 5(a) and 5(b), we use  $k$  as the  $x$ -axis and draw each provider's price, market share, and profit as a function of  $k$ . We show that the MOTS provider is better off as  $k$  increases, gaining a higher price, larger market share, and higher profit (Figure 5(a)); whereas



Figure 5(a) (Color online) MOTS Price, Market Share, and Profit

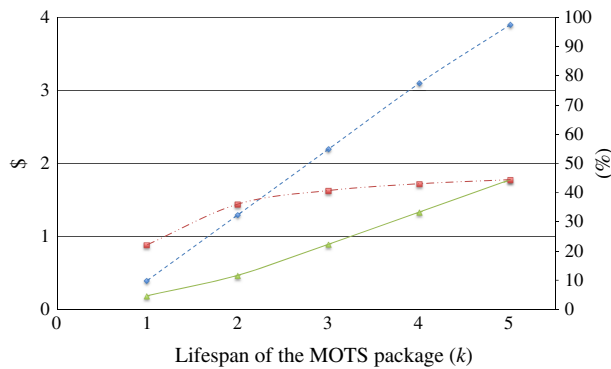
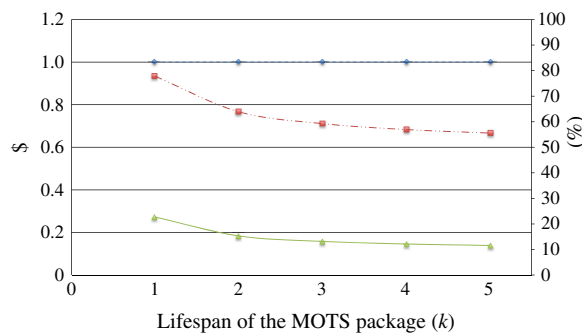


Figure 5(b) (Color online) SaaS Price, Market Share, and Profit



Note. The parameter values are:  $C_{\text{cust}} = 1$ ,  $c = 0.2$ ,  $c_m = 0.3$ ,  $c_s = 0.1$ ,  $u = 3$ ,  $u_r = 1$ ,  $t = 1.3$ ,  $\theta = 0.001$ .

the SaaS provider loses market share and generates a lower profit in each stage (Figure 5(b)).<sup>20</sup> Proposition 6 summarizes our findings.

**PROPOSITION 6 (THE SOFTWARE-LIFESPAN-IMPACT PROPOSITION).** *Although a longer software lifespan gives the MOTS provider an increased competitive advantage, it will not change the overall competitive pattern. As  $t$  decreases, the market will still demonstrate the following transition: MOTS Dominance  $\rightarrow$  Segmented Market  $\rightarrow$  Competitive Market  $\rightarrow$  SaaS Dominance.*

Our  $k$ -stage model is static without considering possible changes that might happen over time. Our results are likely to favor the MOTS provider: They provide an upper bound on the advantage that a long software lifespan could bring. The reason is as follows. Software obsolescence is not modeled in the  $k$ -stage setting. Over time, MOTS software becomes less valuable due to technological and economic obsolescence (Mehra et al. 2014); but the SaaS product will not lose value due to obsolescence. Because of its use-based pricing model (Choudhary 2007), the SaaS provider continually improves its offering to provide access to the most current features as soon as they are available. Taking these factors into consideration, it is possible to revise the  $k$ -stage model such that users' value

from the MOTS software ( $u$ ) decreases in  $k$ , which will weaken the MOTS provider's relative advantage. Also, we do not consider MOTS users' upgrading costs as  $k$  increases. Mehra et al. (2014) showed that users of in-house packaged software typically made great efforts to upgrade and replace hardware and software over time, and that providers tend to charge periodic fees for support or upgrades. These considerations can further diminish any long-term advantage of the MOTS solution.

## 5.2. Competition in a Dynamic Setting: The Changing Lack-of-Fit Costs

So far, we have analyzed the competition with static lack-of-fit costs and shown their importance to the outcome. In practice, these costs can change over time for a variety of reasons, such as the users' IT infrastructure, hardware or business-related changes, technological improvements or regulatory changes. For example, if the SaaS uses a browser interface that relies on nonstandard aspects of IE7, but business circumstances faced by users drive demand for the latest Chrome, the lack-of-fit costs may increase. If the SaaS continuously invests in improving its system-integration features and building a uniform platform to ease across-application collaboration,<sup>21</sup> the lack-of-fit costs may decrease.

Here, we extend our model to capture the essence of competition with possible changes in  $t$ . We consider a two-stage model. At the beginning of the first stage, providers set their prices ( $P_m, p_s$ ). The SaaS application imposes lack-of-fit costs  $t$  on users in the first stage. Users form expectations for any change in lack-of-fit costs: They may expect  $t$  to increase if they anticipate changes in demand or hardware upgrades, or to decrease if they anticipate technological advances that favor the shared-IT business model. Such a change is realized in the second stage, at which point users will consider whether to switch providers. We study the two providers' *forward-looking pricing strategy*, i.e., whether and how they will incorporate expectations for future change in  $t$  into their prices. We make the simplifying assumption that users and providers weight utilities and profits from the two stages equally. In the following analysis, we focus on the scenario in which the providers coexist in the market. Scenarios in which one provider fails are of little interest in this context.

<sup>21</sup> The AppExchange platform by Salesforce.com, an online marketplace for on-demand business software, is an example. It allows many SaaS providers to integrate their applications and therefore promises users a seamless extension of their existing systems (Cowley 2005, Kuchinskas 2006).

<sup>20</sup> The SaaS provider's  $k$ -stage total profit is still increasing in  $k$ , since its charges are based on use.

Figure 6(a) Competition with Increased Lack-of-Fit Costs

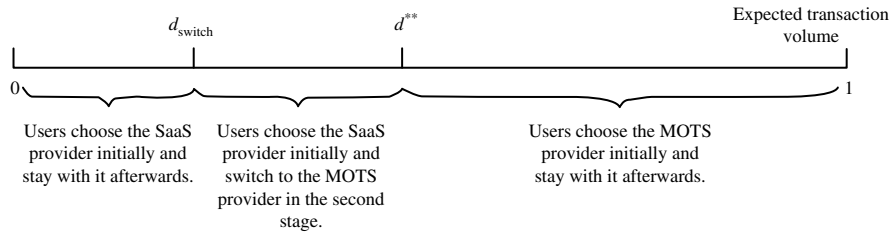
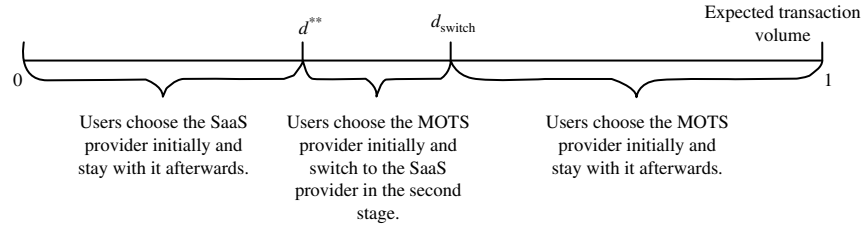


Figure 6(b) Competition with Decreased Lack-of-Fit Costs



**5.2.1. The Competition with Increased Lack-of-Fit Costs.** When  $t$  increases, the competitive environment will favor the MOTS provider. Figure 6(a) depicts such a situation when  $t$  is expected to increase to a higher-level  $t_H$  in the second stage. As the figure shows, in the first stage, with current lack-of-fit costs  $t$ , users in  $[0, d^{**}]$  choose the SaaS solution, and users in  $[d^{**}, 1]$  choose the MOTS alternative. Therefore, the user at  $d^{**}$  is the indifferent user between the two options, as it perceives the same level of total expected utility (from the two stages). In the second stage, with the increased lack-of-fit costs  $t_H$ , existing SaaS users in  $[d_{\text{switch}}, d^{**}]$  switch to the MOTS offering to get customized software with a better fit.<sup>22</sup> The user  $d_{\text{switch}}$  is the marginal switcher who gets the same utility from staying with the original choice and switching, *given* that it chose the SaaS product initially. We solve this problem using backward induction.

First, we consider the second stage. The marginal switcher  $d_{\text{switch}}$  is given by

$$\begin{aligned} Eu_{\text{SaaS}} &= (u - p_s - t_H)d_{\text{switch}} = Eu_{\text{MOTS}} \\ &= (u - c_m - c)d_{\text{switch}} - P_m - C_{\text{cust}} - \frac{c(\Delta u - c)\theta}{\Delta u}. \end{aligned}$$

Next, back to the first stage, the indifferent user  $d^{**}$  is given by

$$\begin{aligned} &\{(u - p_s - t)d^{**}\} \\ &+ \left\{ (u - c - c_m)d^{**} - P_m - C_{\text{cust}} - \frac{c(\Delta u - c)\theta}{\Delta u} \right\} \end{aligned}$$

<sup>22</sup> In some circumstances, existing SaaS users may incur exit costs when they stop using it. Such exit costs are mainly for moving data from the SaaS cloud center. Our model could be extended to include the exit costs for SaaS users. Also, we show that our major findings on providers' pricing strategies, i.e., Propositions 7 and 8, remain valid. Because of space limitations, we do not provide detailed analyses for such a model extension, but they are available on request.

$$\begin{aligned} &= \left\{ (u - c - c_m)d^{**} - P_m - C_{\text{cust}} - \frac{c(\Delta u - c)\theta}{\Delta u} \right\} \\ &+ \left\{ (u - c - c_m)d^{**} - \frac{c(\Delta u - c)\theta}{\Delta u} \right\} \\ \Rightarrow d_{\text{switch}} &= \frac{P_m + C_{\text{cust}}}{p_s + t_H - c - c_m} + \frac{c(\Delta u - c)\theta}{\Delta u(p_s + t_H - c - c_m)}. \quad (7) \end{aligned}$$

The left-hand side is the user's total expected utility if it chooses SaaS in the first stage and then switches to the MOTS offering in the second stage; the right-hand side is its total expected utility if the user chooses MOTS software in the first stage and uses it afterwards. We get

$$d^{**} = \frac{c(\Delta u - c)\theta}{\Delta u(p_s + t - c_m - c)}. \quad (8)$$

The MOTS provider gets  $[d^{**}, 1]$  users in the first stage, and  $[d_{\text{switch}}, d^{**}]$  users in the second stage; the SaaS provider serves  $[0, d^{**}]$  users in the first stage and  $[0, d_{\text{switch}}]$  users in the second stage. Their optimization problems can be formulated as

$$\begin{aligned} \max_{P_m} \Pi_{\text{MOTS}} &= P_m[1 - d_{\text{switch}}]; \\ \max_{p_s} \Pi_{\text{SaaS}} &= (p_s - c_s) \left[ \int_0^{d^{**}} x dx + \int_0^{d_{\text{switch}}} x dx \right]. \quad (9) \end{aligned}$$

**5.2.2. The Competition with Decreased Lack-of-Fit Costs.** When  $t$  decreases, the competitive environment will favor the SaaS provider. Figure 6(b) depicts such a situation when  $t$  decreases to a lower level  $t_L$ . In the first stage, with lack-of-fit costs  $t$ , users in  $[0, d^{**}]$  choose the SaaS, and users in  $[d^{**}, 1]$  choose the MOTS. In the second stage, with the decreased  $t_L$ , existing MOTS users in  $[d^{**}, d_{\text{switch}}]$  switch to the SaaS offering.

We again solve the problem using backward induction. The critical locations for the marginal switcher  $d_{\text{switch}}$  and indifferent user  $d^{**}$  are obtained in a similar way. Here we omit the detailed math derivations.

$$d_{\text{switch}} = \frac{c(\Delta u - c)\theta}{\Delta u(p_s + t_L - c - c_m)}; \quad (10)$$

$$d^{**} = \frac{P_m + C_{\text{cust}}}{p_s + t - c - c_m} + \frac{c(\Delta u - c)\theta}{\Delta u(p_s + t - c - c_m)}. \quad (11)$$

Accordingly, the two providers' optimization problems can be formulated as

$$\begin{aligned} \max_{P_m} \Pi_{\text{MOTS}} &= P_m[1 - d^{**}]; \\ \max_{P_s} \Pi_{\text{SaaS}} &= (p_s - c_s) \left[ \int_0^{d^{**}} x dx + \int_0^{d_{\text{switch}}} x dx \right]. \end{aligned} \quad (12)$$

**5.2.3. Pricing Strategy with Dynamic Lack-of-Fit Costs.** To see how the providers' prices change with a dynamic value of  $t$ , we take the static case with the invariant  $t$  as the benchmark, which has been analyzed in §4. We denote the equilibrium prices in the benchmark case as  $\{P_m^*(t), p_s^*(t)\}$ . Then,  $P_m^*(t \rightarrow t_H)$  and  $p_s^*(t \rightarrow t_H)$  are the equilibrium prices charged by the MOTS and SaaS providers when lack-of-fit costs are expected to increase to  $t_H$ , and  $P_m^*(t \rightarrow t_L)$  and  $p_s^*(t \rightarrow t_L)$  are the equilibrium prices when lack-of-fit costs are expected to decrease to  $t_L$ . Lemma 3 establishes the price changes.

**LEMMA 3.** *When lack-of-fit costs are expected to increase, both providers raise prices:  $P_m^*(t \rightarrow t_H) > P_m^*(t)$ , and  $p_s^*(t \rightarrow t_H) > p_s^*(t)$ . When lack-of-fit costs are expected to decrease, both providers reduce prices:  $P_m^*(t \rightarrow t_L) < P_m^*(t)$ , and  $p_s^*(t \rightarrow t_L) < p_s^*(t)$ .*

Lemma 3 shows that when lack-of-fit costs are expected to increase, both providers' prices will be higher than those in the benchmark; when lack-of-fit costs are expected to decrease, both providers' prices will be lower than those in the benchmark. These results are intriguing, as they reveal the two providers' totally different pricing strategies when facing changes in market conditions. We summarize their pricing strategies in Propositions 7 and 8, respectively.

**PROPOSITION 7 (THE SaaS FORWARD-LOOKING PRICING PROPOSITION).** *The SaaS provider always uses a forward-looking pricing strategy: When lack-of-fit costs are expected to decrease (increase) in the future, the SaaS should reduce (increase) its current price.*

A forward-looking pricing strategy means that when the provider sets the current price, it will incorporate expectations of future market conditions. Proposition 7 states that the SaaS provider will always do so, no matter what it expects as to the future value of  $t$ . An anticipated decrease in lack-of-fit costs suggests a higher value for SaaS future offerings. In this case, we find that the SaaS provider's response is

to reduce its price. The increased value and reduced price together put the SaaS provider in a much better competitive position, enabling it to compete aggressively with the MOTS provider for profitable high-volume users. By contrast, an expected increase in lack-of-fit costs suggests a lower value for SaaS future offerings. In this case, the SaaS provider should nevertheless increase its price and stop competing for high-volume users with the MOTS provider; instead, it should concentrate on exploiting low-volume users that cannot afford the MOTS solution. Such a strategy is consistent with what we have found in the static competition setting: The SaaS provider uses the higher (future) value, lower price pricing strategy.

**PROPOSITION 8 (THE MOTS'S FORWARD-LOOKING PRICING PROPOSITION).** *The MOTS provider uses a forward-looking pricing strategy only when lack-of-fit costs are expected to increase. When such costs are expected to decrease, it will ignore this expectation and use the same pricing strategy as in the benchmark with invariant lack-of-fit costs.*

We find that the MOTS provider, unlike the SaaS provider, will only respond to positive expectations: When  $t$  is expected to increase to  $t_H$ , which will favor the MOTS offering in the competition, the provider will price its product based on the future value  $t_H$  (see the proof of Lemma 3 in the appendix). It will not, however, use a forward-looking pricing strategy when  $t$  is expected to decrease. A decrease in  $t$  will favor the SaaS solution in the competition. In this circumstance, the MOTS provider will choose to ignore this negative expectation. Its pricing strategy (i.e., the best response function used to set the price) is the same as that in the benchmark (see the proof of Lemma 3 in the appendix). It does not take the future value  $t_L$  into consideration.

## 6. Discussion: Recommendations for Competitive Strategies

The real-world competition between SaaS and MOTS providers is much more complicated than our theoretical model suggests. Nevertheless, our model captures a number of interesting and important features, including the lack-of-fit costs of SaaS offerings, the capacity-hedging challenge faced by MOTS users, and the use-based versus one-time-fee pricing difference between the two. The model thus enables us to predict the possible outcome of competition in the software market, and to draw insightful conclusions about providers' competitive strategies. Next, we offer three recommendations to managers, based on our analytical results:

- *Recommendation 1. To succeed and prosper, SaaS providers must reduce the lack-of-fit costs of their applications.*

A key finding of our study is the important impact of lack-of-fit costs  $t$ . Proposition 5 depicts a complete market evolution when lack-of-fit costs could decrease over time, from “SaaS fails” to “SaaS dominates.” To succeed in the competitive marketplace, SaaS providers must offer software applications with a reasonably low value of  $t$ . One direct implication is that not all software applications can be successfully delivered via SaaS. Applications that are suited to the SaaS model are those easily compatible with other systems and programs, with relatively standard functionalities and features, not specific to the enterprises’ core business, and with low interoperability costs. That is why we have seen the fast growth of SaaS in marketplaces such as email, office productivity, accounting, billing, and human resource management software, but less so for complex applications, such as Enterprise Resource Planning (ERP) or Electronic Medical Record (EMR) systems. Most large-scale ERP or EMR systems still run in enterprise data centers and have been customized at great expense to fit each company’s specific needs (Norton and Boulton 2014). As our results suggest, high lack-of-fit costs associated with these types of complex systems make the SaaS model unattractive.

The main advantage of multitenancy is simplicity in configuration management and support over time. Keeping one version on site for many users cuts the software maintenance costs dramatically, compared with keeping multiple copies with custom-variations of the same software. This multitenancy model of SaaS inherently involves lack-of-fit costs for the users. Recognizing that SaaS is not simply a new distribution channel for existing software, and that it requires a fundamentally new set of products and technology architectures, SaaS providers should write software applications in a proper modular structure, with loosely coupled interfaces with other applications. In addition, different enterprise users are likely to have different lack-of-fit costs due to their heterogeneous IT capabilities. SaaS providers therefore could offer multiple online versions to target enterprises of different sizes and complexities. They may even consider offering a certain degree of customization to large clients through some in-depth strategic collaboration.

Industry-wide adoption of software standards and protocols, once achieved, would also help reduce the interoperability costs across different applications and therefore positively affect SaaS. Some efforts have been made in this regard. For instance, Microsoft, Oracle, and Sun have joined forces with IBM and Hewlett-Packard to facilitate technical standards that will govern how commercial software should be written. In addition, many big SaaS providers have started to launch platforms that will allow different types of software to run together, which they believe will also help reduce software integration costs (Schwartz 2007).

Because these strategic moves could increase SaaS competitiveness, we expect the SaaS business model to continue and prosper in the future.

- *Recommendation 2. SaaS priority is to pass on the benefits of economies of scale to users through low prices. Specifically, the SaaS provider should use the “higher value and higher cost efficiency, but lower price” strategy.*

The SaaS provider gains economies of scale through serving a large number of clients based on a shared IT infrastructure. Experience and knowledge from serving one client can be used to enhance efficiency and create value in serving other clients. Our results show that to win the competition with the MOTS provider, it is important that the SaaS provider share the economic gains with its clients by steadily reducing prices. We have observed a wave of price reductions for SaaS products. For example, in March 2012, Microsoft announced the third price reduction for its Office 365 cloud-based software portfolio (Sullivan et al. 2012); Amazon Web Services has continued its price-cutting strategy, and as of July 2013, had reduced its prices 37 times (Babcock 2013). Both Microsoft and Amazon attribute their price reductions to economies of scale that have been achieved due to a large customer base. These price reductions have not been solely motivated by good will; our research demonstrates that it is in the SaaS providers’ interest to charge less when their value and cost efficiency are enhanced. When the SaaS provider can deliver higher value (i.e., a higher  $u - t$ ) and operate with higher cost efficiency (i.e., a larger  $c_m - c_s$ ), the best action is to reduce its price. Such a strategy will put the SaaS provider at an advantage when competing for high-volume users.

The proposed “higher value and higher cost efficiency, but lower price” strategy works particularly well for the SaaS providers with per-transaction charges. Because of the nature of transaction-based pricing, SaaS providers will find that high-volume users are much more profitable than low-volume users. Thus, combining low price and high quality is an effective tool for an SaaS provider to efficiently acquire large firms instead of targeting only small and medium enterprises. However, this may not be the best strategy for the SaaS providers who base their charges on a monthly subscription fee, ignoring the transaction volume implications.

- *Recommendation 3. In the competition with SaaS, the MOTS provider should focus on enhancing the value of its enterprise software, rather than cutting its price.*

We have seen that due to Web technology improvements, the adoption of software standards and protocols, and the SaaS providers’ continual efforts to create a uniform software platform, the value of SaaS is likely to increase over time. Facing such a competitive change in the environment, we suggest that the MOTS provider focus on enhancing the value of its



software instead of competing aggressively with low prices. The cost of buying and using enterprise software has remained stable in past years. In-house enterprise software is not getting cheaper, even in the face of price-cutting competition from SaaS providers. As Proposition 8 states, even when the MOTS provider expects its competitor to gain an advantage in the future, the right strategy is to ignore this expectation and continue to price its product as if it were in a static setting. For the MOTS provider, price is not an effective competitive tool; quality is the key factor. As long as the value of MOTS offerings is sufficiently higher than that of SaaS, the MOTS provider can maintain monopolistic power in the targeted market segment. Therefore, developing software features and functionalities and delivering higher value are critical for MOTS providers.

Finally, we want to point out that Recommendation 3 is mainly for the one-product MOTS provider. There is a possibility that the MOTS provider can offer different versions of the enterprise software. For example, in addition to the version targeting large clients, the provider could offer another version with fewer features (and a lower price) to compete with the SaaS provider in the small and medium enterprises (SME) segment. Versioning has long been recognized as a way to segment the market and gain higher profit. It would not be surprising for the MOTS provider to use a versioning strategy to enhance its competitiveness.

## 7. Conclusion

The fundamental differences in the business models of SaaS and MOTS providers determine their distinct competitive strategies. The top priority for SaaS providers is to invest in reducing their clients' lack-of-fit costs. Continuous technology improvements, increasing adoption of software standards, and efforts to create a uniform platform for different applications cause us to believe that SaaS will eventually attain a solid position in the market. In light of these developments, SaaS providers must continue to reduce prices so that the economic gains will be shared with their clients, in turn making them more competitive. The situation is different, however, for MOTS providers; for them, lowering prices to make their software cheaper is not a good strategy. Rather, they should actively invest in developing full-feature software and enhancing its perceived value.

The key factor that drives these distinct strategies is the providers' different pricing schemes. As it uses use-based pricing, the SaaS provider's profit is linear in the total transaction volume; high-volume users are much more profitable than low-volume users. The SaaS provider thus has a strong incentive to compete aggressively in price to get those high-volume users.

The MOTS provider's profit, however, is concave in the expected transaction volume. Investing in ensuring superior software value is far more effective than following a price-cutting strategy.

Because the SaaS trend seems unstoppable in some segments, established software providers should also consider changing their business models to include MOTS and SaaS offerings. Some providers, for example, Oracle and IBM, have adopted such a hybrid strategy; they typically sell a sophisticated version of their software as a MOTS product while leasing a simplified version as a SaaS product. This hybrid model merits closer investigation: Who should adopt the hybrid model, and what types of software are best suited to it? Should a hybrid provider design its own MOTS and SaaS products so that they are capable of competing with other providers' offerings without cannibalizing themselves? Future research might also consider extending the current model to overcome its limitations. For example, we can study the competition outcomes and provider strategies when clients have heterogeneous lack-of-fit costs, or when the MOTS provider can offer different versions of software to satisfy different users. We expect to obtain deeper and richer implications from these extensions. Another future research direction is to study and compare different pricing schemes by the SaaS provider. For example, which one of our major findings and the recommended competitive strategies will still be valid in a subscription-fee setting? Which pricing method, a per-transaction charge, a fixed subscription fee, or hybrid pricing, performs best in competition with the MOTS provider? Exploring these questions will require the use of a multiple-stage model that highlights the essential differences between these pricing schemes and considers key factors such as the dynamic demand and lifespan of the software systems. In addition, as the SaaS market evolves and matures, we expect that use data will become available to support empirical research. We can use such data to analyze use patterns and solve the capacity-choice problem for SaaS providers, or to conduct a realized cost and benefit analysis for SaaS users and, in turn, make recommendations for the adoption of SaaS.

## Supplemental Material

Supplemental material to this paper is available at <http://dx.doi.org/10.1287/isre.2015.0571>.

## Acknowledgments

When developing this research, the authors visited and studied a large number of SaaS (cloud-computing) providers and users, including Amazon Web Services, IBM On-demand, Carestream Medical Imaging, Philips Medical Imaging (formerly Stentor), Salesforce.com, SingTel, ASG Software Solutions, UNIT4 Business Software, Vision Solutions, Singapore National Cloud Office, Institute of High

Performance Computing at Singapore, and Swiss Banking Services Corp. The authors also benefited from discussions at the Salesforce.com seminar given by Marc Benioff, Asia Cloud Computing Association Roundtable (2012 and 2013), Amazon Web Services Summit 2013, and Cloud Expo Asia (2013). The authors especially acknowledge the help and comments obtained from Professors Haim Mendelson and J. Michael Harrison, of Stanford University; Dr. Shane Owenby (Managing Director, Amazon Web Services); and Dr. Lee Hing Yan (Program Director, National Cloud Office of Singapore). The authors also thank Dr. Rob Kauffman for his many insightful editorial comments and suggestions on the recent draft of this manuscript. All errors and omissions remain the sole responsibility of the authors.

## Appendix. Proof of Propositions and Lemmas

### Proof for Deriving the Optimal in-House IT Capacity $Q_i^*$ for a MOTS User

User  $i$  must determine an optimal capacity level  $Q_i^*$  by solving

$$\begin{aligned} \max_{Q_i} & E[(u - c_m) \min\{D_i, Q_i\} \\ & + (u_r - c_m)(D_i - Q_i)^+ - P_m - C_{\text{cust}} - cQ_i] \\ & = \text{Prob}\{D_i \leq Q_i\} \cdot E[D_i/D_i \leq Q_i] \cdot (u - c_m) + \text{Prob}\{D_i > Q_i\} \\ & \cdot \{(E[D_i/D_i > Q_i] - Q_i) \cdot (u_r - c_m) + Q_i \cdot (u - c_m)\} \\ & - P_m - C_{\text{cust}} - cQ_i. \end{aligned}$$

Since  $d_i$  is uniformly distributed on  $[d_i - \theta, d_i + \theta]$ , we get

$$\begin{aligned} \text{Prob}\{D_i \leq Q_i\} &= \frac{Q_i - d_i + \theta}{2\theta}; \\ \text{Prob}\{D_i > Q_i\} &= \frac{d_i + \theta - Q_i}{2\theta}; \\ E[D_i/D_i \leq Q_i] &= \frac{\int_{d_i - \theta}^{Q_i} x dF(x)}{(Q_i - d_i + \theta)/(2\theta)} = \frac{1}{2}(Q_i + d_i + \theta); \quad \text{and} \\ E[D_i/D_i > Q_i] &= \frac{\int_{Q_i}^{d_i + \theta} x dF(x)}{(d_i + \theta - Q_i)/(2\theta)} = \frac{1}{2}(Q_i + \theta + d_i). \end{aligned}$$

Plug in these values; we have

$$\begin{aligned} \max_{Q_i} & E[(u - c_m) \min\{D_i, Q_i\} \\ & + (u_r - c_m)(D_i - Q_i)^+ - P_m - C_{\text{cust}} - cQ_i] \\ & = \frac{(u - c_m)}{4\theta} \{Q_i^2 - (d_i - \theta)^2 + 2Q_i[d_i + \theta - Q_i]\} \\ & + \frac{(u_r - c_m)}{4\theta} \{d_i + \theta - Q_i\}^2 - P_m - C_{\text{cust}} - cQ_i. \end{aligned}$$

Take the derivative of  $Q_i$ , which gives  $(u - c_m)\{d_i + \theta - Q_i\} + (u_r - c_m)\{Q_i - d_i - \theta\} - 2c\theta = 0$ .

The optimal  $Q_i^*$ , therefore, is given by  $Q_i^* = d_i + \theta(1 - 2c/\Delta u)$ .

### Proof of Lemma 1

First, note that when  $c_s < c_m$ , we always have  $t_1 > 0$ . As long as the SaaS provider can have relative cost efficiency in providing IT services, there is always a possibility that once the lack-of-fit costs are small enough ( $t < t_1$ ), the SaaS provider could dominate the market.

To see  $t_1 < t^*$ :  $t_1 < t^* \Leftrightarrow C_{\text{cust}} + c(\Delta u - c)\theta/\Delta u - (u - c - c_m) \leq 0 \Leftrightarrow 0 \leq P'_m$ . We know that  $P'_m \geq 0$  holds. If not, the MOTS provider is unable to serve as the monopoly in the pre-SaaS market.

To see  $t_2 > t^*$ :  $t_2 > t^* \Leftrightarrow u > c + c_m$ . Since  $u - c - c_m > 0$  to ensure the net value per transaction is positive in the MOTS option, we have  $t_2 > t^*$ .

### Proof of Propositions 2–4

The two providers' optimization problems are formulated in §4.2, with constraints in  $p_s$ ,  $P_m$ , and  $d^*$ , where  $d^*$  is given by Equation (6).

First, we show that when  $t < t_1$ ,  $(P_m^*, p_s^*) = (0, 2c_s + t - c_m - c)$  are the equilibrium prices. This is the constrained solution when  $P_m \geq 0$  and  $d^* \leq 1$  are binding. With  $p_s = 2c_s + t - c_m - c$ , the SaaS provider gains a positive profit and has no incentive to deviate because this price is obtained using its best response function (see Equation (13)). The MOTS provider has no incentive to deviate as well: A price increase will not attract any user, and a price decrease will not bring positive profit. We therefore conclude that this is the Nash equilibrium. In this case, the SaaS provider serves the whole market.

When  $t > t_2$ ,

$$(P_m^*, p_s^*) = \left( \frac{u - c - c_m}{2} - \frac{1}{2}C_{\text{cust}} - \frac{(\Delta u - c)c}{2\Delta u}\theta, c_s \right)$$

are the equilibrium prices. This is the constrained solution when  $c_s \leq p_s$  is binding. Both providers have no incentive to deviate. The MOTS provider is charging at the monopoly price and gains monopoly market share (it is easy to see that  $d^* = d'$  in this case) and therefore has no incentive to change its price. The SaaS provider is operating at the marginal cost and is unable to raise or reduce its price: A price increase will make it unattractive to any potential user, and a price reduction will give a negative profit. We therefore conclude that this is the Nash equilibrium. In this case, the SaaS provider is unable to operate profitably. We thus have proved Proposition 2.

Now we derive the equilibrium prices for  $t_1 \leq t \leq t_2$ . The first-order conditions (FOCs) for the MOTS and SaaS providers are, respectively

$$\begin{aligned} P_m^* &= \frac{(p_s^* + t - c - c_m)}{2} - \frac{1}{2}C_{\text{cust}} - \frac{(\Delta u - c)c}{2\Delta u}\theta, \\ \left( P_m^* + c_{\text{cust}} + \frac{c(\Delta u - c)\theta}{\Delta u} \right)^2 \left( 1 - \frac{2(p_s^* - c_s)}{p_s^* + t - c - c_m} \right) &= 0. \end{aligned} \quad (13)$$

When  $t^* \leq t \leq t_2$ , the equilibrium prices are

$$\left\{ P_m^* = \frac{(u - c - c_m)}{2} - \frac{1}{2}C_{\text{cust}} - \frac{(\Delta u - c)c}{2\Delta u}\theta; p_s^* = u - t \right\}.$$

This is the constrained solution when  $p_s^* \leq u - t$  is binding. Using Equation (6), the marginal user is

$$d^* = \frac{1}{2} + \frac{C_{\text{cust}}}{2(u - c_m - c)} + \frac{c(\Delta u - c)\theta}{2\Delta u(u - c_m - c)} = d',$$

which is the same as Equation (5) in the pre-SaaS market. The MOTS price is also the same as the monopoly price  $P'_m$ . The conclusion that the MOTS provider's profit does not change follows. In addition, at this SaaS price, all consumer

surpluses are extracted. We therefore have proved Proposition 3, i.e., that the two providers are serving the market in an entirely segmented way.

When  $t_1 < t \leq t^*$ , the equilibrium prices are  $\{P_m^* = (c_s + t - c - c_m) - \frac{1}{2}C_{\text{cust}} - ((\Delta u - c)/(\Delta u)\theta); p_s^* = 2c_s + t - c - c_m\}$ . This is the unconstrained solution. Using Equation (6), the marginal user is

$$d^* = \frac{1}{2} + \frac{C_{\text{cust}}}{4(c_s + t - c_m - c)} + \frac{c(\Delta u - c)\theta}{4\Delta u(c_s + t - c_m - c)} = d',$$

and so the MOTS provider's market share decreases compared to the pre-SaaS market. It is easy to see that the MOTS price also falls,  $P_m^* < P'_m$ . As a result, the MOTS provider is negatively affected by the competition. Next, we compare the consumer surplus change after the SaaS provider enters. Small users with low transaction volume, in  $[0, d']$ , who will be priced out of the market in the pre-SaaS scenario, can now access the software through the SaaS provider; at price  $p_s^* = 2c_s + t - c - c_m$ , they gain positive consumer surplus. Users in  $[d', d^*]$  purchase the MOTS software in the pre-SaaS market but now choose to use the SaaS product because the total costs of using SaaS are lower:  $E[u_{\text{MOTS}}(P'_m)] \leq E[u_{\text{MOTS}}(P_m^*)] \leq E[u_{\text{SaaS}}(P_s^*)]$ . They are better off. Finally, users in  $[d^*, 1]$  with a high volume of transactions choose the MOTS solution in both cases, but their payments are lower when the SaaS option exists because the MOTS price is reduced under competition,  $P_m^* < P'_m$ . Thus, the SaaS provider's entry into the market benefits these firms as well, although they are not its clients. Each user is better off, and the total consumer surplus is strictly improved. We have proved Proposition 4.

#### Proof of Lemma 2

The MOTS price,  $P_m^*$ , the SaaS price,  $p_s^*$ , and the marginal user's position,  $d^*$ , are all derived in the proof for Propositions 2–4. It is straightforward to see that

$$\frac{\partial P_m^*}{\partial \theta} = \frac{-c(\Delta u - c)}{2\Delta u} < 0, \quad \frac{\partial P_s^*}{\partial \theta} = 0, \quad \text{and} \quad \frac{\partial d^*}{\partial \theta} = \frac{c(\Delta u - c)}{4\Delta u(c_s + t - c - c_m)} > 0.$$

#### Proof of Lemma 3

Propositions 7 and 8 are proved together with Lemma 3.

We first analyze when lack-of-fit costs increase,  $t \rightarrow t_H$ . In this case, the two providers' decision-making problems are formulated in Equation (9), and the indifferent user  $d^{**}$  and marginal switcher  $d_{\text{switch}}$  are given by Equations (8) and (7), respectively. The FOCs, or best response functions (BRFs), are

$$P_m^* = \frac{p_s^* + t_H - c - c_m}{2} - \frac{1}{2}C_{\text{cost}} - \frac{(\Delta u - c)c}{2\Delta u}\theta$$

(for the MOTS provider), and

$$\begin{aligned} & \frac{c^2(\Delta u - c)^2\theta^2}{2\Delta u^2} \frac{-p_s^* + t - c - c_m + 2c_s}{(p_s^* + t - c - c_m)^3} \\ & + \left( P_m^* + C_{\text{cust}} + \frac{c(\Delta u - c)\theta}{\Delta u} \right)^2 \left( 1 - \frac{2(p_s^* - c_s)}{p_s^* + t_H - c - c_m} \right) = 0 \end{aligned}$$

(for the SaaS provider). The BRF for the MOTS provider is the same as in the static competition game with invariant lack-of-fit costs  $t_H$ . This proves Proposition 8 partially; when lack-of-fit costs are expected to increase, the MOTS provider will price based on the future value of  $t$ . The BRF for the SaaS provider considers current and future values of  $t$ , and so we have proved Proposition 7 partially; the SaaS provider uses forward-looking pricing when lack-of-fit costs are expected to increase. In addition, when we evaluate the FOC of the SaaS provider at  $p_s^*(t)$ , we find its value is positive. Therefore, we get  $p_s^*(t \rightarrow t_H) > p_s^*(t)$ , and  $P_m^*(t \rightarrow t_H) > P_m^*(t)$  follows (using the BRF of the MOTS provider).

Now we analyze the case wherein lack-of-fit costs decrease,  $t \rightarrow t_L$ . The two providers' decision-making problems are formulated in Equation (12), the indifferent user  $d^{**}$  is given by Equation (11), and the marginal switcher  $d_{\text{switch}}$  is given by Equation (10). The two BRFs are

$$P_m^* = \frac{p_s^* + t - c - c_m}{2} - \frac{1}{2}C_{\text{cost}} - \frac{(\Delta u - c)c}{2\Delta u}\theta$$

(for the MOTS provider), and

$$\begin{aligned} & \frac{c^2(\Delta u - c)^2\theta^2}{2\Delta u^2} \frac{-p_s^* + t_L - c - c_m + 2c_s}{(p_s^* + t_L - c - c_m)^3} \\ & + \left( P_m^* + C_{\text{cust}} + \frac{c(\Delta u - c)\theta}{\Delta u} \right)^2 \left( 1 - \frac{2(p_s^* - c_s)}{p_s^* + t - c - c_m} \right) = 0 \end{aligned}$$

(for the SaaS provider). The BRF for the MOTS provider is the same as in the one-stage competition game with invariant lack-of-fit costs  $t$ . This proves Proposition 8 partially; when lack-of-fit costs are expected to decrease, the MOTS provider will ignore the future expectation as to  $t$ . The BRF for the SaaS provider considers both current and future values of  $t$ , and so we have proved Proposition 7 partially; the SaaS provider employs forward-looking pricing when lack-of-fit costs are expected to decrease. In addition, when we evaluate the FOC of the SaaS provider at  $p_s^*(t)$ , we find its value is negative. Therefore, we get  $p_s^*(t \rightarrow t_L) < p_s^*(t)$ , and  $P_m^*(t \rightarrow t_L) < P_m^*(t)$  follows (using the BRF of the MOTS provider).

#### References

- August T, Niculescu MF, Shin H (2014) Cloud implication on software network structure and security risks. *Inform. Systems Res.* 25(3):489–510.
- Babcock C (2013) Amazon cuts cloud prices again. *InformationWeek* (July 10). <http://www.informationweek.com/cloud/infrastructure-as-a-service/amazon-cuts-cloud-prices-again/d/d-id/1110697?>
- Balasubramanian S, Bhattacharya S, Krishnan V (2011) Pricing information goods: A strategic analysis of the selling and pay-per-use mechanisms. Working paper, University of North Carolina, INSEAD, and University of California, San Diego.
- Brill J, Chiu L, Wilcox J (2010) What is the best payment model for SaaS and why: Pay as you go or subscriptions? Accessed May 2015, <http://www.quora.com/What-is-the-better-payment-model-for-SaaS-and-why-pay-as-you-go-or-subscriptions>.
- Choudhary V (2007) Comparison of software quality under perpetual licensing and software as a service. *J. Management Inform. Systems* 24(2):141–165.
- Choudhary V, Tomak K, Chaturvedi A (1998) Economics benefits of software renting. *J. Organ. Comput. Electronic Commerce* 8(4): 277–305.



- Cowley S (2005) Salesforce.com makes platform move with AppExchange. *InfoWorld* (September 9). <http://www.infoworld.com/article/2671745/application-development/salesforce-com-makes-platform-move-with-appexchange.html>.
- Dewan S, Mendelson H (1990) User delay costs and internal pricing for a service facility. *Management Sci.* 36(12): 5102–5117.
- Dubey A, Wagle D (2007) Delivering software as a service. The McKinsey quarterly Web exclusive. Accessed May 2015, [http://mckinsey.de/downloads/publikation/mck\\_on\\_bt/2007/mobt\\_12\\_Delivering\\_Software\\_as\\_a\\_Service.pdf](http://mckinsey.de/downloads/publikation/mck_on_bt/2007/mobt_12_Delivering_Software_as_a_Service.pdf).
- Dunn D (2006) IBM touts missing link to utility computing and shared IT resources. *Information Week* (May 31). <http://www.informationweek.com/ibm-touts-missing-link-to-utility-computing-and-shared-it-resources/d/d-id/1043777?>
- Fan M, Kumar S, Whinston AB (2009) Short-term and long-term competition between providers of shrink-wrap software and software as a service. *Eur. J. Oper. Res.* 196(2):661–671.
- Fishburn P, Odlyzko A (1999) Competitive pricing of information goods: Subscription pricing versus pay-per-use. *Econom. Theory* 13(2):447–470.
- Fishburn P, Odlyzko A, Siders R (1997) Fixed fee versus unit pricing for information goods: Competition, equilibria, and price wars. *First Monday* 2(7):1–18.
- Glanz JP (2012) Pollution and the Internet. *New York Times* (September 22). [http://www.nytimes.com/2012/09/23/technology/data-centers-waste-vast-amounts-of-energy-belying-industry-image.html?\\_r=0](http://www.nytimes.com/2012/09/23/technology/data-centers-waste-vast-amounts-of-energy-belying-industry-image.html?_r=0).
- Guo P (2009) A survey of software as a service delivery paradigm. TKK Technical reports in computer science and engineering, Helsinki University of Technology, Helsinki, Finland. <http://www.cse.hut.fi/en/publications/B/5/>.
- Gurnani H, Karlapalem K (2001) Optimal pricing strategies for Internet-based software dissemination. *J. Oper. Res. Soc.* 51(1):64–70.
- Huang KW, Sundararajan A (2011) Pricing digital goods: Discontinuous costs and shared infrastructure. *Inform. Systems Res.* 22(4):721–738.
- InformationWeek (2012) Software as a service survey—2007. *InformationWeek* (January 13). <http://www.informationweek.com/whitepaper/Software/Hosted-Software-Applications/software-as-a-service-survey-2007-wp1213891754874/75052>.
- Katzan HJ (2010) On an ontological view of cloud computing. *J. Service Sci.* 3(1):1–6.
- Koehler P, Anandasivam A, Ma D, Weinhardt C (2010) Customer heterogeneity and tariff biases in cloud computing. *Proc. 31st Internat. Conf. Inform. Systems, St. Louis*.
- Kuchinskias S (2006) Salesforce finally ships AppExchange. *Ecommerce* (January 17). <http://www.internetnews.com/ec-news/article.php/3578066/Salesforce+Finally+Ships+Appexchange.htm>.
- Ma D, Seidmann A (2008) The pricing strategy analysis for the software-as-a-service business model. *Proc. 5th Internat. Conf. Econom. Grids, Clouds, Systems and Services, Palmas de Gran Canaria, Spain*, 103–112.
- Mahowald R, Sullivan C (2012) Worldwide SaaS and cloud software 2012–2016 forecast and 2011 vendor shares. IDC Online Rep. (August). Accessed November 2014, <http://www.idc.com/getdoc.jsp?containerId=236184>.
- Mehra A, Seidmann A, Mojmunder P (2014) Product life-cycle management of packaged software. *Production Oper. Management* 23(3):366–378.
- Mell P, Grance T (2009) The NIST definition of cloud computing. National Institute of Standards and Technology, Information Technology Laboratory. Accessed May 2015, <http://csrc.nist.gov/groups/SNS/cloud-computing/index.html>.
- Muller J, Kruger J, Enderlein S, Marco H, Zeier A (2009) Customizing enterprise software as a service applications: Back-end extension in a multi-tenancy environment. Filipe J, Cordeiro J, eds. *Enterprise Information Systems*, Vol. 24 (Springer-Verlag, Berlin Heidelberg), 66–77.
- Norton S, Boulton C (2014) Why big companies delay using the cloud for some applications. *Wall Street Journal* (July 16). <http://www.wsj.com/articles/why-big-companies-delay-using-the-cloud-for-some-applications-1405555098>.
- Outsource2India (2011) Software as a service (SaaS). Accessed November 2014, <http://www.outsource2india.com/software/articles/software-as-a-service.asp>.
- Retana G, Forman C, Narasimhan S, Niculescu MF, Wu DJ (2012) Technical support and IT capacity demand: Evidence from the cloud. *Proc. 33rd Internat. Conf. Inform. Systems, Orlando, FL*.
- Schwartz E (2007) Salesforce.com launches SaaS platform. *InfoWorld* (April 23). <http://www.infoworld.com/article/2663376/applications/salesforce-com-launches-saas-platform.html>.
- Seidmann A, Zhang J (2010) Perpetual versus subscription licensing under quality uncertainty and network externality effects. *J. Management Inform. Systems* 27(1):39–68.
- Singh C, Shelor R, Jiang J, Klein G (2004) Rental software valuation in IT investment decisions. *Decision Support Systems* 38(1): 115–130.
- Sullivan C, Konary A, Webster M, Mahowald R (2012) The price is right? Microsoft reduces pricing on Office 365 and Azure cloud services. IDC Link (March 30). [http://webcache.googleusercontent.com/search?q=cache:ky\\_Rf6DBYHwJ:download.microsoft.com/download/9/9/D/99D5C2DF-06AA-48D7-94E1-2D3C1DA4E709/MSFT\\_Reduces\\_O365-Windows\\_Azure\\_Pricing.pdf+&cd=1&hl=en&ct=clnk&gl=sg&client=safari](http://webcache.googleusercontent.com/search?q=cache:ky_Rf6DBYHwJ:download.microsoft.com/download/9/9/D/99D5C2DF-06AA-48D7-94E1-2D3C1DA4E709/MSFT_Reduces_O365-Windows_Azure_Pricing.pdf+&cd=1&hl=en&ct=clnk&gl=sg&client=safari).
- Sundararajan A (2004) Nonlinear pricing of information goods. *Management Sci.* 50(12):1660–1673.
- Susarla A, Barua A, Whinston AB (2003) Understanding the service component of application service provision: An empirical analysis of satisfaction with ASP services. *MIS Quart.* 27(1): 91–123.
- Susarla A, Barua A, Whinston AB (2009) A transaction cost perspective of software as a service business model. *J. Management Inform. Systems* 26(2):205–240.
- Varian H (2000) Buying, selling, and renting information goods. *J. Indust. Econom.* 47(4):473–488.
- Zimmerman J (1979) The costs and benefits of cost allocations. *Accounting Rev.* 54(3):504–521.
- Zwim E (2012) Is the cloud really cheaper? *ComputerWorld* (February 1). <http://www.computerworld.com/article/2500880/cloud-computing/is-the-cloud-really-cheaper.html>.