BANK OF ENGLAND

Working Paper No. 400
Liquidity-saving mechanisms and bank behaviour

Marco Galbiati and Kimmo Soramäki

July 2010

BANK OF ENGLAND

Working Paper No. 400

Liquidity-saving mechanisms and bank behaviour

Marco Galbiati[1] and Kimmo Soramäki[2]

Abstract

This paper investigates the effect of liquidity-saving mechanisms (LSMs) in interbank payment systems. We model a stylised two-stream payment system where banks choose (a) how much liquidity to post and (b) which payments to route into each of two 'streams': the RTGS stream, and an LSM stream. Looking at equilibrium choices we find that, when liquidity is expensive, the two-stream system is more efficient than the vanilla RTGS system without an LSM. This is because the LSM achieves better co-ordination of payments, without introducing settlement risk. However, the two-stream system still only achieves a second-best in terms of efficiency: in many cases, a central planner could further decrease system-wide costs by imposing higher liquidity holdings, and without using the LSM at all. Hence, the appeal of the LSM resides in its ability to ease (but not completely solve) strategic inefficiencies stemming from externalities and free-riding. Second, 'bad' equilibria too are theoretically possible in the two-stream system. In these equilibria banks post large amounts of liquidity and at the same time overuse the LSM. The existence of such equilibria suggests that some co-ordination device may be needed to reap the full benefits of an LSM. In all cases, these results are valid for this particular model of an RTGS payment system and the particular LSM.

Key words: Payment system, RTGS, liquidity-saving mechanism.

JEL classification: C7.

(1) Bank of England. Email: marco.galbiati@bankofengland.co.uk
(2) Helsinki University of Technology. Email: kimmo@soramaki.net

The views expressed in this paper are those of the authors, and not necessarily those of the Bank of England. The authors thank participants in: the 6th Bank of Finland Simulator Seminar (Helsinki, 25–27 August 2008); the 1st ABM-BaF conference (Torino, 9–11 February 2009); and the 35th Annual EEA Conference (New York, 27 February-1 March 2009). The authors are also indebted to Marius Jurgilas, Ben Norman, Tomohiro Ota and other colleagues at the Bank of England for useful comments and encouragement. Kimmo Soramäki gratefully acknowledges the support of OP-Pohjola-Ryhmän tutkimussäätiö. This paper was finalised on 11 May 2010.

The Bank of England's working paper series is externally refereed.

Information on the Bank's working paper series can be found at
www.bankofengland.co.uk/publications/workingpapers/index.htm

Publications Group, Bank of England, Threadneedle Street, London, EC2R 8AH
Telephone +44 (0)20 7601 4030 Fax +44 (0)20 7601 3298 email mapublications@bankofengland.co.uk

ISSN 1749-9135 (on-line)

Contents

Summary

Interbank payment systems form the backbone of financial architecture; their safety and efficiency are of great importance to the whole economy. Most large-value interbank payment systems work in RTGS (real-time gross settlement) mode: each payment must be settled individually by transferring the corresponding value from payer to payee in central bank money. As such, all settlement risk is eliminated.

But an RTGS structure may incentivise free-riding. A bank may find it convenient to delay its outgoing payments (placing it in an internal queue) and wait for incoming funds, in order to avoid the burden of acquiring expensive liquidity in the first place. As banks fail to 'internalise' the systemic benefits of acquiring liquidity, RTGS systems may suffer from inefficient liquidity underprovision.

Inefficiencies may also emerge for a second reason. Payments queued internally in segregated queues are kept out of the settlement process and do not contribute to 'recycling' liquidity. A tempting idea is therefore to pool these pending payments together in a central processor, which could look for cycles of offsetting payments and settle them as soon as they appear. This would save liquidity, and might also reduce settlement time: payments could settle as soon as it is technically possible to do so. Segregated queues may instead hold each other up for a long time, not 'paying to each other' because none is doing so.

Such central queues are called 'liquidity-saving mechanisms' (LSMs). There are a number of studies on plain RTGS systems, but only a few on RTGS systems augmented with LSMs. Our work contributes to this line of research.

We first model a benchmark system, ie a plain RTGS system where each bank decides: (i) the amount of liquidity to use; (ii) which payments to delay in an internal queue (payments are made as banks randomly receive payment orders, which need be executed with different 'urgency'). The benchmark model is then compared to an RTGS–plus–LSM system, where banks decide: (i) the amount of liquidity to use in RTGS as above; and (ii) which payments to submit to the LSM stream, where payments are settled as soon as offsetting cycles form.

A necessary caveat is that we consider a specific LSM, comparing it to a specific model of internal queues. Other LSMs, perhaps associated with different settlement rules, may yield different outcomes. For example, one could think of a system where *all* payments (even those sent to the RTGS stream) are first passed through the LSM. Then, if LSM settlement does not happen instantly because a cycle has not formed, the urgent RTGS payments are immediately settled by transferring liquidity. This is another way of interacting between the LSM and RTGS streams – one of the many possible ones not considered here.

We first look at the liquidity/routing choices of a social planner willing to minimise overall costs, defined as the sum of liquidity costs and delay costs. In the plain RTGS system, the planner's choice is dichotomous: if the price of liquidity exceeds a certain threshold, the planner delays all payments in the internal queues. Otherwise, it delays none, while asking banks to provide some

liquidity. In this case, payments could still be queued in the RTGS stream for a while, if banks run out of liquidity. A similar dichotomy appears in the system with an LSM: the planner uses either only the LSM (when liquidity costs exceed a given threshold), or only the RTGS stream, increasing liquidity in RTGS as the liquidity price falls. Thus, from a central planner perspective, the LSM enhances the operation of the system only in extreme circumstances.

However, payment systems are not run by a 'central planner', but are populated by independent banks interacting strategically. We therefore look at the equilibrium liquidity/routing choices. A typical equilibrium here has banks routing part of their payments to RTGS, and part into the LSM, with the reliance on the LSM increasing with the price of liquidity. Despite the fact that such an outcome is inefficient (the planner would choose either of the two streams, never both), it can still be better than the one emerging without the LSM. So, an LSM may lead to a 'second-best' outcome, improving on the vanilla RTGS system.

The system with an LSM however also possesses some 'bad' equilibria. These feature the somehow paradoxical mix of high liquidity usage, intense use of the LSM, and costs which exceed those of the vanilla RTGS system. The reason behind the existence of such equilibria is probably the following: if many payments are sent in the LSM, this can be self-sustaining, in the sense that each bank finds it convenient to do so. However, the RTGS stream may become less expedite (as fewer payments are processed there), which may in turn imply that the equilibrium level of liquidity is also large. This suggests that LSMs can be useful, but they may need some co-ordination device, to ensure that banks arrive at a 'good' equilibrium.

Most of our results (above all, the ability of an LSM to improve on a vanilla RTGS system) depend on a key parameter: the price of liquidity. We do not perform any calibration of the model's parameters, so we cannot say if our LSM is advisable for any specific system. However, LSMs in general are likely to become increasingly desirable. Indeed, in the wake of the recent financial crisis, banks are likely to be required to hold larger amounts of liquid assets relative to their payment obligations. This may increase their interest in mechanisms that reduce the liquidity required to process a given value of payments.

1 Introduction

Interbank payment systems form the backbone of the financial architecture. Given the value of payments transacted there (typically around 10% of a country's annual GDP daily – Bech *et al* (2008)), their safety and efficiency are of great importance to the whole economy.

The main cost faced by the banks operating in these systems is related to the provision of liquidity, needed to settle the payments. Indeed, most large-value interbank payment systems use the real-time gross settlement (RTGS) modality, whereby a payment obligation is discharged only upon transferring the corresponding amount in central bank money. While this eliminates settlement risk, it also increases liquidity required: if two banks have to make payments to each other, these obligations cannot be 'offset' against each other. Instead, each bank must send the full payment to its counterparty.

The RTGS structure may therefore incentivise free-riding. A bank may find it convenient to delay its outgoing payments (placing it in an internal queue) and wait for incoming funds which it can 'recycle'. By so doing, a bank can avoid acquiring expensive liquidity in the first place. There are three main reasons why such 'waiting strategies' are in practice limited to a level that allows payment systems to actually work. First, system controllers may detect and penalise free-riding behaviour. Second, system participants typically agree on common market practices and may punish non-cooperative behaviour. Third, banks themselves have an interest in making payments in a timely fashion. The cost of withholding a payment too long may eventually exceed the cost of acquiring the liquidity required for its execution.

However, it is a well-known fact that a certain volume of payments is internally queued for a while. These payments do not contribute to any 'liquidity recycling' as they are kept out of the settlement process. A tempting idea is therefore to co-ordinate these pending payments according to some algorithm which may allow saving on liquidity.[1]

These algorithms are called 'liquidity-saving mechanisms' (LSMs), and systems employing them are generally termed hybrid systems. There are many kinds of hybrid systems; the simplest type combines two channels for settlement: one which works by offsetting queued payments, and one which works in RTGS mode. Banks may then use the first for less urgent payment, and the second for transactions that need to be settled instantly.

Given the amounts of liquidity circulating in payment systems (the average daily turnover in CHAPS exceeds £300 billion), and given that banks do delay payments internally, hybrid features may substantially reduce the amount of liquidity needed to process payments. Put differently, given a certain amount of available collateral and a certain volume of payments to settle, adoption of an LSM may increase settlement speed. For these reasons, LSMs are being used increasingly in interbank payment systems: while in 1999 hybrid systems accounted for 3% of the value of

[1] It should be noted that if the mere submission to a central queue does not have legal implications in terms of settlement (ie payments are not settled until perfectly offset), then the settlement risk which led to the demise of end-of-day-netting systems, is not re-introduced. Hence, central queues with offsetting do not defeat the purpose of the gross payment modality

payments settled in industrialised countries, in 2005 their share had grown to 32% (Bech *et al* (2008)). It should be noted that LSMs need not introduce settlement risk. To ensure this, it is sufficient to establish that a payment placed in an LSM creates no presumption of settlement, and its legal status remains identical to that of a non-submitted payment (ie one held in an *internal* queue). Settlement then occurs only when an offsetting 'cycle' forms, at which point payments instantly settle according to the real time, gross, risk-free modality.

In this paper, we argue that introduction of an LSM in an RTGS system amounts to changing the 'game' between participants, thereby changing the trade-off liquidity cost/delay costs. To study this change, we first model a plain RTGS system, where banks decide: (i) the amounts of liquidity to devote to settlement; (ii) how many (and which) payments to hold in internal schedulers. Besides these internal queues, whose size is willingly determined by the banks, this system has also a central queue – one where a bank's payments are queued in a segregated fashion, should a bank accidentally run out of liquidity.[2] This plain RTGS system is then augmented by an LSM. Here banks decide: (i) the amount of liquidity to devote for settlement; (ii) how many (and which) payments to submit to the LSM stream. So, instead of internal schedulers, the banks use the LSM, where payments are settled at zero liquidity cost, as soon as perfectly offsetting cycles form.

Using this setup, we try to answer the following questions:
1) What are the banks' equilibrium choices in the plain RTGS system?
2) How much liquidity and/or delays can the introduction of an LSM reduce in theory, ie if the liquidity and routing choices were made by a benevolent planner?
3) What are the banks' equilibrium choices in the second system (RTGS + LSM)? Are they efficient, and how do they compare with the outcome obtained without LSM?

The paper is organised as follows. Section 2 discusses the model's relationship with the existing literature. Section 3 describes the model. Section 4 solves it and presents the results. Section 5 concludes.

2 Relationship with the literature

There are three branches in the literature on LSMs in interbank payment systems. The first one considers the problem of managing a central queue in insulation. The problem is interesting from an operational research perspective. For example, the 'Bank Clearing Problem'[3] is a variant of the 'knapsack problem' and belongs to a class of computationally hard problems. Hence, there is a need to find approximate algorithms for solving these problems (see eg Güntzer *et al* (1998) and Shafransky and Doudkin (2006)). An exact solution is given by Bech and Soramäki (2002) for the special case where payments need to be settled in a specific order.

The second branch of the literature is aimed at testing the effectiveness of specific LSMs by carrying out 'counterfactual' simulations. This approach has been used before implementation of LSMs into actual systems. Leinonen (2005, 2007) provide a summary of such investigations and

[2] These payments are then settled when the bank receives incoming funds.
[3] The problem of selecting the largest subset of payments that can be settled with given liquidity.

Johnson *et al* (2004) simulate the application of an innovative 'receipt reactive' gross settlement mechanism using US Fedwire data. These works have the advantage of being based on real data, but take behaviour as exogenous (even if sometimes historical data are modified to enhance realism). However, it could be objected that if the system is changed in a significant way, as with the introduction of an LSM, behaviour could change substantially, thus invalidating the data used in the simulations.

Third and last, some theoretical papers model LSMs as games, where bank behaviour is endogenously determined. Martin and McAndrews (2008) develop a two-period model where each bank in a continuum has to make and receive exactly two payments of unit size. Banks have to choose when to make payments, and how (they can choose to pay either via the RTGS stream, or via the LSM). Delayed payments generate costs as does the use of liquidity. Banks may be hit by liquidity shocks – ie the urgency of certain payments is *ex-ante* unknown. The model is solved analytically under assumptions on the pattern of payments that may emerge.[4] As the authors show, an LSM enlarges the strategies available to the banks, as it allows them to make payments conditional on receiving payments. While *a priori* beneficial, this is shown to produce perverse strategic incentives, which may counteract the mechanical benefits of an LSM.

The computational engine for the LSM offsetting algorithm employed in the present paper is borrowed from the first set of papers (Bech and Soramäki (2002)). But as the paper concentrates on the banks' strategic behaviour, it is closely related to the third, game-theoretic branch of the literature. However, in contrast to Martin and McAndrews (2008), we solve our model numerically by means of simulations. Our conclusions are broadly in line with theirs: LSMs may generate efficiency gains. However, undesirable outcomes may also result. In Martin and McAndrews (2008) the overall balance depends on a number of parameters: the size of the system, the cost of delay, the proportion of time-critical payments (in their model, payments are either time-critical or not). Our model instead offers sharper predictions, as the only crucial parameter is the cost of liquidity. This is a consequence of the different (more parsimonious) construction of our model, which also means that any comparison between the two can only be in rather general terms.

Using simulations allows us substantial freedom in designing our model. For example, we need not restrict our attention to the case of exactly two payments sent by (and to) each bank. Nor do we have to look only at a scenario with only two time periods. Instead, we can allow for arbitrarily many payments to be made, in all possible patterns and sequences, over an arbitrarily long day. The cost of a more realistic pay-off function is that all our results are numerical.

3 Model

Our framework is a simple model of a payment system, adjusted in two different ways to describe the two systems that we compare. Banks make choices – to be illustrated later – that jointly determine system performance and hence their costs or pay-offs. The game-theoretic structure of

[4] Eg in a 'long cycle case', payments are all linked in a cycle so the LSM would yield maximum benefits; in another extreme case, payments can only be paired.

the model is straightforward: a single simultaneous-move game, for which we find the Nash equilibria.

As described later, the model has an implicit time dimension. However, this only pertains to the settlement process, ie to the machinery used to derive the banks' pay-offs. However, once the choices are simultaneously made, the expected-value pay-offs are determined so there is no dynamic interaction between banks. A main innovation of the paper is the way pay-offs are determined: they are numerically generated by an algorithm which mimics a payment system in a fairly realistic way.

We allow banks to exchange many payments over many time-intervals, generating complex liquidity flows with 'queues', 'gridlocks' and 'cascades' (see Beyeler *et al* (2007) for details on the physical dynamics of this process). We argue that this enhances realism by incorporating the complex system's internal liquidity dynamics into the pay-off function.

Summing up, the model is a straightforward game-theoretic representation of a payment system, where its complexity is encapsulated in the pay-off function which in turn is computed via simulations. The parameters used in the simulations are summarised in Appendix 2.

3.1 Payment instruction arrival

Our model consists of N banks, who receive payment instructions (orders) from exogenous clients throughout a 'day'.

Each instruction is the order to pay 1 unit of liquidity to another bank with certain 'urgency'. An instruction is thus a triplet (i, j, u), where i and j indicate the payer and payee, and u the payment's urgency (discussed below). Payment instructions are randomly generated from time 0 (start of day) to time T (end of day) according to a Poisson process with given intensity. [5]

For each arriving instruction, payer (i) and payee (j) are randomly chosen from the N banks with equal probability. As a consequence, the system forms a complete and symmetric network in a statistical sense. Each bank sends the same number of payments to any other bank on average. However, this may differ from day to day. On one day a bank may be a net sender vis-à-vis any other bank, on others a net payer.

The urgency parameter u is drawn from a uniform distribution U~[0,1], and reflects the relative importance of settling a payment early. If payment r with urgency u_r, is delayed by t time intervals, it generates a delay cost equal to $u_r t$, to be met by the payer

Completeness of the payment network is a simplifying assumption. However, it is not at all unrealistic for systems with a low number of participants such as the UK CHAPS where banks send and receive payments to and from each other. Symmetry also simplifies our work, and is also useful for technical reasons explained later on. As for the assumption of a uniformly

[5] Details on the parameters are given in the appendix. An alternative strategy to the Poisson model would be to set the length of the day to T time ticks, and generate one payment in each tick, so a bank is hit by T/N payment orders on average. The two models are substantially equivalent, as in the Poisson model 'nothing happens' when no payment is generated. Only, delays are longer in the Poisson process, as even when 'nothing happens', queued payments still generate delays.

distributed u, the simulations show that this is not essential: qualitatively similar results would obtain using a two-modal beta distribution (so most payments are either very urgent, or not urgent at all), or a bell-shaped beta distribution (with most payments 'quite' urgent, and only few a little or very urgent).

3.2 Payment settlement

A bank can route each payment into either of two streams: (i) the RTGS stream or (ii) a queue; the latter can be internal or an LSM. Payments submitted into the RTGS stream settle immediately upon submission, but only if the sender bank has enough liquidity. If the sender lacks sufficient liquidity, the payment is queued in RTGS[6] and is released for settlement when the sender's liquidity balance is replenished by an incoming RTGS payment. Upon settlement, liquidity is transferred from payer to payee. For stream (ii) we consider two cases, corresponding to two models.

The internal queues work in the simplest way: a payment sent in the queue is withheld for the whole day, and submitted in gross terms to the RTGS stream at final time T. While clearly available to banks (barring specific throughput requirements), this second stream represents a rather extreme queuing behaviour. In reality banks may delay payments only for a certain time, and release them following sophisticated rules. We use this very stylised benchmark for the sake of simplicity.

In contrast, the LSM is managed by a controller, who continuously offsets payments on a multilateral basis. To find offsetting cycles, we use the Bech and Soramäki (2002) algorithm. This finds cycles of maximum size under the constraint that each bank's payments are settled according to a strict order.[7] Because payments settle only by perfect offset, the LSM stream requires no liquidity.[8]

Our aim is to compare the two systems. The first system is a natural benchmark for a plain RTGS system. The second one is a specific example of a dual-stream system, as we adopt a specific offsetting algorithm. Our choice is driven by simplicity arguments.[9] For the LSM in particular, we adopt that specific algorithm because this yields optimal outcomes in a precise, technical sense (see Bech and Soramäki (2002)).

3.3 The game: choices and costs

At the start of the day each bank makes two choices: (i) its opening intraday liquidity in the RTGS system $\lambda_i \in [0,\Lambda]$ and (ii) an urgency threshold $\tau_i \in [0,1]$. Payment instructions with urgency greater than τ_i are settled in the RTGS system. Payment with urgency smaller or equal to

[6] While this queue can also be considered a 'central queue', we use the term 'central queue' to refer to the LSM queue.
[7] In our model the ordering is by urgency of the payments.
[8] Apart from payments which are still unsettled at final time T. These are moved into the RTGS stream and settled according to RTGS rules.
[9] As we noted, internal queues may work in a more sophisticated way in our model. Similarly, the LSM could interact with the RTGS in a more complex way than we assumed: for example, the controller might allow payments to be 'retracted' from the LSM and be sent in the RTGS stream.

the threshold are either queued internally or routed to LSM, depending on the model.[10] As the urgency parameter of the payments is drawn from U~[0,1], τ_i is also the (expected) percentage of payments that bank *i* queues internally or routes to LSM.

Once banks have chosen their opening intraday liquidity and urgency threshold, settlement of payments takes place mechanically: banks receive payment instructions and process them according to urgency.

Costs are defined as in Galbiati and Soramäki (2008). At the end of the day each bank pays a total cost, defined as the sum of (a) the liquidity costs incurred in acquiring the opening intraday liquidity and (b) the delay costs, which depend on the delays experienced during the day. Given a profile of choices $\sigma=\{\sigma_1, \sigma_2, ...\sigma_N\}$ where $\sigma_i=(\lambda_i,\tau_i)$ is bank *i*'s strategy, the costs borne by *i* are:

$$\begin{aligned} C_i(\sigma) &= \alpha\lambda_i + D_i(\sigma) \\ &= \alpha\lambda_i + \sum_r u_r(t_r - t_r') \end{aligned} \qquad \textbf{(1)}$$

where α is the price of liquidity and $(t_r\text{-}t_r')$ is the lag between reception and execution of payment *r* with urgency u_r. Delay costs thus increase linearly with payment urgency. The dependence of C_i on τ_i and on 'others' choices σ_j comes via the delays, which depend on the τ's and λ's of all banks in the system.

3.4 Equilibrium

The model has *N* players, actions λ_i and τ_i for each player, and costs/pay-offs determined as described in the above Section. We concentrate on the symmetric equilibria of this game, ie on choice profiles $((\lambda_1,\tau_1),\ldots(\lambda_i,\tau_i),\ldots(\lambda_N,\tau_N))$ such that: (i) all banks choose the same actions $((\lambda_i,\tau_i) = (\lambda_j,\tau_j) \forall\, i, j)$ and (ii) each (λ_i,τ_i) is a best reply to others' choices. We call a strategy profile 'equilibrium' only if any unilateral deviation from it is not beneficial to the deviator – even if it would lead to a non-symmetric outcome.

However, by restricting attention to symmetric equilibria, we may miss equilibria where banks adopt different, albeit mutually optimal, choices. Our focus on symmetric equilibria is mainly dictated by simplicity reasons. However, extra-model considerations suggest that such asymmetric equilibria (should they exist) would be unlikely to survive in reality. First, symmetry seems 'reasonable', as banks are homogeneous in our model. Second, if a bank posted less liquidity than others, it might be seen to 'free-ride' and could be penalised in the long run. Finally, in reality, banks do not know the choices of their counterparties. What they typically do know is some average indicator of the whole system, and this is what they play against. If *N* is large, all banks will face the same 'average opponent', and being identical, they will all choose the same best reply to that. This confirms that symmetric equilibria are the ones to concentrate on in this paper.[11]

[10] More complex routing rules are conceivable. We restrict attention to this for simplicity.

[11] Equilibria where banks choose the same liquidity but different thresholds are unlikely for theoretical reasons. The more *i* uses the LSM, the more any other *j* should use it. The liquidity choices are different. Here, the substitutability effects may well induce asymmetric equilibrium behaviour: for example, low-l_i, high-l_j may be part of an equilibrium because, from *i*'s viewpoint, *j*'s liquidity is a substitute for *i*'s own liquidity.

4 Results

In Section 4.1 we illustrate the mechanics of settlement in the two systems (with and without an LSM). We show how delays, and hence costs, depend on banks' choices of liquidity and thresholds. All the results are obtained numerically by simulating the settlement process for each combination of 'my choices' vs. 'others' choices', thus obtaining the pay-off function of our game.[12] Details on the numerical exploration of the delay and cost function are given in the Appendix. In Section 4.2 we then identify and compare the corresponding equilibria. In most of what follows, we take the viewpoint of a single bank ('I'), facing the rest of the system ('Them').

4.1 Settlement mechanics

This Section shows how delays are determined by the banks' choices in the two systems. Hence, it illustrates the possible benefits that an LSM may yield if handled by a 'benevolent central planner'. Reference will be made throughout to the delay costs, ie the urgency-adjusted delays D defined in equation **(1)**.[13] Section 4.1.2 shows the banks' overall costs, defined by equation **(1)** as the sum of delay plus liquidity costs.

Concerned with the mechanics of the settlement process, this Section does not consider choices as strategic (this is the topic of Section 4.2). In game-theoretic terms, this Section is about the pay-off function of the game underlying our model.

4.1.1 Delay costs

Recall that both models establish a relationship between (i) liquidity provided, (ii) delay threshold chosen and (iii) payment delays. Such a relationship is shown in Figure 1 for the two models: in each case, banks choose the same λ and τ. This picture suggests that a system with an LSM (lower surface) can substantially reduce overall delays.

For any choice of liquidity and threshold, the system with LSM does better than the vanilla system for the following reason. The RTGS stream produces exactly the same amount of delays in the two cases. Instead, the central queue (LSM) is more efficient than the system of internal queues. Indeed, in the latter payments are delayed until the end of the day, accumulating the maximum possible delay.

By concentrating on this extreme type of internal queuing we make the LSM/internal queue difference as stark as possible. This 'takes apart' the surfaces in Figure 1. It is difficult to assess *a priori* what the effect would be *on equilibrium choices* of 'smarter' internal queuing routines, whose effect would be to bring the two surfaces of Figure 1 closer to each other. As mentioned

[12] We look for symmetric Nash equilibria, so we only need consider profiles where all 'others' make the same choices: $\sigma_j=\sigma_k$ for $j, k \neq i$; this reduces the parameter space from $([0,\Lambda]\times[0,t])^N$ to $([0,\Lambda]\times[0,t])^2$.

[13] The Appendix decomposes such costs into their two sources: costs arising from delays in RTGS, when a bank submits a payment but does not have sufficient liquidity, and costs arising from delays in the queues, internal or central, ie LSM.

above, there is an endless variety of queuing rules that could be adopted (both for the LSM and internal scheduler) – we limit ourselves to model simple choices.

Figure 1: Delay costs for system with internal queues (top) and with LSM (bottom)

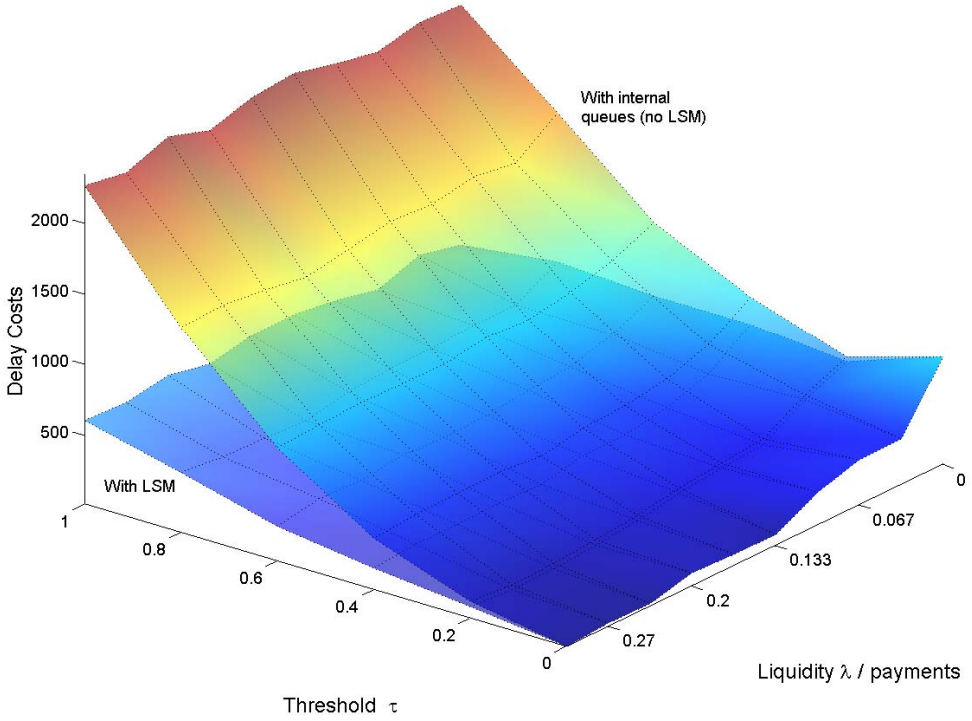
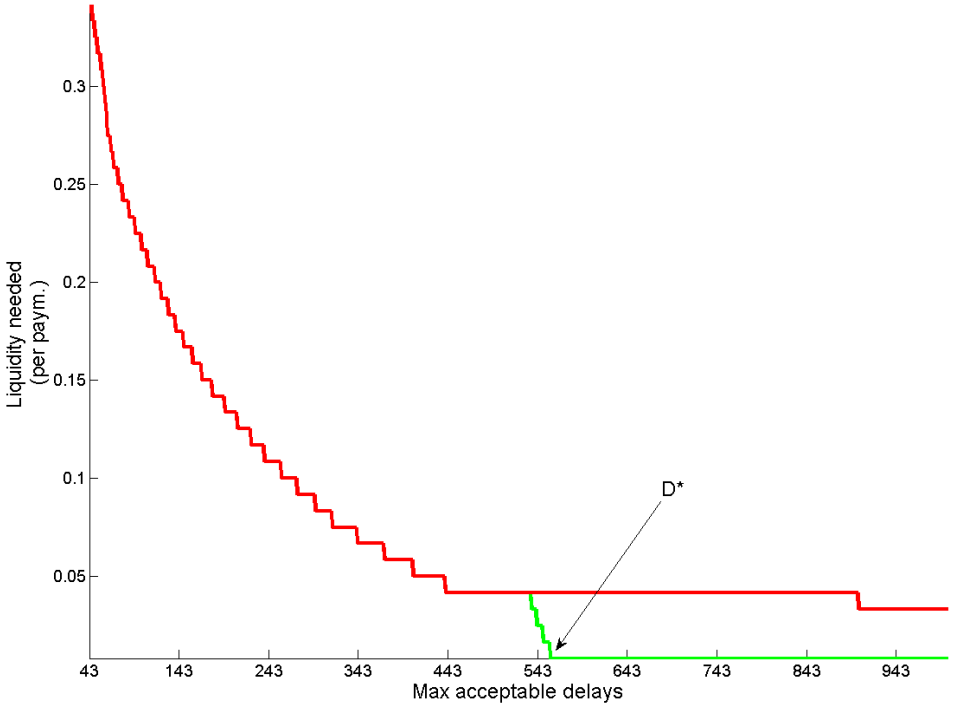


Figure 1 may not fully express the gains from an LSM. Indeed, the threshold τ is a choice variable which has, *per se*, little economic meaning. The key question on the mechanical gains from an LSM is probably:

Given a level of delays, what is the minimum amount of liquidity needed not to exceed it (allowing any choice of τ)?

Figure 2 answers this question. Here, an LSM is shown to reduce liquidity needs only if the targeted delays level does not fall below a threshold D^*. Indeed, if delays are required not to exceed such level, a 'central planner' would have no option but to use the RTGS stream for all payments by setting τ=0, and providing the system with large amounts of liquidity. Once the LSM is unused, its nature becomes irrelevant.

Figure 2: Liquidity needs for given amount of delays (Red: internal queues. Green: LSM)



So, if delays are allowed to exceed D^* the planner sets τ=1, relinquishing the RTGS stream and using exclusively the LSM stream. Hence, the central planner makes a dichotomous choice: it uses either one of the two streams (depending on the constraints imposed on delays/liquidity), but never both at the same time. The reason for this dichotomy is that both RTGS and LSM feature decreasing returns in the volume of processed payments.[14]

Given these results, it seems easy to jump to the rather negative conclusion that an LSM is useful only when participants are prepared to accept very long delays, or if the system is extremely short of liquidity. In both cases, the perspective of *de facto* abandoning the RTGS mode (because only the LSM is used) might raise concerns. This conclusion is incorrect though: the results presented so far are about the mechanics of the two systems, as handled by a central planner enforcing liquidity and payment choices. The real gains from an LSM emerge when choices are made by independent banks in a strategic context. Section 4.2 will show that in the banks' equilibrium choices typically both streams are used and that the LSM is advantageous for a broad range of parameter values. Before illustrating these strategic choices in Section 4.2, we finish describing the system's mechanics, presenting its total costs.

[14] See Appendix 2 on this.

4.1.2 Total costs (pay-off function)

The previous Section has illustrated the dependence of delay costs on liquidity and thresholds. Total costs are obtained by adding liquidity costs to delay costs (see equation **(1)**). We thus arrive at the total costs, or the pay-off function of our game.

Figure 3: Total costs in the two systems

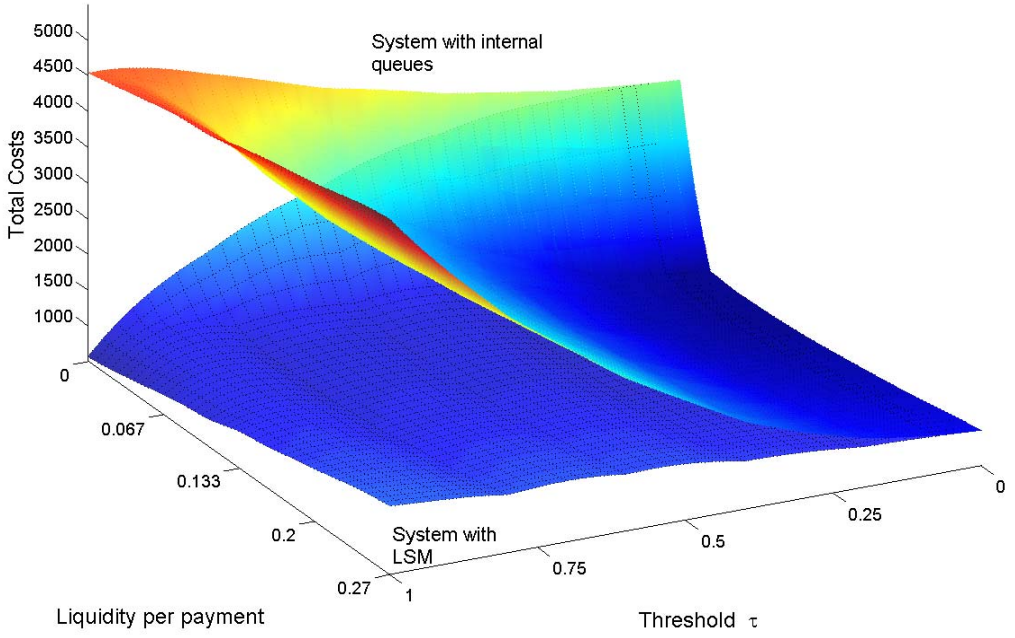


Figure 3 illustrates a representative bank's costs (for an arbitrary liquidity cost level α), when every bank chooses the same λ and τ. An LSM yields potentially large gains in terms of total costs – especially for low levels of liquidity and high usage of the LSM.

Figure 4: 'My' costs for different choices by 'others' – with LSM

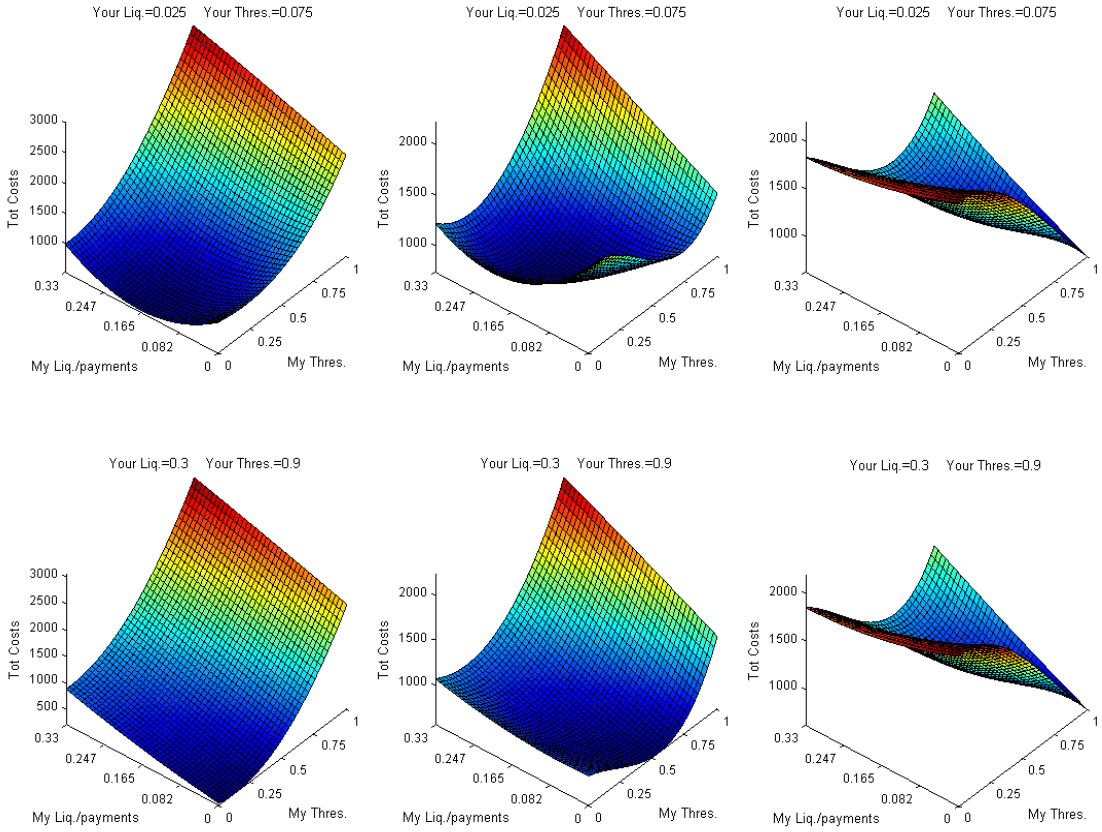


Figure 4 refers to a system with an LSM. It shows how 'my' costs depend on 'my' choices, for various 'others' choices, which are possibly different from 'my own'. The first row shows the impact of 'my' choices when 'others' liquidity is low, and the second row where it is high. Moving left to right along a row, the 'others' threshold is increased. It can be noted that the dependence of 'my' pay-offs on 'my' choices varies dramatically. For example, when others provide little liquidity and route most payments to RTGS (top-left), it is beneficial for me to route all payments to RTGS, but provide somewhat more liquidity than the others. On the other hand, if other banks provide large amounts of liquidity and route half of payments to RTGS (bottom-middle), 'I' should provide no liquidity (others already do so sufficiently) and should route most payments to RTGS.

The efficiency of each stream (RTGS or LSM) depends on the share of payments routed to it by the 'others' (ie by their threshold). In addition, the RTGS stream also depends on the amount of liquidity committed by 'others': the more liquidity 'others' post, the more attractive the RTGS becomes, as 'my' liquidity and 'others' liquidity are complements.

4.2 Equilibria

The remainder of the paper looks at the equilibria reached by the banks in the two models. As mentioned earlier, we concentrate on symmetric equilibria, ie action profiles where all banks make the same choices. To find these, we look (numerically) at fixed points of the best reply correspondence, ie for each choice by the 'others', we compute 'my' optimal choice of τ and λ. Next, we look for choices which are best replies to themselves.

A key parameter in the model is the price of liquidity α in equation **(1)**. This is arguably the variable over which central banks and policymakers have the greatest influence. Thus we look at how the equilibrium varies when the price of liquidity α changes. Because an accurate calibration of the model is beyond the scope of this paper, we let the parameter α vary in a range wide enough for the equilibria to span the whole strategy space – keeping the price of delays fixed.

4.2.1 RTGS with internal queues

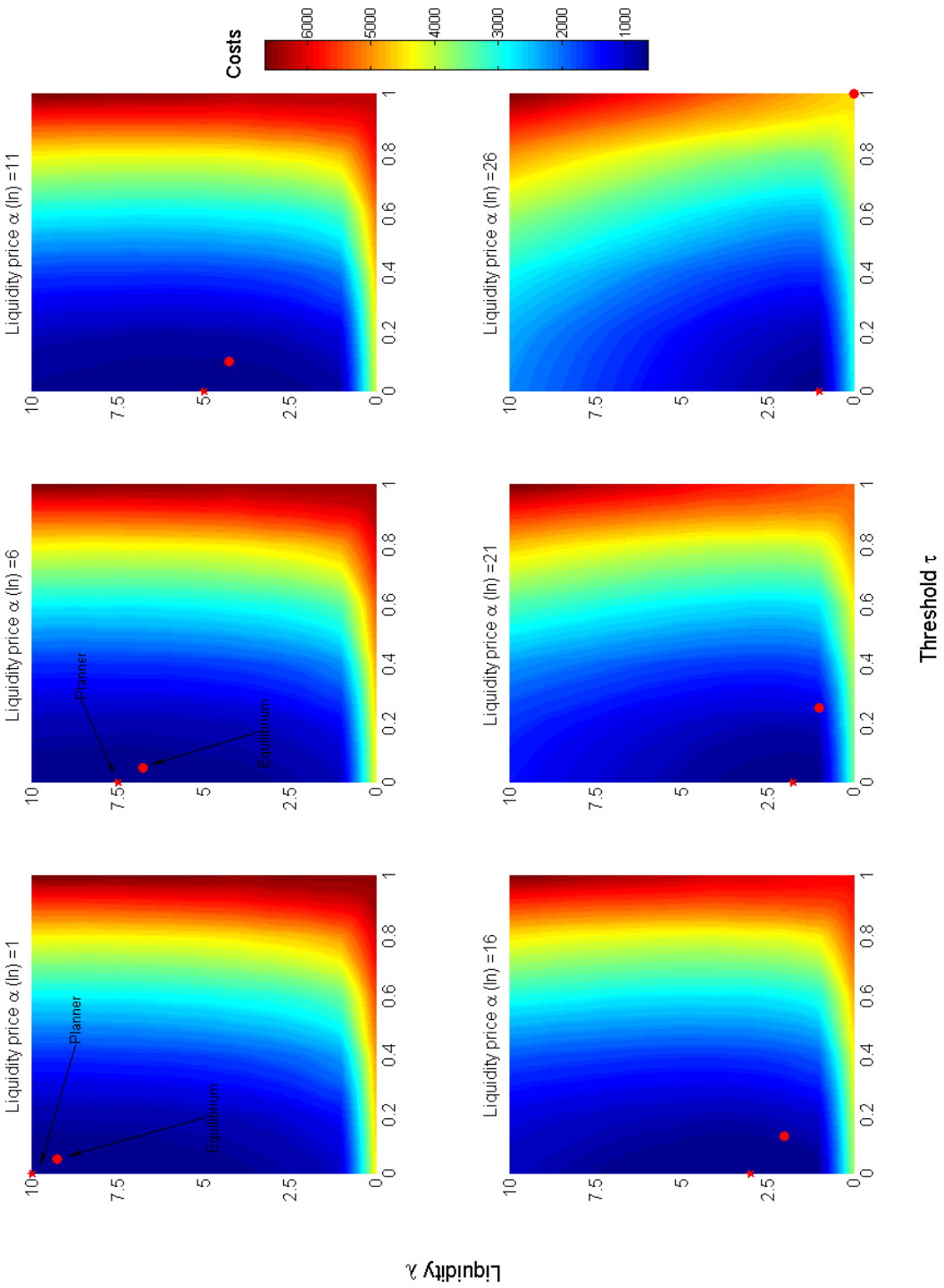
Figure 5 shows equilibria and the planner's choices for different prices of liquidity. Equilibrium choices are represented by dots, while the planner's choices are represented by stars. The background gradient shows system-wide costs which are, by definition, lowest at the planner's choice.

As anticipated in Section 4.1.1, the planner's choice of τ is dichotomous. Either all payments are sent to RTGS, or they are all internally queued until the end of the day.

Equilibrium choices are less stark, and banks typically use both queues and RTGS. When the relative price of liquidity rises, banks post less liquidity and resort more to using internal queues. Importantly, the equilibrium is inefficient: a cost-minimising planner would provide more liquidity to the system and would never delay payments internally. Equilibrium costs are always more than 15% higher than the social optimum, reaching multiples thereof at high liquidity prices. Only when the liquidity price is *extremely* high, the equilibrium coincide with the planner's optimal choice, both being $\lambda=0$, $\tau=1$.

The reason for this inefficiency is explained by two externalities. On the one hand, there may be a positive externality in liquidity provision: incoming payments to a bank can be recycled for making other payments, so liquidity is in a sense a common good, as in Angelini (1998). As a result, equilibrium liquidity provision (λ) falls short of the social optimum. On the other hand, internal queues generate a negative externality: banks have an incentive to delay the less urgent payments and use liquidity for more urgent ones. But by doing so, they slow down the beneficial liquidity recycling in RTGS, which in turn affects other banks. Hence banks queue more than they should from a social perspective – ie τ exceeds the level that would be chosen by the planner.

Figure 5: Equilibria and planner's choices without LSM. Each chart refers to a given price of liquidity (lowest price on top-left; highest price on bottom-right)



4.2.2 RTGS with internal LSM

As with internal queues, the routing of payments to an LSM allows banks to reserve liquidity for urgent payments in RTGS. But, while internal queues merely postpone settlement until the end of the day, our LSM allows for settlement as soon as offsetting cycles are found, without making use of liquidity. Thus our LSM offers a potentially more efficient way of queuing payments.

However, increased efficiency of the second stream induces banks to use it more intensely. Also, an increase in τ causes a reduction in RTGS volumes, which in turn causes this stream to lose in terms of efficiency. Hence, there is a trade-off between the efficiency levels of the two streams.[15] When 'played with' by individual banks, these effects produce unexpected outcomes, as we show next.

Figure 6 shows that, when liquidity costs change, the equilibria change essentially as in the RTGS with internal queues model.[16] In particular as the price of liquidity (α) rises, liquidity provision (λ) decreases and usage of the LSM (τ) increases. Compared to the social optimum, liquidity provision is too low and payments are delayed too much. For very high relative prices of liquidity, both banks and the planner rely exclusively on the LSM. Only then is the equilibrium efficient. The planner never uses both streams at the same time – as was already discussed in Section 4.1.1.

The main novelty with an LSM compared to internal queues is that, for an intermediate range of liquidity costs, multiple equilibria emerge (shown as separate clusters of ε-equilibria). In a typical case, there are up to three equilibrium types:

a) a corner equilibrium $\lambda=0$, $\tau=1$ (all payments vian LSM);
b) equilibria with moderate use of both liquidity and LSM ('good equilibria'); and
c) equilibria with high usage of both liquidity and LSM ('bad equilibria').

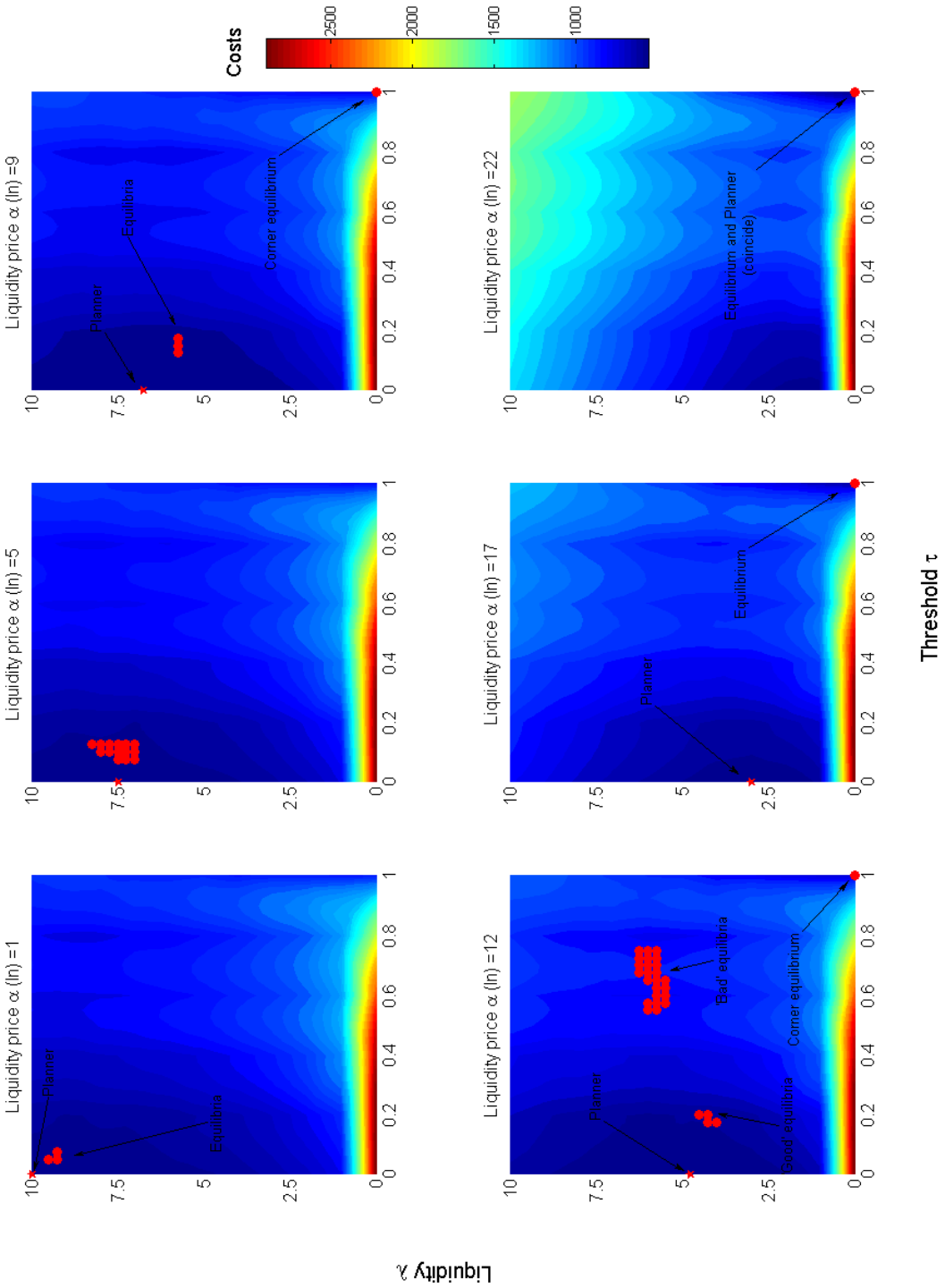
Equilibria under (a) are socially optimal (ie coincide with the planner's choice) only in the extreme circumstances of very high liquidity costs. However, they are typically better than the equilibria under (c). These latter are called 'bad' as they feature the highest costs among all equilibria. Equilibria under (b) are typically those with lowest costs, so we call them 'good'.

Although 'bad', the equilibria under (c) are interesting for their paradoxical features: high liquidity usage (λ) *and* high reliance on the LSM (τ). Their existence is probably explained as follows. LSM features economies of scale (see Section 4.1), so its high usage may be self-sustaining. But, as mentioned at the start of this Section, overuse of the LSM is detrimental to the RTGS stream, which may in turn require inefficiently high amounts of liquidity.

[15] This is not the case with internal queues, where average delay times are independent of τ.

[16] For technical reasons related to numerical approximations in the simulations, we do not look at Nash equilibria, but at ε-Nash equilibria, ie strategy profiles from which unilateral deviation yields a gain not exceeding a (small) ε. These equilibria come in 'clouds'. Here $\varepsilon = 0.001$: unilateral deviations improve pay-offs by less than 0.1%. Socially optimal choices (planner's) are shown as a star.

Figure 6: Equilibria and planner's choices for a system with LSM. Each chart refers to a given liquidity price α (lowest α on top-left; highest α on bottom-right)



4.2.3 Comparison of the two systems

We now compare the two models: RTGS with internal queues and RTGS with LSM. With LSM we have 'clouds' of ε-equilibria; hence, for each cloud, we pick average values of costs, liquidity and thresholds. A 'bad' cloud is then that with the highest average costs, and the 'good' cloud the one with the lowest average cost.

The blue lines in Figure 7 show the equilibrium costs attained with an LSM, normalised by the corresponding costs obtained without LSM. The solid line represents 'good' equilibria and dots represents (average values of) 'bad' equilibria. The charts on the right zoom into those on the left.[17]

Looking at the cost ratio first we see that e.g. with a liquidity price α=13, the good LSM equilibria are about 5% cheaper than the equilibrium with internal queues. Savings become more sizable at higher prices of liquidity, when banks commit less liquidity, and channel more payments into the LSM than they would queue internally.

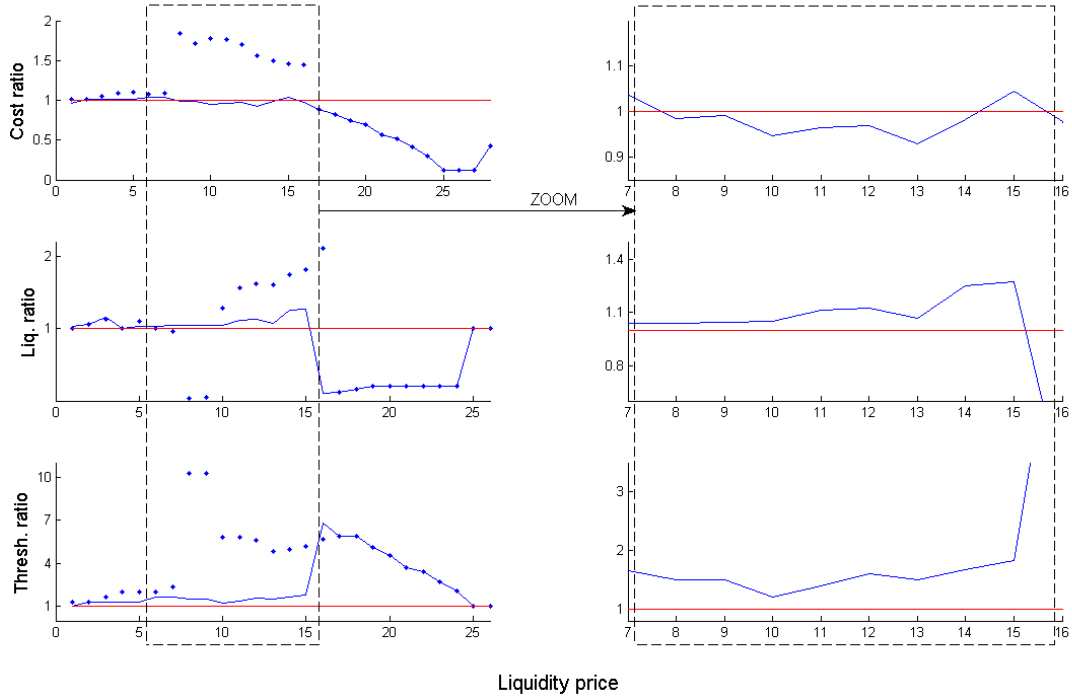
However, for an intermediate range of liquidity costs, there exist bad equilibria where total costs are 50%-70% higher than the costs in the good equilibria. Bad equilibria are characterised by both a high liquidity usage, and a more intense use of the LSM.

Somehow surprisingly, for liquidity costs not exceeding a certain threshold, liquidity usage can be higher *with* LSM than without (see Figure 7, second row). This is because, attracted by an efficient LSM, banks send more payments there, and thus fewer in the RTGS stream (higher threshold). As a consequence, the RTGS stream loses in efficiency (less recycling takes place), something that banks make up for by posting more liquidity – unless liquidity costs are so high that banks accept higher delays for those (relatively small) volumes routed in RTGS.

For intermediate liquidity prices, the LSM 'bad' equilibria (dots) yield costs >1 (ie higher than without LSM). Instead, 'good' LSM equilibria (blue line) yield costs <1 (ie lower than without LSM).

[17] The jump at the 16th value of the liquidity price is due to the fact that, at that point, the cloud of 'good equilibria' in the LSM system suddenly disappears, leaving the corner (λ=10, τ=1) as unique equilibrium – see Figure 7. Discontinuity of Nash equilibria is common, even when a 'smooth' pay-off function is modified in a continuous way (in our case, the smooth function in Figure 8 changes smoothly, when the liquidity price changes).

Figure 7: LSM costs, liquidity and threshold, standardised by the corresponding values without LSM



5 Conclusions

This paper compares two stylised payment systems. In both of them, banks can queue non-urgent payments to reserve liquidity for urgent ones. In the first system, queued payments are held 'internally' and are submitted by the banks for settlement at the end of the day. In the second system, queued payments are routed to an LSM and are settled throughout the day by an offsetting algorithm. As expected, LSM is more efficient than decentralised queuing as payments are settled throughout the day (Figure 1 and Figure 3). As mentioned in the introduction, an LSM needs not introduce settlement risk: settlement occurs only when an offsetting 'cycle' forms, at which point payments are settled in real time, gross modality.

We first look at the liquidity and queuing threshold choices of a social planner who is minimising total costs. When only internal queues are available, the planner delays payments only for extremely high liquidity costs (Figure 5). In all other circumstances the planner submits all payments to the RTGS stream. The same is true if the planner can use an LSM: it sends payments in the LSM only if liquidity is extremely expensive; otherwise, is settles all payments via the RTGS stream (Figure 6). Thus, from the planner's perspective, an LSM is only valuable in extreme circumstances.

When looking at banks acting strategically, we find fewer dichotomous choices: for a range of liquidity prices, banks use both direct RTGS settlement and queues (whether these are internal queues or the LSM). They also queue more at higher liquidity costs. In equilibrium, banks underprovide liquidity and suffer both higher delays and overall costs than socially optimal: the 'central planner' would only use the RTGS stream and provide it with more liquidity than the banks. This inefficiency is due to the fact that banks tend to free-ride on each others' liquidity.

However, a system with an LSM is capable of delivering better outcomes than those resulting in the absence of an LSM. Indeed, for a range of liquidity prices, there are 'good' LSM equilibria which feature lower total costs than their non-LSM counterparts. This is due to faster settlement and often (although not always) due to lower liquidity usage (Figure 7).

Our results point out a caveat, though: for a range of liquidity prices, a system with an LSM may also generate 'bad' equilibria with (i) high liquidity usage, (ii) intense use of the LSM and (iii) high costs exceeding those obtained without LSM (Figure 7, zoomed area). This suggests that liquidity-saving mechanisms can be useful, but they may need some co-ordination device, to ensure that banks arrive at a 'good' equilibrium.

Appendix 1: Simulation parameters

We use simulations to determine the pay-off function of our game, ie the relationship between choices and total costs, illustrated in Section 2.3. We use the following parameters:

- The number of banks $N = 5$.
- The Poisson process (and length of the day) is calibrated as to produce an average of Z=30 daily payments per bank. This number is essentially arbitrary. However, we know that, under normal circumstances, CHAPS banks hold liquidity in the range of 5% to 25% of their gross daily payments. This suggests that, if we fix Z=30, Λ should be chosen such that Λ/Z exceeds such 25% upper bound (so equilibrium choices are not artificially constrained by our parameters choice), but is somewhat comparable to it. This is why we set Λ=10, so Λ/Z=33%.
- The price of liquidity α is also arbitrary; however, when we vary it for our comparative statics exercises, its range needs to be calibrated in a somewhat meaningful way. As mentioned above, a realistic value for the ratio (equilibrium liquidity)/Z is in the region [0.05, 0.25] for CHAPS banks. Thus, we let α vary in a range which produces equilibrium choices falling in a comparable range. And indeed, in our experiments a bank's equilibrium liquidity range from 0 to 10, ie from 0 to 33% of daily gross payments (Z=30).

To compute the pay-off function of bank i (equation **(1)**), we need to find the delays experienced by i when the rest of the system chooses $\{(\lambda_j\tau_j)\}$, for $j \neq i$. As mentioned in the main text, we look at symmetric equilibria. Hence, we treat the 'rest of the system' as one player assigning to each 'other' bank the same action. This greatly reduces the action profiles to explore, because now the delays $D_i((\lambda_i\tau_i),(\lambda_i\tau_j))$ are a function of four variables only. We compute them as follows.

We run simulations for a restricted number of two-player action profiles. In particular, we simulate the settlement process for λ taking on all integers in [0,10], and τ any number in [0,0.2,0.4,..1]. That is, we compute $11^2 \times 6^2 = 4'356$ values of the delay function, for just as many action profiles. Because payment orders arrive in a random order, we need to simulate a good number of 'days' for each action profile, to obtain a reliable estimate of the 'average day'. We deemed that n=200 days for each action profile are a large enough sample – because at that point the observed average across days becomes little sensitive to further increases in n. Hence, we simulate $200 \times 11^2 \times 6^2 = 871'200$ days in total.

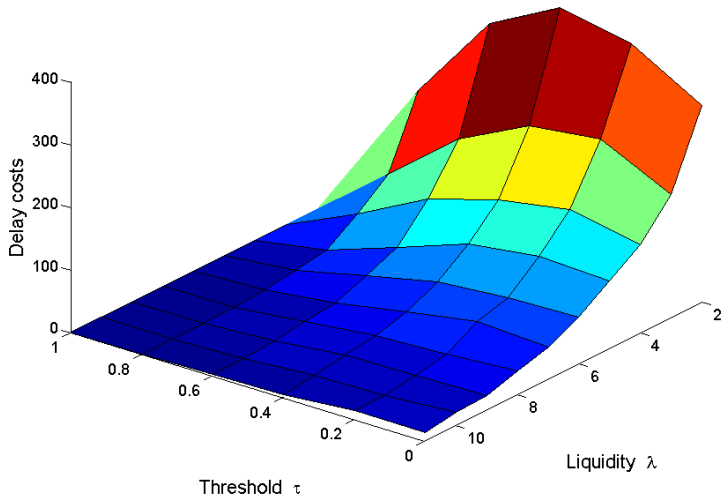
Yet, 11×6 choices for each bank are not enough to obtain 'smooth' results: when computing the equilibria, undesired artefacts emerge. Hence, we numerically smooth out and interpolate the delay function $D_i((\lambda_i\tau_i),(\lambda_i\tau_j))$ on a refined grid, a four-dimensional cube with $41^4 = 2'825'761$ points, which correspond to banks choosing λ in [0,10] in steps of 0.25 (41 liquidity levels) and τ in [0,1] in steps of 0.025 (41 threshold levels). This is the delay function $D_i((\lambda_i\tau_i),(\lambda_i\tau_j)) = D(\sigma)$. Adding liquidity costs as in equation **(1)**, we get to the pay-off function.

Appendix 2: Decomposition of delay costs

A) RTGS stream

Figure 8 shows how delay costs in RTGS depend on λ and τ when all banks make the same choices (we choose this representation for clarity; in reality 'my' delays depend on four variables: my choices of λ and τ, and 'their' choices of λ and τ).

Figure 8: Delay costs in RTGS as a function of threshold and liquidity



Obviously, delay costs are reduced by increasing liquidity (unless τ=1, because then no payment is actually directed into RTGS). It can be noted that 'returns on liquidity' are decreasing, ie an additional unit of liquidity reduces delays more when liquidity is low, than when it is high.

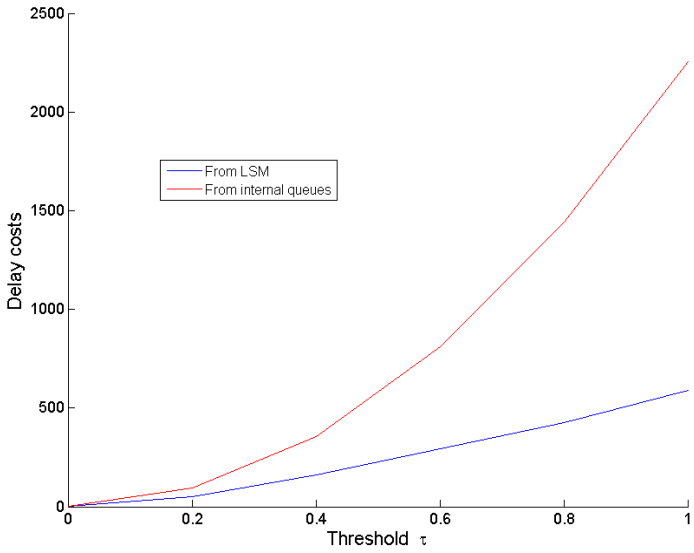
An increase in the threshold (ie less payments routed to RTGS) increases delay costs for low levels of τ - the more so, the less the available liquidity. This is probably due to the fact that, as low urgency payments are subtracted from RTGS, 'liquidity recycling' is disrupted. This effect is eventually balanced by the fact that fewer payments can be settled swiftly with less liquidity. Interestingly, liquidity has a stronger impact in reducing delays when not all payments are routed to RTGS (τ>0). Indeed, if all payments are routed to RTGS, liquidity is absorbed by less urgent payments too, so its 'returns' in terms of decreasing delay costs are reduced.

The relationship between τ and RTGS delay costs is generally non-monotonic: when liquidity is scarce, it is not convenient to route too many payments to RTGS: low-urgency payments may clog the system and cause more urgent ones to be unduly delayed. When liquidity is abundant, it is worthwhile to route all payments to RTGS, to minimise delays.

B) Second stream (internal queues or LSM)

Delay costs in internal queues are simple. Obviously they are independent of λ, as internal queues consume no liquidity during the day.[18] On the other hand, internal queues' delay costs are a quadratic function of τ. Indeed, every payment settles at the end of the day, so the average time spent in the queue is half a day, ie $T/2$. The urgency of each payment is uniformly drawn from [0,τ], so it is τ/2 on average. Hence, directing a volume of payments τ through internal queues produces delay costs totalling $(T/2)(\tau/2)\tau=(1/4)T\tau^2$.

Figure 9: Delay costs from internal queues and from LSM



Delay costs in LSM are also independent of λ. More specifically total delays can be calculated as $x\cdot(\tau/2)\cdot\tau$, where x is the average time delayed, (τ/2) the average urgency and τ the volume routed to RTGS. Simulations show that total delays scale as $\alpha\cdot\tau$, so one infer that $x\sim\alpha\cdot 1/\tau$. In a sense, LSM displays increasing returns to scale with respect to processed volumes. The larger the pool of payments on which an LSM searches for cycles, the more likely (and longer) will be the cycles themselves.

C) Comparing delay costs in the two streams

Figure 10 and Figure 11 show the two components of delay costs: RTGS delays, and second-stream delays. On the horizontal axis, the threshold τ; on the vertical axis, the costs. The four panels are for increasing levels of liquidity.

[18] Only at the end of the day, are queued payments sent to RTGS and settled there. But, as they are added to the RTGS balance, the total amount of non-executed payments will equal the difference between incoming and outgoing payment orders. This is exogenous and so independent of banks' choices.

Figure 10: Total delay costs: RTGS with internal queues as a function of threshold. Each chart in the panel represents a given level of liquidity in the system

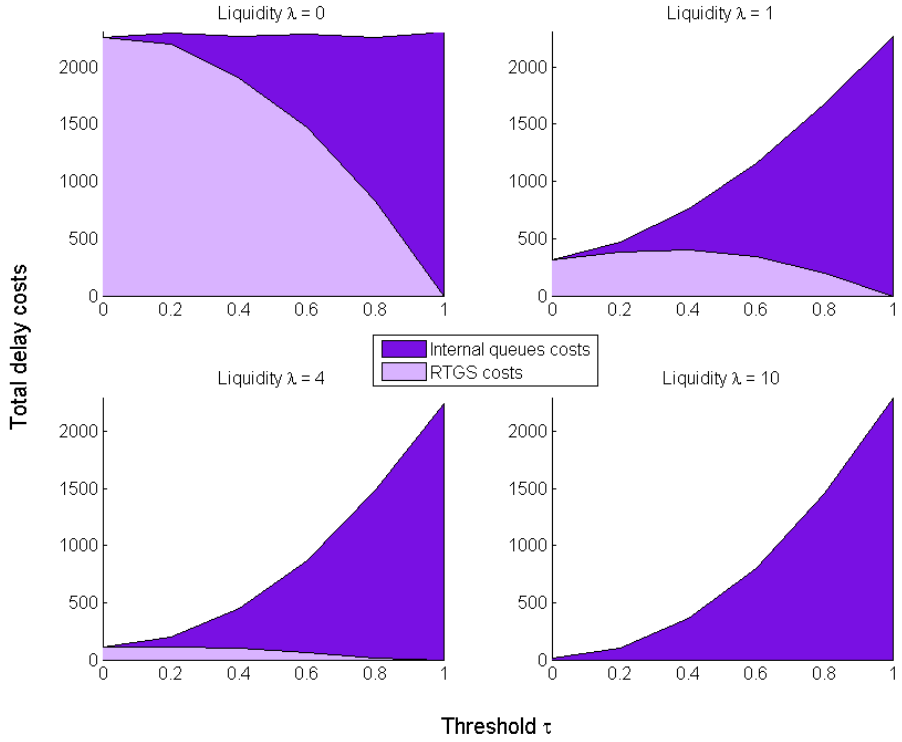
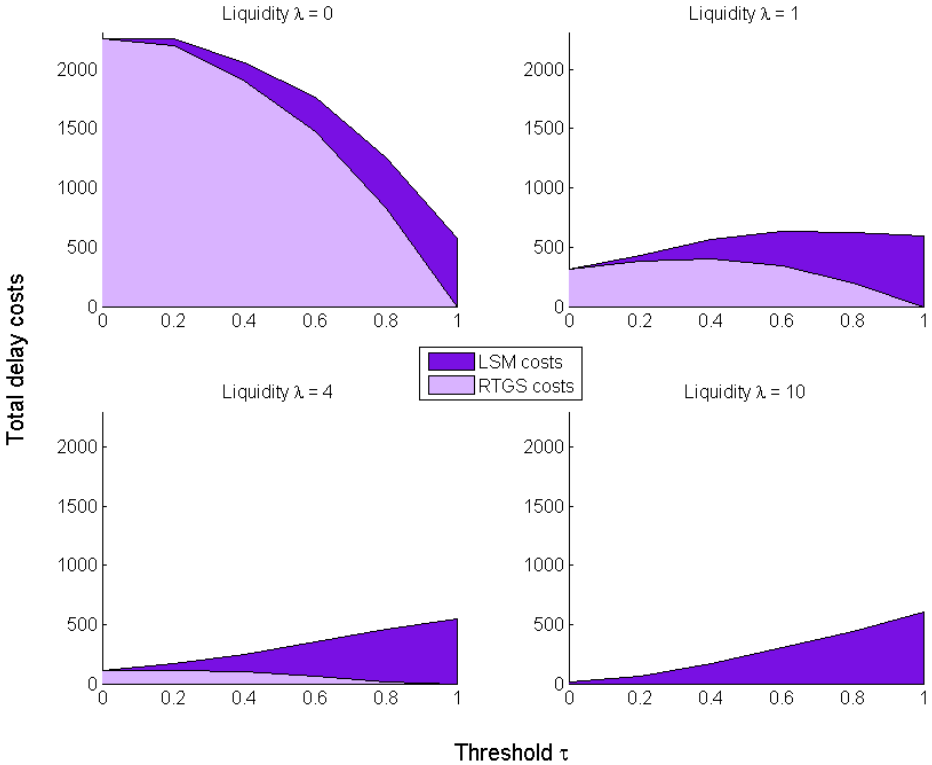


Figure 11: Total delay costs: RTGS with LSM as a function of threshold. Each chart in the panel represents a given level of liquidity in the system



References

Angelini, P (1998), 'An analysis of competitive externalities in gross settlement systems', *Journal of Banking and Finance*, Vol. 22, pages 1-18.

Bech, M, Preisig, C and Soramäki, K (2008), 'Global trends in large value payment systems', *Federal Reserve Bank of New York Economic Policy Review*, Vol. 14, No. 2.

Bech, M and Soramäki, K (2002), 'Liquidity, gridlocks and bank failures in large value payment systems', *E-money and Payment Systems Review,* Central Banking Publications, London.

Beyeler, W, Bech, M, Glass, R and Soramäki, K (2007), 'Congestion and cascades in payment systems', *Physica A*, Vol. 384, Issue 2, pages 693-718.

Galbiati, M and Soramäki, K (2008), 'An agent-based model of payment systems', *Bank of England Working Paper no. 352.*

Güntzer, M M, Jungnickel, D and Leclerc, M (1998), 'Efficient algorithms for the clearing of interbank payments', *European Journal of Operational Research*, Vol. 106, pages 212-19.

Johnson, K, McAndrews, J J and Soramäki, K (2004), 'Economizing on liquidity with deferred settlement mechanisms', *Federal Reserve Bank of New York Economic Policy Review* Vol. 10, No. 3, pages 51-72.

Leinonen, H (2005) (ed), 'Liquidity, risks and speed in payment and settlement systems – a simulation approach', *Bank of Finland Studies*, E: 31.

Leinonen, H (2007) (ed), 'Simulation studies of liquidity needs, risks and efficiency in payment networks', *Bank of Finland Studies*, E: 39.

Martin, A and McAndrews, J J (2008), 'Liquidity-saving mechanisms', *Journal of Monetary Economics*, Vol. 55(3), pages 554-67.

Shafransky, Y M and Doudkin, A A (2006), 'An optimization algorithm for the clearing of interbank payments', *European Journal of Operational Research*, Vol. 171(3), pages 743-49.