Liquidity saving mechanisms  
and  
bank behavior  

Marco Galbiati†  
Kimmo Soramäki‡  

28 January 2009  

Abstract  
We investigate the benefits of liquidity saving mechanisms in interbank payment systems. We set up, simulate and compare two models, representing respectively a ‘vanilla’ payment system, and a payment system with a liquidity saving mechanism.  
In the first system, banks can route payments into a real-time gross payment stream (RTGS) or can queue them internally in what we call a liquidity management mechanism (LMM). In the second system banks choose between the RTGS stream and a central liquidity saving mechanism (LSM). In both systems banks choose the intraday liquidity balances for the RTGS stream. At the end of the day banks pay costs that depend on the chosen intraday liquidity balances (liquidity costs) and on the delays experienced during the day (delay costs).  
We compare the equilibrium choices in the two models with each other and with the choices of a benevolent planner. By so doing, we draw conclusions on the efficiency and desirability of the two systems.

† Bank of England. E-mail: Marco.Galbiati@bankofengland.co.uk. The views expressed in this paper are those of the authors, and not necessarily those of the Bank of England.  
‡ Helsinki University of Technology. E-mail: kimmo@soramaki.net. Contact information at: www.soramaki.net.  
The authors thank participants to the 6th Bank of Finland Simulator Seminar (25-27 August 2008) and seminar participants at Bank of England (13 January 2009).
1 Introduction

Interbank payment systems are used by banks to settle claims that arise from their trading activities with each other or from customer demands to transfer funds from one bank or another. These systems form the backbone of the financial architecture and their safety and efficiency is of high importance to the underlying economy. The daily flow of payments carried in interbank payment systems generally accounts to 10% of the annual gross domestic product of a country (Bech et al 2008). The main direct cost for banks in these systems (in addition to operations costs) are costs related to liquidity that is needed to settle the payments. On the other hand banks would wish to settle payments rather earlier than later.

Most interbank payment systems use real-time gross settlement (RTGS) as the modality for settling payments. In RTGS payments are settled individually and only if cover for their settlement is available. As a consequence RTGS payment systems require large amount of liquidity: if two banks have to make payments to each other, these transfers cannot be compensated: either bank actually must send the full payment to its counterparty. However, once a bank receives funds, it can “recycle” the funds intraday as cover for its own payment. This structure incentivizes free-riding: a bank may find it convenient to delay its payments (placing it in an internal queue), waiting for incoming funds, and thus avoiding the burden of acquiring expensive liquidity in the first place. There are three main reasons why such “waiting strategies” in practice are limited, so that payment systems actually work: first, intervention by system controllers, who typically sanction free riding behaviour, when detected. Second, peer pressure: the system’s participants themselves may punish non-cooperative behaviour. And third, delay costs: banks have an interest to make payments in a timely fashion; the cost of withholding a payment may eventually exceed the cost of acquiring the liquidity required to its execution, and so banks do not wait indefinitely.

However, it is well known that a certain volume of payments is internally queued for a while. While kept in the internal schedulers, these payments do not contribute to “recycling liquidity”, as they are kept out of the settlement process. They just sit on the books of the banks, and by so doing they generate costs. A tempting idea is therefore to pool these payments in a central queue, to settle them more efficiently on a coordinated way; in particular, payments which offset each other could be settled without requiring any costly liquidity. It should be noted that if the mere submission to a central queue does not have legal implications in terms of settlement (i.e. payments are not settled until perfectly offset), then the settlement risk, which lead to the demise of end-of-day-netting systems, is not re-introduced. Hence, central queues with offsetting does not defeat the purpose of the gross payment modality.

Such central queues are called “liquidity saving mechanisms” (LSMs): allowing to net payments, they permit to save on liquidity. Given the amounts of liquidity circulating in payment systems, these gains may be large. For example: to execute their payments, the banks in the UK CHAPS system borrow from
the Bank of England between 20 and 50 billion Pound Sterling on a daily basis, against pledge of high-quality collateral. And, the argument goes, this collateral may have more profitable uses elsewhere - for example, to collateralize securities clearing, interbank loans, or to generate income from securities lending. From another perspective: for a fixed amount of liquidity, if a payment system adopts an LSM, it is likely to become more resilient, as its liquidity needs may be reduced.

Liquidity saving mechanisms have been on the agenda of policy makers for about a decade, and now many payment system implement different central queuing facilities. There is a vast variety of them, differing in a number of dimensions. For example: how often should the controller look for payment cycles that can be netted? Should the LSM settle only perfectly netting cycles that require no liquidity at all, or should banks have the option of contributing any missing liquidity, thus accelerating settlement? Are submissions to the LSM irrevocable, or can banks retract payments from it, and when? Can individual banks monitor the central queue?

Liquidity saving mechanisms have been studied with simulations for quite long and systems have evaluated the effectiveness of their algorithms before implementation. Leinonen (2005 and 2007) provide collections of such investigations. Johnson et al. (2004) proposes an innovative “receipt reactive” settlement mechanism as an effective LSM. Guenzter et al (1998) Shafransky (2006) develop approximate algorithms for solving the Bank Clearing Problem (i.e. problem of selecting largest subset of payments that can be settled with a given liquidity) from an operations research perspective. Recently McAndrews and Martin (2008) have developed theoretical models on liquidity saving mechanisms incorporating bank behavior, while Galbiati and Soramaki (2008), who study liquidity choices in an agent-based model of a payment system, forms the basis for this paper.

We argue that different LSMs give rises to different “games” between the system’s participants, who will face differently shaped trade-offs between liquidity and delay costs. This paper is a first exploration into these strategic aspects. Our model is very simple, but will contain the essential described above: payments, liquidity recycling, liquidity costs, internal queues, possibly central queues, and delay costs.

We first model a benchmark case: a plain RTGS system where banks choose i) the amounts of liquidity to devote to settlement and ii) how many (and which) payments to hold in internal schedulers. Then, this case is compared to the case where an LSM is available: queued payments are pooled and settled at zero liquidity cost, when possible. Looking at these scenarios, we try and answer the following questions:

1) What is the outcome of a plain RTGS system where banks can internally queue payments? What are banks’ equilibrium liquidity/queuing choices? How do they compare to the choices of a “benevolent planner” which maximizes social welfare?

2) How much liquidity and delays can an LSM reduce in theory? This is a
simple study of the mechanic properties of our LSM.

3) What is the outcome of an RTGS system with an LSM? Is this efficient? How does this compare with the outcome obtained without an LSM?

To anticipate, we find that 1), individual banks underprovide liquidity and queue internally too much compared to what is socially optimal. This is due to the externalities in liquidity / queuing choices mentioned above (see e.g. Angelini 1998). Also 2), when handled by a benevolent planner, an LSM may largely reduce liquidity needs. However, only if the planner is not too exigent in terms of the delays she is prepared to accept. Indeed: if delays must be reduced below a certain level, no payments at all can be queued; and at that point having or not having an LSM is irrelevant\(^1\). Finally 3), for an intermediate range of the liquidity price an LSM may generate two different outcomes. One of them has lower costs than those without LSM: this is a “good” equilibrium. However, the other outcome entails under some parameter values higher costs than those resulting when the LSM is not in place. Interestingly (and against our initial intuition), the “bad” equilibria are those with over-use of the central queue and also higher liquidity usage. These findings suggest clear policy implications: liquidity saving mechanisms are a useful tool, but they need some active management on the part of the system’s controller, or some coordination tool to ensure that banks adopt the low-cost equilibrium.

The paper is organized as follows: Section 2 describes the model; Section 3 solves it, presenting our results. Section 4 concludes.

2 General framework

Our general framework is simple model, which is tweaked in two different ways to describe the two systems we need to compare. The model represents \( N \) banks using a payment system. Banks make choices -to be illustrated later-, that jointly determine the system performance, and thus the banks’ costs or payoffs. The game-theoretic structure of the model is straightforward: we have a one shot, simultaneous-move game, of which we find the Nash equilibria.

As it will be clear from the description in next sections, the model has a dynamic element (“morning”, “then...”, “at the end of the day...”). However, this temporal dimension only pertains to the settlement process, i.e. to the machinery used to derive the banks’ payoffs. In reality, once choices are simultaneously made, payoff are determined in expected value; hence, there is no dynamic interaction in a strategic sense. A main innovation of the paper is the way payoffs are determined: these are numerically generated by an algorithm which mimics a payment system with some realism. We allow banks to exchange hundreds of payments over thousands of time-intervals, generating

\(^1\)In our model, even the ‘plain RTGS stream’ has some features of a central queue. If a bank doesn’t have enough liquidity to settle a payment that it submitted to this stream, the payment is placed in a queue and released immediately as liquidity becomes available from an incoming payment.
complex liquidity flows with “queues”, “gridlocks” and “cascades” (See Beyeler et al (2007) for details on the physical dynamics of this process). We argue this enhances realism by trying to incorporate the complex system internal liquidity dynamics into the payoff function. Summing up, the model is a straightforward game-theoretic representation of a payment system, whose complexity is encapsulated in the payoff function which is computed by means of simulations.

2.1 Payment instruction arrival

During each settlement day \( N \) banks receive payment instructions from some exogenous “clients”. Payment instructions are generated according to a Poisson process with given intensity. Each bank is equally likely to receive the generated payment instruction, and each other bank is equally likely to be the recipient of the payment. So the payment system forms, in a statistical sense, a complete and symmetric network. Each payment has unit value and an urgency drawn from \( U \sim [0,1] \). The urgency parameter reflects the relative importance of settling the payment early: if payment \( r \), with urgency \( u_r \), is delayed by \( t \) time-intervals, it will cause the bank to suffer a cost \( u_r t \).

2.2 Payment settlement

A bank can route each of its payments into either of two streams: i) the RTGS stream or ii) a second stream. Payments submitted into the RTGS stream settle immediately upon submission, but only if the sender bank has enough liquidity. If instead the sender doesn’t have enough liquidity, the payment is queued, and is settled when the sender’s liquidity balance is replenished by an incoming RTGS payment. Upon settlement liquidity is transferred from the payer to the payee. For stream ii) we consider two cases, corresponding to two models.

In the first model, stream ii) is a (rather extreme) representation of bank’s internal queue: payments routed there are delayed for the whole day, and then submitted at once to the RTGS stream. By using this stream (also called LMM - liquidity management mechanism) for low urgency payments, a bank can reserve liquidity for more urgent payments. Routing non-urgent payments in an internal queue is always feasible for a bank, so the two-stream model with ‘RTGS plus LMM’ is our benchmark model.

In the second model, the internal queues (LMM) are replaced by a single central queue (liquidity saving mechanism - LSM), which searches continuously for payments that would form offsetting cycles of any size. To find offsetting payments, we use the Bech and Soramaki (2001) algorithm, which finds an optimal subset of payments to settle, under the constraint that each bank’s payments are settled according to a strict order - here by decreasing urgency. Payments settle in LSM only if they fully offset, so the LSM requires no liquidity. However, any LSM payment that is still queued at the end of the day, is moved into the RTGS stream, and settled there according to RTGS rules.

Our aim is to compare the benchmark system “RTGS-plus-LMM”, with the “RTGS-plus-LSM” system. The first system is a natural benchmark, because
the option of internal queues is always available to banks. The second system
is only a particular example of a dual-stream system. Other LSMs could be
considered, or other rules of interaction between streams could be considered.
We choose the Bech-Soramäki algorithm for its simplicity and the fact that it
ensures an optimal outcome when payments are settled in a strict order, in
out case their urgency. Finally, a three-stream system (with RTGS, LMM and
LSM) is ignored here because, from the perspective of a single bank, LMM is
dominated by LSM: both mechanism require no intraday liquidity, but delays
are weakly longer in LMM. Hence, no bank would use the LMM if LSM is
available.

2.3 The game: choices and costs
At the beginning of the day each bank makes two choices: i) its opening in-
trday liquidity in the RTGS system \( \lambda_i \in [0, \Lambda] \) and ii) an urgency threshold
\( \tau_i \in [0, 1] \). Payment instructions with urgency larger than \( \tau_i \) are settled in the
RTGS system; the others are routed to the second stream (either LMM or LSM
depending on the model)\(^2\). As the urgency parameter is drawn from \( U \sim [0, 1] \),
\( \tau_i \) is also the percentage of payments that bank \( i \) routes into the second stream.

Once banks have chosen their opening liquidity and urgency threshold, set-
tlement of payments takes place mechanically: banks receive payment instruc-
tions, which are submitted according to their urgency. Possibly, delays build up
in each of the two streams.

Costs are 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 initial buffer of liquidity and b) the delay costs which depend
on the delays accumulated during the day. Given a profile of choices \( \sigma = \{\sigma_1, \sigma_2, \ldots, \sigma_N\} \) where \( \sigma_i = (\lambda_i, \tau_i) \) is bank \( i \)'s strategy, the costs borne by \( i \) are:

\[
C_i(\sigma) = a \lambda_i + D_i(\sigma) = a \lambda_i + \sum_r u_r (t_r - t'_r)
\]

where \( a \) is the liquidity price, and \( (t_r - t'_r) \) is the lag between reception and
execution of payment \( r \) (\( u_r \) is the payment's urgency). The dependency of \( C_i \)
on \( \tau_i \) and all other \( \sigma_j \)'s comes through the delays, which indeed depend the \( \tau \)
and \( \lambda \)'s of all banks in the system.

2.4 Equilibrium
The model has \( N \) players, actions \( \lambda_i \) and \( \tau_i \) for each player, and costs/payoffs
determined as described in the above section. We concentrate on the symmetric
equilibria of this game, i.e. on those choices 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 \( (\lambda_i, \tau_i) \) is a best reply to others' choices.

\(^2\)More complex routing rules are conceivable. We restrict attention to this for simplicity.
By restricting attention to symmetric equilibria, we may miss equilibria where banks adopt different, albeit mutually optimal, choices. However, extra-model considerations suggest that such asymmetric equilibria (should they exist at all) would be unlikely to survive in reality. First, if a bank posted less liquidity than its partners, it might be seen to "free-ride", and would be sanctioned in the long run. Second, in real word, banks do not know the choices of each of their counterparties; what they do know is typically some average indicator of the whole system, and this is what the 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. Which confirms that symmetric equilibria are the ones to concentrate on.

3 Results

We illustrate first the mechanics of settlement; that is, we show how delays depend on the banks' choices of liquidity and thresholds in the various streams. Then we illustrate the dependence of costs on the banks' choices - this is the payoff function that banks face in the RTGS-plus-LMM or RTGS-plus-LSM systems. Finally, we report and compare the corresponding equilibria.

All results are obtained by simulating the settlement process for different combinations of the banks' choices. As we look for symmetric Nash-equilibria, we only need run simulations for each combination of “my choices” vs “other’s choices where they do the same”. This reduces the size of the parameter space to \( ([0, \Lambda] \times [0, \tau])^2 \), from \( ([0, \Lambda] \times [0, \tau])^N \). Details on the numerical exploration of the delay and cost function are in Appendix 1.

In most of what follows, we take the point of view of a single bank (referred to as 'I'), facing the rest of the system (referred to as 'Them').

3.1 Settlement mechanics

Total delay costs \( D \) (see Eq.1) accrue from delays in both RTGS and in the second stream (LSM / LMM). We show how these two sources of delays depend on the banks’ choices.

3.1.1 RTGS delays

Figure 1 shows how delay costs in RTGS depend on \( \lambda \) and \( \tau \) when all banks make the same choices (we choose this representation for clarity. In reality "my")
delays depend on four variables: “my” choices of $\lambda$ and $\tau$, and “their” choices of $\lambda$ and $\tau$).

![Figure 1: Delay Costs in RTGS](image)

Obviously, delay costs are reduced by increasing liquidity (unless $\tau = 1$, because then no payment is actually directed into RTGS). And, ‘returns to liquidity’ are decreasing. An increase in the threshold (i.e. less payments being routed to RTGS) increases delay costs for low levels of $\tau$ - the more so the less liquidity is available. 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 ($\tau > 0$). Indeed, if all payments are routed in RTGS, liquidity is absorbed by less urgent payments too, so its ‘returns’ in terms of decreasing delays are reduced.

The relationship between $\tau$ and RTGS delay costs is, in general, non monotonic: when liquidity is scarce, it is not convenient to route too many payments into RTGS: low urgency payments may clog the system, and cause more urgent ones to be unduly delayed. When liquidity is abundant, it is worth to route all payments into RTGS to minimize delays.
3.1.2 Second-stream delays

Delay costs in LMM are simple. Obviously they are independent of $\lambda$, as LMM consumes no liquidity during the day\(^6\). On the other hand, LMM delay costs are a quadratic function of $\tau$. Indeed, every LMM payment settles at the end of the day, so the average time spent in the queue is half a day’s length, i.e. $T/2$. The urgency of each payment is uniformly drawn from $[0, \tau]$, so it is $\tau/2$ on average. Hence, directing a volume of payments $\tau$ into LMM produces delay costs which total $T^2 \tau^2 / 4$.

Delay costs in LSM are also independent of $\lambda$. Simulations show that they are (almost) a linear function of $\tau$. Because the average urgency of a payment in LSM is $\tau/2$, this implies that the average time spent in the LSM scales (almost) with $1/\tau$\(^7\). In a sense, the LSM features increasing returns to scale with respect to processed volumes. The larger the pool of payments from which the algorithm can search for cycles, the more likely these cycles are found. Delay costs in LMM and LSM are compared in Figure 2.

---

\(^6\)Only at the end of the day, queued payments will be 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. Which is exogenous and so independent on banks’ choices.

\(^7\)Delay costs are $D_{LSM} = x^2 \tau$, where $x$ is the average time delayed, $\frac{\tau}{2}$ the average urgency and $\tau$ the volume routed in RTGS. Simulations show that $D_{LSM} \approx \alpha \tau$ so $x \approx \alpha / \tau$. 

---

Figure 2
Delay Costs in LMM and LSM alone
3.1.3 Overall delays

Figure 3 shows how overall delay costs depend on $\lambda$ and $\tau$ in the two systems (for illustrative purposes again we illustrate the case of all banks making the same choice). Overall delay costs can be substantially reduced in the system with LSM (lower surface).

Figure 3
Total delay costs: RTGS-plus-LMM vs RTGS-plus-LSM

Figures 4.1 and 4.2 show the composition of delay costs into its two components: RTGS and second-stream (the four subcharts are for increasing levels of liquidity).
Figure 4.1
Total delay costs: RTGS-plus-LMM

Figure 4.2
Total delay costs: RTGS-plus-LSM
3.2 Equilibria

The reminder of the paper looks at the equilibria reached by the banks in the two systems. A key parameter in the model is the liquidity price \(a\) in Eq.1). This is arguably the variable on which central banks and policy makers have stronger influence. So, we look at how the equilibrium varies when \(a\) changes. An accurate calibration of the model is beyond the scope of this paper. Hence, we let \(a\) vary in a range wide enough that the equilibria span the whole strategy space.

3.2.1 RTGS-plus-LMM

Figure 5 shows how the equilibrium in the RTGS-plus-LSM system vary, when liquidity cost varies (increases, from left to right). Equilibrium choices are represented by a circle; they are to be compared with the choices of a planner who minimizes the costs of the whole system (stars). The background gradient shows system-wide costs; i.e., it shows how worse any \((\lambda, \tau)\) is, compared to the planner’s choice (dark blue = little, dark red = much).
Figure 5
Equilibria of RTGS-plus-LMM
Figure 5 shows that when liquidity price $\alpha$ rises, banks post less liquidity and resort more to internal queues. More importantly, the equilibrium is inefficient: a cost-minimizing planner would provide more liquidity to the system, and would delay less. Equilibrium costs are never less than 15% higher than the social optimum, reaching multiples of it for high liquidity prices. Only for extremely high liquidity costs, the equilibrium coincides with the planner’s optimal choice, both being $\lambda = 0, \tau = 1$.

The reason for this inefficiency are two externalities. On the one hand, a positive externality in liquidity provision: incoming payments to a bank can be ‘recycled’ to make other payments, so liquidity is in a sense a common good (see also Angelini (1998), Galbiati and Soramäki (2008)). Due to this, equilibrium liquidity provision ($\lambda$) falls short of the social optimum. On the other hand, internal queues generate a negative externality: banks have incentives to delay the less urgent payments, to use liquidity for more urgent ones. But, by so doing they slow down the beneficial liquidity recycling in RTGS, harming other banks. Hence, banks queue more than they should from a social perspective -i.e. $\tau$ exceeds what’s chosen by the planner.

It should be noted that the planner’s choice of $\tau$ is of a bang-bang type: either all payments are settled in RTGS, or they are all queued in LMM (until the end of the day).

### 3.2.2 RTGS-\textit{plus}-LSM

Like the LMM, the LSM allows banks to reserve liquidity for urgent payments. However while LMM merely postpones settlement until the end of the day, the LSM allows settlement \textit{without} liquidity. Not only; the LSM reduces settlement time, as its payments are continuously settled whenever offsetting cycles are found. Increased efficiency of the second stream induces banks to use the LSM more intensely, with a reduction in costs. However, increase in $\tau$ also causes a reduction in RTGS volumes, which in turn causes this stream to loose in efficiency. Hence, there is a trade-off between the efficiency levels of the two streams\footnote{This is not the case with an LMM, where average delay times are independent on $\tau$, the volumes queued.}. When "played on" by individual banks, these effects may produce perverse outcomes, as we see next.

When liquidity costs change, the equilibria change essentially as in the RTGS-\textit{plus}-LMM system shown in Figures 5 (above). In particular: i) when $\alpha$ rises, $\lambda$ drops and $\tau$ increases; ii) the equilibrium liquidity falls short of the social optimum, and queued payments are in excess; iii) for very high liquidity costs, both banks and planner use the second stream alone -at which point the equilibrium is efficient; iv) the planner never uses both streams at the same time. However, for an intermediate range of liquidity costs, the RTGS-\textit{plus}-LSM system give rise to multiple equilibria, very different from each other.

Figure 6 is similar to Figure 5, representing the system’s equilibria at various liquidity prices. However, the contour now shows the incentives to deviate from
each \((\lambda, \sigma)\), when all ‘others’ choose \((\lambda, \sigma)\). The Nash equilibria clearly lie where such incentives are zero, i.e. at the bottom of the contour. But, because payoffs are numerically computed, equilibria are difficult to detect when the contour is very flat and low. Indeed, very small but positive incentives to deviate may be indication that that a given \((\lambda, \sigma)\) is not an equilibrium, or may just be an artifact of the finiteness of the grid on which costs are computed (see Appendix for details on the simulations). We thus decide not to look for Nash-equilibria, but for \(\varepsilon\)-equilibria, i.e. strategy profiles from which unilateral deviation yields a gain no larger than a (small\(^9\)) \(\varepsilon\).

Figure 6 then shows that, when \(a\) exceeds a critical point, a corner equilibrium appears, where banks acquire no liquidity and send all payments in the LSM.

\(^9\)We impose that a deviation must not improve payoffs by more than 0.1%. 
Figure 6
Equilibria of RTGS-plus-LSM
As $a$ increases even further, the $\lambda = 0, \tau = 1$ equilibrium persists, but other equilibria emerge, where banks use both streams and some liquidity. Those with low $\lambda$ feature low costs - these are "good" equilibria. The others are "bad". Apart from the corner equilibrium, the bad equilibria are somewhat paradoxical: they feature higher costs, higher liquidity usage ($\lambda$) and higher LSM usage ($\tau$). The existence of such equilibria is probably explained as follows. The LSM features economies of scale (see Sect. 3.1.2), so high usage of it may be self-sustaining. But, as mentioned at the beginning of this section, over-use of the LSM is detrimental to the RTGS stream - which may then need higher amounts of liquidity.

3.2.3 Comparison of the two systems

The key comparison of this paper is between the two systems: RTGS-plus-LMM and RTGS-plus-LSM. With LSM we have 'clouds' of $\varepsilon$-equilibria; hence, for each 'cloud', we pick average values of costs, liquidity and thresholds. We then call 'bad' the cloud with the highest average costs, and 'good' that with the lowest average costs.

Figure 7 then shows the ratio between LMM (red) and LSM (blue) equilibrium values - solid line for 'good cloud' values; dots for the 'bad cloud'. The chart on the left zooms into the left chart; so eg, when $a = 13$, the good LSM equilibria are about 5% cheaper than the LMM equilibrium. Savings become more sizable for higher liquidity costs.

Figure 7  
Equilibria: comparison
4 Conclusions

This paper offers a parsimonious model of two payment systems: one where payments can be internally queued, and one where a liquidity saving mechanism (LSM) is available. The LSM offers two benefits: non-urgent payments can be queued there, reserving liquidity for more urgent payments -but this is also a feature of internal queues (LMM). The additional convenience of the LSM is that payments there can be offset in cycles at no liquidity cost and settlement may take place very soon (as soon as matching payments reach the central queue).

As expected, an LSM can bring about benefits compared to internal queues. However, the high ‘mechanical’ advantages of a central queue might be mitigated by strategic behaviour: there are perverse equilibria with high liquidity usage, intense use of the LSM, and yet costs that exceed those that obtain in a system without the LSM.

These finding suggest that liquidity saving mechanism are useful tools, which however may need some coordination device to ensure that banks arrive at the good equilibrium. A necessary caveat is that these findings are for one particular liquidity saving mechanism, compared to one rather extreme model of internal queues. Other LSMs, perhaps associated while other settlement rules, may yield different outcomes.
5 Appendix I

To compute the payoff function of bank $i$ (eq. 1), we need to find the delays experienced by $i$ when the rest of the system chooses $\{(\lambda_j\tau_j)\}_{j \neq i}$. As mentioned on pg. 3, we can treat the 'rest of the system' as one player, and assign to it symmetric action profiles $(\lambda_j\tau_j) = \{(\lambda_1\tau_1), (\lambda_2\tau_2), \ldots\}$ such that $(\lambda_1\tau_1) = (\lambda_2\tau_2) = \ldots$. This greatly reduces the action profiles to explore, because then delays $D_i((\lambda_i\tau_i), (\lambda_i\tau_j))$ are then a function of 4 variables only. We compute them as follows.

We run simulations for a restricted number of 2-player action profiles. In particular, we simulate the settlement process for $\lambda$ taking on all integers in $[0, 10]$, and $\tau$ any number in $[0, 0.2, 0.4, \ldots, 1]$. That is, we compute $(11 \times 6)^2 = 4356$ values of the delay function, for just as many action profiles. To do so, because payment orders arrive in a random order, we need to simulate at least 200 'days' for each action profile to obtain a reliable estimate of the 'average day'. Hence, we simulate $200 \times (11 \times 6)^2 = 871'200$ days in total.

Yet, $11 \times 6 = 66$ choices for each bank are not enough to obtain 'smooth' results: when computing the equilibria, undesired artifact emerge. Hence, we numerically smoothen and interpolate the delay function $D_i((\lambda_i\tau_i), (\lambda_i\tau_j))$ on a refined grid, a 4-dimensional cube with $41^4 = 2^{825'761}$ points, which correspond to banks choosing $\lambda$ in $[0, 10]$ in steps of 0.25 (41 liquidity levels) and $\tau$ 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, we obtain the the cost, or payoff function defined in eq. 1). Using such payoff function, equilibria are computed -numerically, of course.
References


