



---

# BANK OF FINLAND DISCUSSION PAPERS

---

9 • 2001

Morten L. Bech – Kimmo Soramäki

Financial Markets Department

13.6.2001

## Gridlock Resolution in Interbank Payment Systems

Suomen Pankin keskustelualoitteita  
Finlands Banks diskussionsunderlag

---

---

# BANK OF FINLAND DISCUSSION PAPERS

---

9 • 2001

Morten L. Bech – Kimmo Soramäki

Financial Markets Department

13.6.2001

## Gridlock Resolution in Interbank Payment Systems

The views expressed here are those of the authors and do not necessarily represent those of the Bank of Finland or Danmarks Nationalbank.

Morten L. Bech, Danmarks Nationalbank, Copenhagen.

Kimmo Soramäki, Bank of Finland, Helsinki

The authors wish to thank Harry Leinonen, Karsten Biloft, Heikki Koskenkylä, Anita Gantner, Esa Jokivuolle and Jesper Berg for their helpful comments and suggestions.

Suomen Pankin keskustelualoitteita  
Finlands Banks diskussionsunderlag

---

ISBN 951-686-719-7  
ISSN 0785-3572  
(print)

ISBN 951-686-720-0  
ISSN 1456-6184  
(online)

Suomen Pankin monistuskeskus  
Helsinki 2001

# Gridlock Resolution in Interbank Payment Systems

Bank of Finland Discussion Papers 9/2001

Morten L. Bech – Kimmo Soramäki  
Financial Markets Department

## Abstract

The paper analyses the severity of gridlocks in interbank payment systems operating on a real time basis and evaluates by means of simulations the merits of a gridlock resolution algorithm. Data used in the simulations consist of actual payments settled in the Danish and Finnish RTGS systems. The algorithm is found to be applicable to a real time environment and effective in reducing queuing in the systems at all levels of liquidity, but in particular when intra-day liquidity is scarce.

Key words: payment systems, liquidity, gridlock, algorithms, settlement

# Takalukkojen purku pankkienvälisissä maksujärjestelmissä

Suomen Pankin keskustelualoitteita 9/2001

Morten L. Bech – Kimmo Soramäki  
Rahoitusmarkkinaosasto

## Tiivistelmä

Tässä tutkimuksessa analysoidaan pankkien välisissä ajantasaisesti toimivissa maksujärjestelmissä esiintyvien ns. gridlock-tilanteiden (maksujen lukkiutumien) syntymisen vakavuutta sekä arvioidaan simulointien avulla tutkimuksessa esitetyn gridlock-tilanteiden purkualgoritmin tehokkuutta. Simulointiaineistona käytetään Suomen ja Tanskan keskuspankkien RTGS-järjestelmistä kerättyjä maksutapahtumatietoja. Algoritmi havaitaan käyttökelpoiseksi reaaliaikaisessa ympäristössä sekä tehokkaaksi gridlock-tilanteiden purkamisessa kaikilla likviditeettitasoilla ja erityisesti tilanteissa, joissa pankkien päivänsisäinen likviditeettitilanne on kireä.

Asiasanat: maksujärjestelmä, likviditeetti, gridlock, algoritmit, katteensiirto

# Contents

Abstract	3
Tiivistelmä	4
1 Introduction	7
2 Gridlocks in RTGS systems	9
2.1 The causes for gridlocks	9
2.2 Problem formulation	10
2.3 Solution to the problem	13
3 Gridlock resolution	15
3.1 Discrete optimisation problem	15
3.2 Implication of the sequence constraint	16
3.3 Solution algorithm	18
4 Simulations	19
4.1 Simulation setting	19
4.2 Payment data	20
5 Results	23
5.1 The lower and upper bond	23
5.2 The trade-off between liquidity and delay	23
5.3 Gridlocks experienced in the system	27
6 Conclusions	31
References	32
Appendix 1.	33

# 1 Introduction

The settlement of interbank liabilities has traditionally taken place in interbank payment systems operated by central banks. Many central banks in the industrialized world have during the last decade built Real-Time Gross Settlement (RTGS)<sup>1</sup> systems for the settlement of interbank funds transfers. In an RTGS system payments are processed individually with finality throughout the day and these systems are primarily used for time critical and/or large value payments due to the low settlement risk<sup>2</sup>.

The volume of interbank payments increased dramatically throughout the 1980s and 1990s as a result of rapid financial innovation and globalization of financial markets. At the same time the developments in information technology made gross settlement in real time a technologically feasible alternative. Historically, interbank payments have been settled via end of day netting systems, but as the volumes of funds transfers increased, central banks became concerned about the systemic risk<sup>3</sup> stemming from netting systems. In netting systems where payments are credited to customer accounts before final interbank settlement, a failure of one participant may cause the failure of other participants if proper risk controls are not in place. Gross settlement in real time eliminated the settlement risk, reduced the systemic risk and became the prevalent option chosen by most central banks.

However, the risks were traded off against an increased need for intra-day liquidity (i.e. funds available for settlement). Central banks have found that the provision of free uncollateralized intra-day liquidity is not a viable solution to the liquidity need since it merely results in a transfer of credit risk to the central bank. As a result intra-day liquidity is provided by central banks through an overdraft facility either subject to interest or backed up by collateral. Liquidity is thus costly either in form of an explicit fee or implicitly in the form of the opportunity cost of the pledged collateral.

BIS (1993) defines a gridlock as a “situation that can arise in a funds or securities transfer system in which the failure of some transfer instructions to be executed (because the necessary funds or securities balances are unavailable) pre-

---

<sup>1</sup> For descriptions on RTGS systems please refer to BIS (1997)

<sup>2</sup> For the ease of exposition the terms bank and central bank are used as short hands for a participant in and the operator of the interbank payment system, respectively. In many countries participants include non-banks and the RTGS system is designed and operated in close co-operation between the central bank and the banking community.

<sup>3</sup> The risk that a failure by one participants in a transfer system to meet its obligations causes other participants to be unable to meet their obligations, possibly threatening the stability of the financial system (BIS 1993)

vents a substantial number of other instructions from other participants from being executed". We use a slightly different definition for gridlocks and define them as situations where the system's inability to settle any queued payments is not due to the lack of liquidity per se but can be attributed to the requirement that payments be settled one at a time. We shall refer to situations where the lack of liquidity prevents settlement as deadlocks. Allowing for simultaneous settlement can solve gridlocks whereas deadlocks can only be solved by the infusion of additional liquidity (the scarce resource) in the system.

In this article we analyze the severity of gridlocks in an interbank payments system operating on a real-time basis, describe ways to avoid and resolve gridlocks and analyze by means of simulations the merits of a specific gridlock resolution algorithm applicable to an RTGS environment.

The paper is organized as follows: In section two we model the problem and provide an operational definition of gridlocks. In section three we present an algorithm applicable to a real-time environment and discuss its properties. In section four we define the simulation setting and the indicators that are used to analyze the effects of the algorithm, as well as describe the payment data used in the simulations. In section five we present our results. Section six concludes the paper.



## 2 Gridlocks in RTGS systems

### 2.1 The causes for gridlocks

Banks manage their liquidity throughout the day in order to minimize the cost of settling obligations on behalf of their customers or obligations resulting from their own treasury operations. Depending on the time-criticality of the payments, banks will at least occasionally have an incentive to hold less liquidity on their settlement accounts than what is needed for immediate settlement of all obligations. In many systems banks also delay payments in their internal systems, instead of forwarding them into the central queuing facility. Because incoming payments are the source of liquidity with the lowest cost, banks have an incentive to settle their outgoing payments only after they have received liquidity from incoming payments. While it may be optimal for the individual banks to hold less liquidity than what is needed for immediate settlement of all payments or delay payments in their internal systems, it is not necessarily optimal at the system level.

The delay in payment settlement caused by insufficient liquidity has a cost for both the sender and the receiver. For the sender these costs may be implicit, in the form of deterioration in customer service, or explicit, in the form of sanctions. For the receiver the cost is either the cost of not receiving the liquidity (which might force it to acquire more costly liquidity in order to settle its pending payments) or the cost of having to delay its own payments. If the receiver has to delay its payments it faces the same costs as the sender of the first payment. Likewise the receiver of this second payment faces the same types of costs as the receiver of the first payment. This way the costs are cumulated forward in the system until a bank acquires enough liquidity to settle its pending payment. This negative externality creates a dead-weight losses at system level (Angelini 1998).

Dead-weight losses occur when payments are delayed, either in a central queue or in the banks' internal systems for the purpose of saving liquidity. When payments are delayed, the system faces the risk of gridlock, which can add substantially to the dead-weight losses experienced by the system.

Currently the cost of liquidity, at least in Europe, is relatively low and as a consequence the payments are settled smoothly. Intraday liquidity is provided by the central banks against full collateralisation and no fees or interest is charged on the amount used. The result is that queues are not a major issue on the daily level. However, should money market disturbances increase the opportunity cost of collateral or disrupt the availability of collateral in general, features that optimize on the liquidity used for settlement of payments might very well become appreciated. For systems where the lack of liquidity is an ongoing concern, gridlock resolution naturally provides the greatest benefits.

Central bank policy goals in payment systems mainly address the smooth functioning of the system, including efficiency, and the control of risks, especially the systemic risk. Efficient resolution of gridlocks enhances the smooth functioning of the system and reduces the liquidity risk and cost by effecting faster settlement of payments. Analogously, it will reduce the costs of settlement at a given level of delays, by enabling the banks to hold lower balances on the settlement account and/or incur smaller overdrafts. In order to discourage banks from using incoming payments as their only or main source of liquidity, agreements among participants to process a certain share of payments prior to some time of the day are in place in many countries. Also, differential pricing of payment processing can create an incentive structure that contributes to a smoother settlement of payments. Incentives in payment settlement have been covered *inter alia* by Angelini (1998), Kahn and Roberds (1998), McAndrews and Rajan (2000) and Bech and Garrat (2001). This article concentrates on resolving gridlocks in an environment where payments are sent to the central queue, i.e. where information on all queued transfers and their preferable order of settlement is available centrally.

## 2.2 Problem formulation

We study a system with a centrally located queue. Banks transfer funds to each other continuously throughout the day and the settlement of these funds takes place when the sending bank's account is debited and the receiving bank's account credited. As long as the sending bank's account balance (including any possible overdraft limit) is equal to or exceeds the value of a payment, settlement takes place immediately. If the balance of the account is not sufficient to cover the payment, the payment is put in the centrally located queue. Queued payments are released according to a scheme predefined by the bank itself. The first pending payment in the queue is released as soon as the bank has accumulated enough liquidity to cover the payment.<sup>4</sup>

Even though it might not be possible to individually settle the first pending payment of any bank due to lack of funds it might be possible to simultaneously settle a subset or all of the payments queued. Currently, a range of different measures have been put in place to ensure the smooth settlement of payments, mainly based on the netting of queued transfers which is either invoked by the central

---

<sup>4</sup> The source of liquidity is not explicitly modeled here. In general there are four sources of funds: balances maintained with the central bank, credit extensions based on pledge or repo transaction with the central bank, operations with other banks through the money market, and incoming payments from other banks not related to money market operations. In most RTGS systems the incoming payments are a major source of liquidity, e.g. in TARGET as a whole these account for some 70% of the gross liquidity need.

bank when needed, or at predefined intervals. A necessary requirement for all gridlock resolution features is that the central bank has information concerning all incoming and outgoing payments. In practice, this will require a central queuing mechanism in the RTGS system.

The objective of gridlock resolution is to identify and simultaneously settle the largest possible subset of the payments in queue subject to any constraints in the design of the system. We use a formulation with two of the most common constraints, the *liquidity* and the *sequence* constraint. The liquidity constraint states that the liquidity position of the bank can never fall below a pre-defined limit. The sequence constraint states that the payments have to be settled in a pre-defined order (in practice usually by the first-in first-out principle)

In order to formally define a gridlock and the gridlock resolution problem we introduce some notation. Assume that we have  $n$  banks indexed by  $i$ . Let  $Q_i$  be the set of queued payments of bank  $i$  and let  $Q = \cup_{i=1}^n Q_i$  be the set of all queued payments. Similarly, let  $X_i$  denote the subset of queued payment of bank  $i$  to be settled simultaneously and  $X = \cup_{i=1}^n X_i$ . The ex ante balance and the ex post balance of bank  $i$  is given by  $\bar{B}_i$  and  $B_i(\cdot)$ , respectively. The total amount of outgoing payments from the queue of bank  $i$  is  $S(X_i)$  and the total amount of incoming payments to bank  $i$  from the queues of all other banks is  $R(X_{-i})$ , where  $-i$  denotes all banks except bank  $i$ .

Furthermore, let the sequence relation  $\succ_i$  be a complete and transitive settlement order on  $Q_i$  chosen by bank  $i$ , i.e.  $\succ_i$  provides a ranking of the payments in terms of in which sequence they should be settled.

We define a gridlock in the present context as follows:

**Definition 1** (*Gridlock*)

A gridlock is a situation where  $Q \neq \emptyset$  and there exists a nonempty  $X \subseteq Q$  such that if the payments in  $X$  were settled simultaneously then

$$B_i(\bar{B}_i, X) = \bar{B}_i - S(X_i) + R(X_{-i}) \geq 0, \quad \text{for } i = 1, \dots, n \quad (1)$$

and

$$\forall x \in X_i \nexists q \in Q_i \setminus X_i \text{ such that } q \succ_i x, \quad \text{for } i = 1, \dots, n \quad (2)$$

The first condition, *the liquidity constraint*, stipulates that if the payments in  $X$  were simultaneously settled then the ex post balance  $B_i(\cdot)$  of each bank has to be non-negative. The ex post balance is equal to the ex ante balance,  $\bar{B}_i$  (including any intra-day day credit line from the central bank), minus the total amount of

payments sent by the bank,  $S(X_i)$ , plus the amount of payments received,  $R(X_{-i})$ .

The second condition, *the sequence constraint*, states that the ordering of the payments,  $\succ_i$ , on the set of queued payments for each bank must be adhered by  $X$ . The ordering encompasses all arrangements according to which payments are released from the queue and it can be unique for each bank. The rationale for the sequence constraint stems from two facts. Firstly, intra-day liquidity management in a real time environment is a highly complicated task and many banks are thus reluctant to let a third party control the sequence in which payments are released from their queue. Secondly, central banks are equally unlikely to accept this role due to the various legal issues that might arise if a time critical payment was not settled in due time. In practice most centrally located queuing arrangements have adopted variants of the FIFO ("First in, first out") rule. However, all that is required for the results derived below is that the sequence by which payments can be released from a queue is fixed and predetermined from the viewpoint of the system. As we shall see this assumption has profound implication for the solution of the problem.

Further on, we define deadlock as follows:

**Definition 2 (Deadlock)**

*A deadlock is a situation where  $Q \neq \emptyset$  and  $X$  (as defined in definition 1) is empty i.e.  $X = \emptyset$*

A deadlock is a stalemate of payments between banks, where the payments cannot be settled by any means without infringing upon the sequence constraint. A deadlock is only resolved by the addition of adequate liquidity in the system or by the addition of payments in the queues, so that the inclusion of these payments turns the situation into a gridlock.

The objective of the gridlock resolution is either assumed to be maximization of the number or the value of payments settled<sup>5</sup>. If the objective is to maximize the number of payments we shall denote the objective by  $N(X)$  and if the objective is to maximize the value of payments we shall use  $V(X)$ . Let  $O(X)$  be the generic objective of the central bank.

We define the gridlock resolution problem (GRP) as

---

<sup>5</sup> In theory, the objective could be more elaborate, e.g. to maximize a weighted mixture of the number and the value of payments settled or even other arguments.

**Definition 3** (*Gridlock Resolution Problem, GRP*)

The gridlock resolution problem is  $\max_{X \subseteq Q} O(X)$  s.t. the liquidity constraint given in (1) and the sequence constraint given in (2)

In addition to either of the two objectives the gridlock resolution mechanism should also satisfy at least some of the following criteria if it is to be implemented in practice.

(Fairness) The gridlock resolution mechanism should be neutral in the sense that the solution does not favor one or more banks relative to others.

(Computation Time) In a real-time environment where the settlement of payments cannot be suspended for an extended period of time; the time needed to find a solution should be very low.

(Legal Risk) The gridlock resolution mechanism should not expose the central bank, the service provider or the participants to any significant legal risks.

These properties in relation to the proposed gridlock resolution mechanism will be discussed later in the paper and in the appendix.

## 2.3 Solution to the problem

In general, the queue  $Q$  can be in one of three states:

- empty,** in which case gridlock resolution is not relevant,
- gridlocked,** in which case some or all payments can be settled simultaneously subject to the constraints discussed above or
- deadlocked,** in which case no payments can be settled.

The state of the queue is unobservable *ex ante* but observable *ex post*. Prior to gridlock resolution only the queue,  $Q$ , can be observed, but not the state. There are three possible types of solutions to the problem. The solution  $X$  can either be

- null ( $X = \emptyset$ ), in which case the queue was **deadlocked** *ex ante* and remains **deadlocked** *ex post*,
- partial ( $X \subset Q$ ), in which case the queue was **gridlocked** *ex ante* and the remaining part of the queue  $Q \setminus X$  is **deadlocked** *ex post* or
- full ( $X = Q$ ), in which case the queue was **gridlocked** *ex ante* and is **empty** *ex post*.

The relationship between the state of the queue and solutions to the GRP is summarized in Table 1.

**Table 1. State of Queue**

<b>Ex ante (unknown)</b>		<b>Gridlock Resolution</b>		<b>Ex post (known)</b>	
State	Size	Solution	Size	State	Size
Deadlocked	$Q$	Null	$X = \emptyset$	Deadlocked	$Q$
Gridlocked	$Q$	Partial	$X \subset Q$	Deadlocked	$Q \setminus X$
Gridlocked	$Q$	Full	$X = Q$	Empty	$\emptyset$

## 3 Gridlock resolution

### 3.1 Discrete optimisation problem

The problem of gridlock resolution can conveniently be modeled as a discrete optimization problem. The queue of the  $i$ th bank  $Q_i$  contains  $m_i$  payments and the sequence relation,  $\succ_i$ , provides a strict ordering of these payments from 1 to  $m_i$ . Let  $A$  represent the range of permissible values for payments set by the system<sup>6</sup>. The  $k$ th payment in the queue of bank  $i$ , consists of three elements

- (i) the amount,  $a_{i,k} \in A \subset \mathbf{R}_+$ ,
- (ii) the receiver of the payment,  $r_{i,k} \in \{1, 2, \dots, n\} \setminus \{i\}$  and
- (iii) an indicator of whether the payment is part of the solution to the GRP,  $x_{i,k} \in \{0, 1\}$ .

Let  $X = \{x_1, x_2, \dots, x_n\}$  be a set of indicator vectors - one for each bank of dimension  $m_i$ . The GRP can, depending on whether the value or number of payments settled is maximized, be written as

$$\max_X \begin{cases} V(X) = \sum_{i=1}^n \sum_{k=1}^{m_i} a_{i,k} x_{i,k} \\ N(X) = \sum_{i=1}^n \sum_{k=1}^{m_i} x_{i,k} \end{cases} \quad (3)$$

s.t.

$$B_i(\bar{B}_i, X) = \bar{B}_i - S(x_i) + R(X_{-i}) \geq 0, \quad \text{for } i = 1, \dots, n$$

$$x_{i,k+1} \leq x_{i,k}, \quad \text{for } i = 1, \dots, n \text{ and } k = 1, \dots, m_i - 1,$$

where  $S(x_i) = \sum_{k=1}^{m_i} a_{i,k} x_{i,k}$ ,  $R(X_{-i}) = \sum_{j=1}^n \sum_{k=1}^{m_j} a_{j,k} x_{j,k} I(r_{j,k} = i)$  and  $I(\cdot)$  is the indicator function, i.e.  $I(E)$  equals one if the event  $E$  is true and zero otherwise.

Solving a discrete optimization problem of this kind usually involves an algorithm to direct the search through the set of feasible candidates. A vast literature is available on optimal algorithm design. The related problem of clearing or netting of interbank payments is analyzed in Guntzer et al (1998). They refer to the problem of maximizing  $V(\cdot)$  subject to the liquidity constraint as the *Bank Clearing Problem* (BCP).

---

<sup>6</sup> An example is the Fedwire system, where the maximum value for securities transfers is set to \$50 million

**Definition 3** (*Bank Clearing Problem*)

The Bank Clearing Problem (for the whole system) is

$$\begin{aligned} & \max_X V(X) \\ & \text{s.t.} \\ & B_i(\bar{B}_i, X) = \bar{B}_i - S(x_i) + R(X_{-i}) \geq 0, \quad \text{for } i = 1, \dots, n \end{aligned} \tag{4}$$

where  $V(\cdot)$ ,  $S(\cdot)$  and  $R(\cdot)$  are as defined in equation (3).

The BCP is NP complete<sup>7</sup>. Heuristically speaking, the NP completeness of the problem implies that it is not possible to find a solution algorithm which is guaranteed to find the optimal solution, and at the same time have a computation time which does not grow exponentially as the size of the problem grows. However, while the BCP in theory is very demanding it can approximately be solved using algorithms proposed in Guntzer et al (1998). In summary, solutions to the BCP are not guaranteed to have a computation time that will not suspend the settlement of payments for an extended period of time. Furthermore, if the solution found is only an approximation to the optimal solution with some probability, then the gridlock resolution is unlikely to meet the fairness criterion described above, since at least one bank is bound to prefer the true - albeit unknown - optimal solution.

### 3.2 Implication of the sequence constraint

The GRP imposes an additional constraint on the optimization compared to the BCP. The addition of the sequence constraint to the BCP implies the following:

- The solution to the GRP is going to yield a value of the objective function, which is less or equal to the solution to the BCP.
- A reduction in the (time) complexity of the problem. Let  $h$  be the sum of all payments in queue i.e.  $h = \sum_{i=1}^n m_i$ . A thorough search of all the possible solutions requires  $2^h$  combinations to be checked in the BCP whereas only  $h$  combinations have to be checked in the GRP case. The GRP is not NP complete. In practice, the reduction in computation time when comparing the BCP and the GRP will depend on (a) the number of payments in queue of

---

<sup>7</sup> For a discussion of NP complete problems see Horowitz, Ellis and Sahni (1978). Famous NP complete problems include *the travelling salesman* and *the knapsack* problem



- each bank,  $m_i$ , (b) the number of banks  $n$  itself and (c) on the actual values of the payments.
- The solution to the GRP is the optimal one for sure instead of only an approximate solution with some positive probability.
  - The optimal solution is going to be indifferent to whether the objective is the maximization of value or number of payments settled. A proof is provided in the appendix.

There is a clear trade-off between the wish to settle the largest subset of payments possible on the one hand and to infringe on the control over the liquidity management process of the individual banks on the other hand. The solution to the GRP implies the set of desirable properties, listed in section 2.2 before, which are not assured by a solution to the BCP, but comes at the cost of a “smaller” set of payments being settled.

The fact that the solution of the GRP is indifferent to whether the objective function is  $V(\cdot)$  or  $N(\cdot)$  allows us to write the maximization problem more compactly in terms of just the number of payments,  $l_i$ , to be picked from the queue of each bank. Let  $l = (l_1, l_2, \dots, l_n)$  and  $m = (m_1, m_2, \dots, m_n)$ . Using the same functional symbols we rewrite the problem as follows:

$$\begin{aligned} & \max_{0 \leq l \leq m} \sum_{i=1}^n l_i \\ & \text{s.t.} \\ & B_i(\bar{B}_i, l) = \bar{B}_i - S(l_i) + R(l_{-i}) \geq 0, \quad \text{for } i = 1, \dots, n \end{aligned} \tag{5}$$

and

$$\begin{aligned} S(l_i) &= \sum_{k=1}^{l_i} a_{i,k} \\ R(l_{-i}) &= \sum_{j=1}^n \sum_{k=1}^{l_j} a_{j,k} I(r_{j,k} = i) \end{aligned} \tag{6}$$

The objective is taken to be the number of payments and the indicators,  $x_{i,k}$ , have been replaced by sums over  $i = 1, \dots, l_i$ . The sequence constraint has been subsumed in the problem.

### 3.3 Solution algorithm

A simple algorithm can solve the GRP given by equation (5) and (6)<sup>8</sup>. The algorithm starts by including all payments in the solution and removes payments one by one from deficient banks balance until either all banks have a positive ex post balance or all payments have been removed from the initial solution. Formally, the algorithm goes as follows:

- Step 1: Include all queued payments in the solution i.e.  $l_i^* = m_i \forall i = 1, \dots, n$
- Step 2: Calculate  $B_i(\bar{B}_i, l^*) \forall i = 1, \dots, n$   
If  $B_j(\bar{B}_j, l^*) < 0$  for some  $j$  then execute step 3.  
If  $B_i(\bar{B}_i, l^*) > 0$  for all  $i = 1, \dots, n$  then stop.
- Step 3: Choose any  $j$  such that  $B_j(\bar{B}_j, l^*) < 0$  and remove from the last payment in queue for this bank from the solution. That is  $l_j^* = l_j^* - 1$ . Repeat step 2.<sup>9</sup>

The algorithm always finds the optimal solution (which might be empty) and is fair across banks. The properties of the algorithm and the solution found are discussed in more detail in the appendix.

---

<sup>8</sup> The algorithm was developed by Danmarks Nationalbank in co-operation with Department of Mathematical Modelling at The Technical University of Denmark as part of the KRONOS project. KRONOS is the new Danish RTGS system.

<sup>9</sup> A perhaps, somewhat counterintuitive fact is that the choice of which of the deficient banks to remove a payment from does not influence the final solution (see appendix).

## 4 Simulations

### 4.1 Simulation setting

The simulations consist of two scenarios: one where no gridlock resolution is used and one where the algorithm presented in the previous section is applied. We calculate two indicators for both scenarios: the amount of liquidity used for settlement and an indicator that measures the delays in settlement on the system level.

The amount of liquidity available in the system affects the number and duration of gridlocks. If enough liquidity for each participant to settle their payments immediately is available, naturally no gridlocks occur. We will refer to this amount of liquidity for each bank  $i$  as the upper bound  $UB_i$ . On the other extreme all banks might have just enough liquidity to settle all the days payments until the end of the day. We shall refer to this amount of liquidity as the lower bound of liquidity  $LB_i$ . Let  $d_i$  denote the total number of payments send by bank  $i$  over the course of the business day. The  $k$ th payment of the  $i$ th bank,  $p_{i,k}$  consists of three elements

- (i) the amount,  $a_{i,k} \in A \subset \mathbb{R}_+$ ,
- (ii) the receiver of the payment,  $r_{i,k} \in \{1, 2, \dots, n\} \setminus \{i\}$  and
- (iii) a time stamp  $t_{i,k}$  of when the payment was sent.

The lower bound of liquidity for the  $i$ th bank can be written as

$$LB_i = \max(0, \sum_{j=1}^n \sum_{k=1}^{d_j} a_{j,k} I(r_{j,k} = i) - \sum_{k=1}^{d_i} a_{i,k}) \quad (7)$$

where the first sum is the value of payments received and the second sum is the value of the payments send over the course of the business day by bank  $i$ .

If the value of payments received during the day is larger than the value of payments sent, a bank only needs to use the liquidity it receives in the form of incoming payments for settling its own payments and thus the lower bound equals zero. If the value of payments sent exceeds the value of payments received, the difference has to be available at least at the end of the day.

All simulations with and without application of the gridlock resolution algorithm were run with six different levels of liquidity,

$$L_i = \alpha(UB_i - LB_i) \quad (8)$$

where  $\alpha = \{0, 0.2, 0.4, 0.6, 0.8, 1\}$ . It was assumed that banks could not post or withdraw liquidity from the system during the day.

The settlement delay for each payment was calculated as the time difference between payment initiation by the sending bank and final settlement of the payment, when the business day was split into time intervals of one minute. The delay indicator,  $\rho_i$ , for each bank was then calculated as the ratio of the cumulated sum of queued payments to the cumulated sum of outgoing payments up to the end of the day

$$\rho_i = \frac{\sum_{s=1}^T \sum_{k=1}^{d_j} a_{i,k} I(p_{i,k} \in Q(s))}{\sum_{s=1}^T \sum_{k=1}^{d_j} a_{i,k} I(t_{i,k} < s)} \quad (9)$$

where  $T$  is the number of time intervals and  $Q(s)$  is the queue of bank  $i$  at time  $s$ . If all payments are settled immediately upon entry into the system,  $\rho_i = 0$ , and if all payments are delayed until the end of the day then  $\rho_i = 1$ . The system level  $\rho$  is a weighted average of the individual bank level  $\rho_i$ 's, the weights being the value of payments settled by each bank during the day.

In our definition of the GRP we are implicitly assuming that payments are settled in whole value. Koponen and Soramäki (1998) and Leinonen and Soramäki (1999) find, in a similar set up, that splitting of payments can substantially reduce gridlocks without increasing liquidity needs but also point to the potential legal caveats. It must be noted that this indivisibility assumption does not preclude payments from being split prior to the application of the resolution mechanism, it merely states that payments to be settled must have predefined values (i.e. splitting is possible by adjusting the range of permissible values,  $A$ , as explained in 3.1).

## 4.2 Payment data

The simulations were run with both data from the Finnish BoF-RTGS system and the Danish Inquiry and Transfer System. BoF-RTGS is the Finnish national RTGS system of TARGET and operates in euro. The Danish Inquiry and Transfer System is the central bank RTGS system for the Danish krona and is not linked to TARGET. The Finnish system is thus an 'open' system with liquidity flows to and from the system as the Danish system is a 'closed' system with liquidity flows only between participants in the system. The Danish data consists of 64 days of transactions processed in the system during the last three months of 1999. Data extrapolated from the Finnish BoF-RTGS system consists of the last 100 days of year 2000. Key figures concerning both systems are summarized in Table 2 below. The systems simulated were besides the opening hours identical for both sets of data.

**Table 2. Key figures concerning payment flows in both systems (mil euro)**

	DN Inquiry and Transfer System (DK)			BoF-RTGS (FI)		
	Minimum	Maximum	Average	Minimum	Maximum	Average
Individual payment value	0.001	1 227	10	0.001	2 098	10
Daily payment flow (value)	1 358	13 783	9 352	4 638	32 718	15 045
Daily payment flow (number)	490	2 342	925	558	1 872	1 428

### DN Inquiry and Transfer System

During the analyzed period 146 account holders sent or received payments in the DN Inquiry and Transfer System. The daily turnover of the system ranged from 10 to 103 billion kronor (1.4 billion to 13.8 billion euro) with an average of 63 billion kronor (9.4 billion euro) per day. The number of payments processed in the system ranged between 490 and 2342 with an average of 925 payment per day. The average number of payments sent on a day by a participant was only 12 and the average value processed 869 million kronor (121 million euro). However, the system is highly concentrated; in terms of value the three largest banks account for almost half and the ten largest banks for almost 90% of the all of payments processed in the system. The figures do not include transfers related to securities settlement. These transfers were removed from the data since they were not considered to constitute an actual payment but rather a reservation of liquidity.

### BoF-RTGS

The BoF-RTGS system had 13 participants during the simulated period, and thus in terms of account holders the system is substantially smaller than the Danish system. In Finland, the Central Association of the Finnish Co-operative Banks (Osuuspankkikeskus, OPK) and the Finnish Savings Banks' Association (Säästöpankkiliitto) function as central credit institutions for their member banks, which reduces the number of direct participants.

The daily turnover of the system ranged from 4.6 billion euro to 32.7 billion euro with an average of 15 billion euro per day. Some 32% of these were incoming TARGET transfers. The number of payments processed ranged between 558 and 1872 with an average of 1428 payments per day. These figures do not include payments related to PMJ (Banks Clearing System)<sup>10</sup> transfers, payments related to

---

<sup>10</sup> PMJ is a system for retail payments with net settlement twice a day. The bilateral net transfers are effected through BoF-RTGS.

the maintenance of the currency supply, and TARGET payments from and to the Bank of Finland, as these were removed from the data for the purpose of the simulations. The value of payments settled the BoF-RTGS was thus somewhat higher than the value of payments settled in the DN Inquiry and Transfer System. The payment flows in BoF-RTGS are very concentrated; in terms of value the three largest banks account for almost two thirds of payments processed in the system.

The low turnover days in both systems can be attributed to low market activities during Christmas and New Year. Both sets of data were extrapolated from the central bank accounting systems and as such the timing of payments does not necessarily reflect the timing of payments in another environment. In the Danish system, no central queuing facility is available, and as a result banks manage their liquidity in their internal systems by delaying payments before forwarding them into the central bank system. These decisions are bound to be different when their liquidity holdings are reduced or increased or when the system characteristics are changed. To some degree this is the case in BoF-RTGS as well, in spite of the availability of a central queuing facility. The simulations presented in the next section do not take the possible responses to system and liquidity changes into account, and merely show the effects of changes in liquidity and the use of the gridlock resolution algorithm in an environment where all other aspects are kept as they are. As such the results might overestimate queuing when liquidity is reduced in the system, both in simulations with and without the use of gridlock resolution.

## 5 Results

### 5.1 The lower and upper bound

The systems were simulated at six different levels of liquidity ranging from the lower to the upper bound. The lower bound is the liquidity needed to settle all the payments at the end of the day and the upper bound is the liquidity needed to settle all payments immediately. On average the lower bounds of liquidity were 10.7% and 4.3% of the total value of payments for the Danish and Finnish data, respectively. The upper bounds were 37.2% and 27.4%, respectively.

**Table 3. Summary of liquidity requirements in both systems (mil. euro)**

	DN Inquiry and Transfer System			BoF-RTGS		
	(DK)			(FI)		
	Minimum	Maximum	Average	Minimum	Maximum	Average
System UB of liquidity	634	4 925	3 421	639	5 957	2 746
System LB of liquidity	269	2 276	958	11	3 233	423
UB as % of payment flow, %	29.2	50.7	37.2	15.9	48.9	27.4
LB as % of payment flow, %	4.1	24.0	10.7	0.1	26.6	4.3

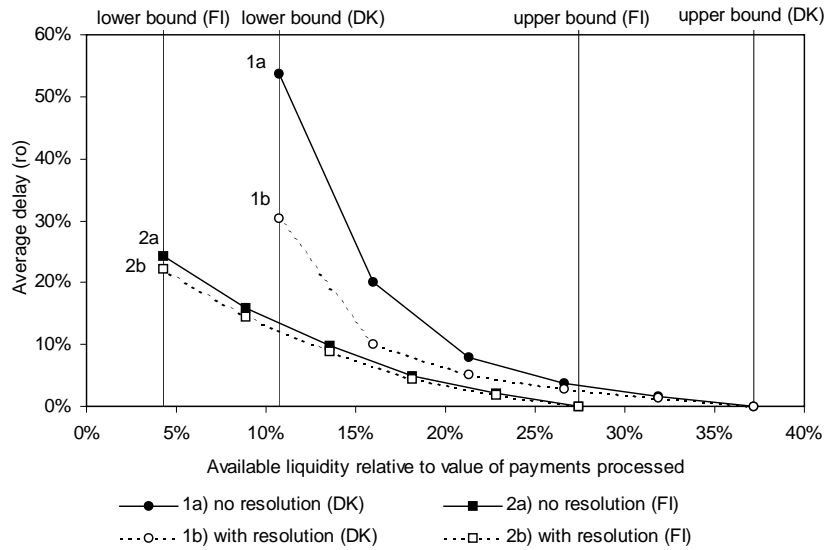
The smaller lower and upper bounds for the Finnish system - shown in Table 3 - can be explained by two factors: Firstly, the number of participants in the Finnish system is smaller and the payment flows are more homogenous (the netting effect is thus higher and the intra-day payment flows more balanced). Secondly, because of the extra liquidity, which the Finnish banks receive from other components of the TARGET system.

### 5.2 The trade-off between liquidity and delay

The trade-off between liquidity and delay for the simulated systems with and without the gridlock resolution mechanism is shown in Figure 1. The horizontal axis shows the amount of liquidity available in the system relative to the total value of payments processed in the system. The vertical axis shows the delay indicator  $\rho$  discussed in section 4.1 before.

**Figure 1.**

**Trade-off between liquidity and delay**



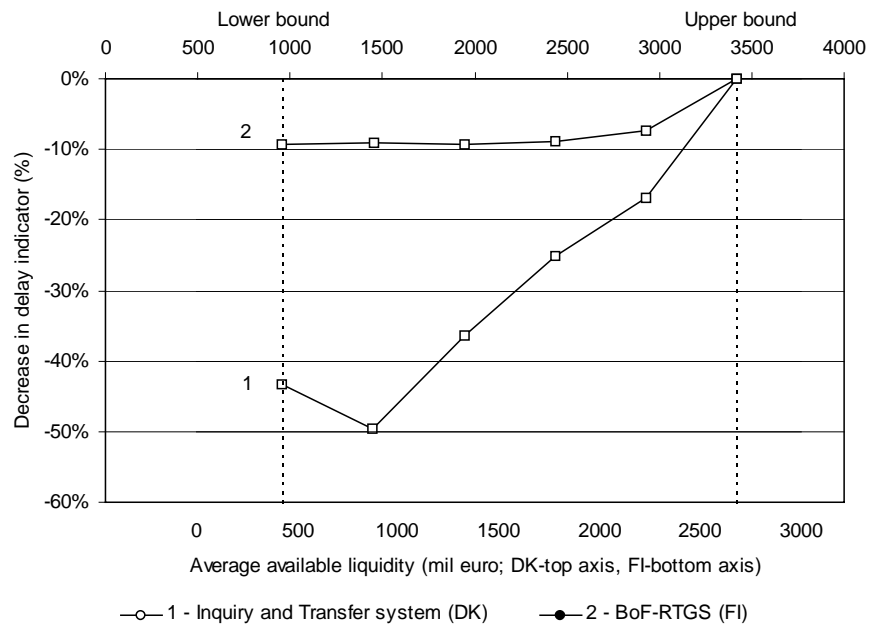
The range at which liquidity can be substituted for settlement delay at system level is rather wide in both cases, on average some 23% of the daily value of payments with Danish data and 26% with Finnish data. Both curves are convex, the curve representing simulations on Danish data to a larger extent than the curve with Finnish data. The convexity of the curves can be explained by the increase in gridlocks and the resulting increase in deadweight losses that are experienced when the available liquidity in the system is reduced.

As can be seen in the figure, the settlement delay is reduced by the proposed gridlock resolution mechanism at all levels of liquidity below the upper bound. The reduction in delay is larger at the lower levels of liquidity in both systems. Moreover, the gridlock resolution mechanism is significantly more effective with Danish data than the Finnish data.

The relative reduction in settlement delays at different levels of liquidity is illustrated in Figure 2. The delay indicator was reduced by up to 50% for low levels of liquidity with Danish data whereas simulations with Finnish data showed a considerably more modest reduction. At low and moderate levels of liquidity, delay could be reduced by some 10%. This is a consequence of the smaller number of gridlocks that are experienced in the system (see section 5.3). At high levels of liquidity the reduction was rather small with both sets of data.



**Figure 2. Reduction in delays**

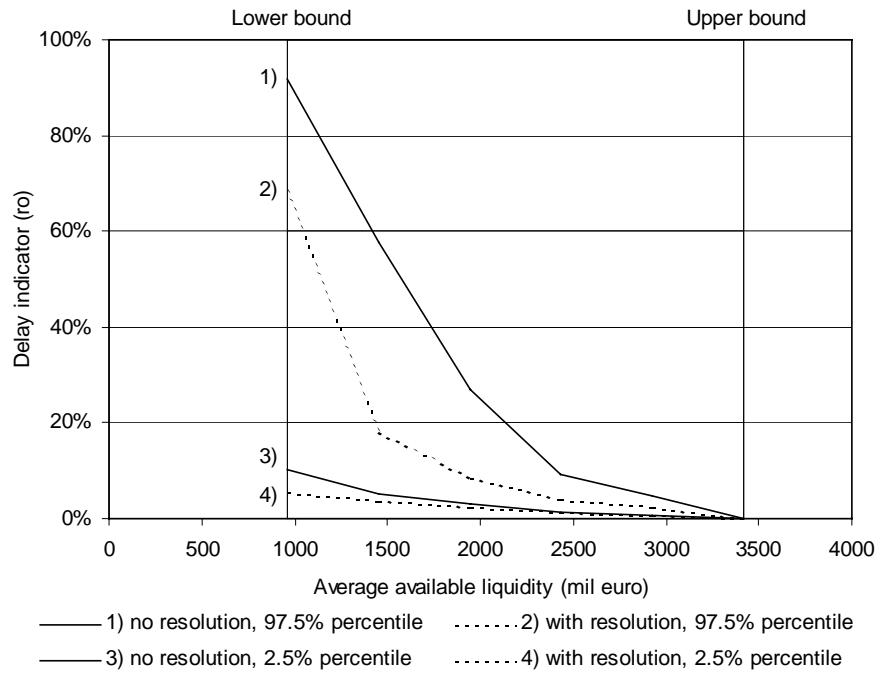


Also the average value of queued payments was reduced. The reduction amounted with Danish data to about 2 million kroner (0.3 million euro) at high levels of liquidity and 35 million kroner (4.7 million euro) at low liquidity levels. The maximum value of all queued payments during the whole period was reduced between 0.7 and 1.9 billion kroner (94 and 255 million euro), at high and low levels of liquidity respectively. With Finnish data the average reduction was about 3 million euro at high levels and 10 million euro at low levels of liquidity. The effects on the maximum value of payments queued were almost non-existent.

As was shown in Figure 1 the delays in the system increase as the liquidity available is reduced. In addition, also the variability of the settlement delay increases. The 95% inter percentile ranges are shown for both systems with and without gridlock resolution in Figures 3a and 3b respectively. As can be seen from the figure, the 95% inter percentile range without gridlock resolution (area between curves 1 and 3) for the delay indicator grows from zero to almost 80% in the Danish case when the liquidity available is reduced from the upper to the lower bound. The comparable range is up to 35% with simulation on Finnish data. The use of the gridlock resolution mechanism also reduces the variability of the settlement delay on a daily level (area between curves 2 and 4). In particular, the gridlock resolution mechanism reduces the probability of days with severe settlement delays, since it reduces the 97.5% percentile substantially (curves 1 and 2). The reduction in the 2.5% percentile (curves 3 and 4) is not affected as much by the gridlock resolution in either set of data.

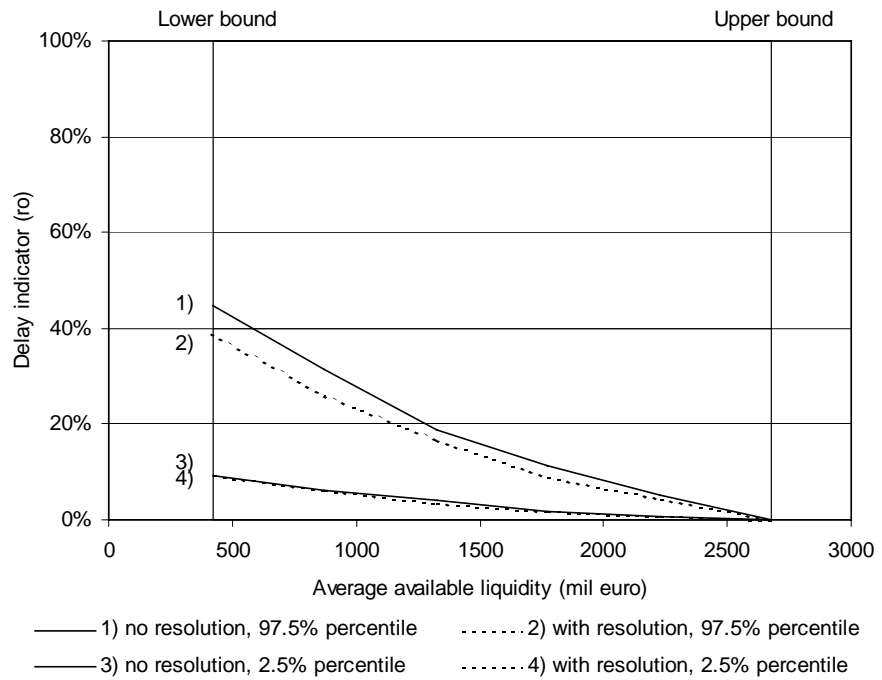
**Figure 3a.**

**95% inter percentile range for the delay indicator  
(Danish data)**



**Figure 3b.**

**95% inter percentile range for the delay indicator  
(Finnish data)**

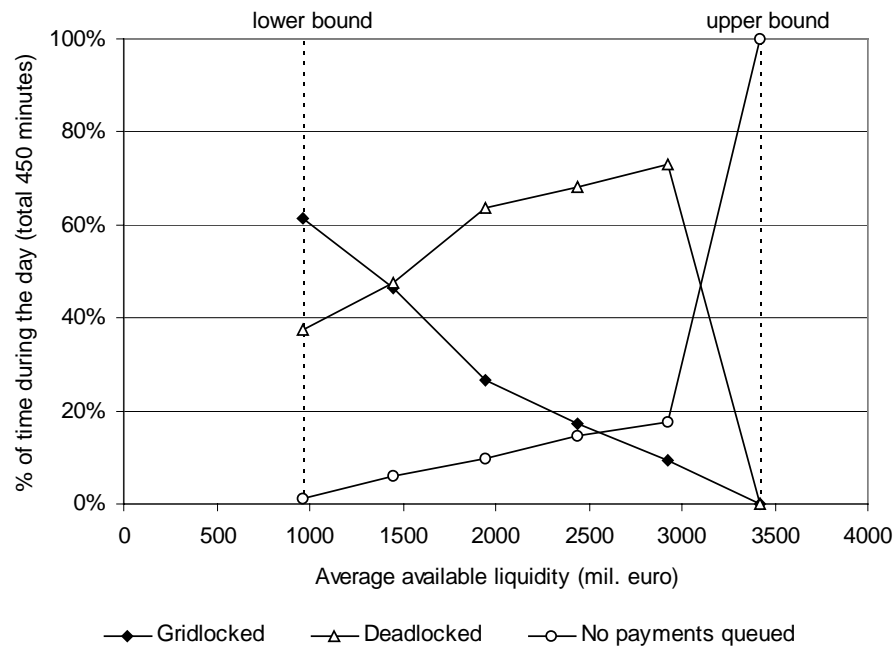


### 5.3 Gridlocks experienced in the system

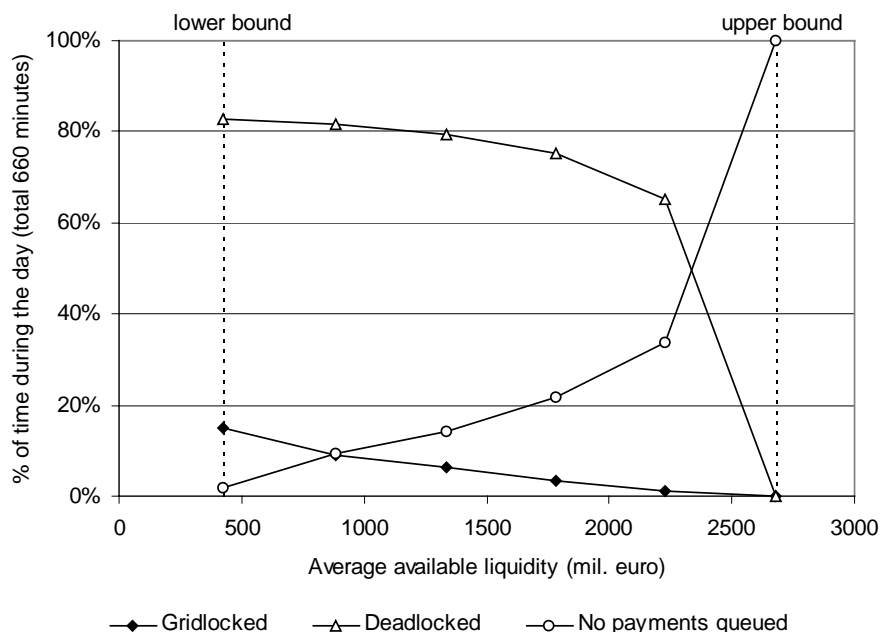
In the previous section we described how the trade-off between liquidity and delay was affected by the implementation of a gridlock resolution mechanism. In this section we look at the state of the queue and the actual solutions to the gridlock resolution problem in more detail.

The state of queue on an average day as a function of available liquidity is shown for the two simulated sets of data in Figures 4a and 4b. The simulations show that even a modest reduction in the liquidity available leads to a substantial reduction in the fraction of the day (a total of 450 and 660 minutes, for Danish and Finnish data respectively) where no payments are queued. Reducing the level of liquidity by one fifth resulted in the formation of queues on 82% of the time on an average day with Danish data and on 66% with Finnish data. At the lower bound of liquidity queues were a permanent feature with both sets of data. In fact, on average the systems experienced the state of “no payments queued” for less than 15 minutes per day when operating at the lower bound of liquidity.

**Figure 4a. System state without resolution (Danish data)**



**Figure 4b. System state without resolution (Finnish data)**



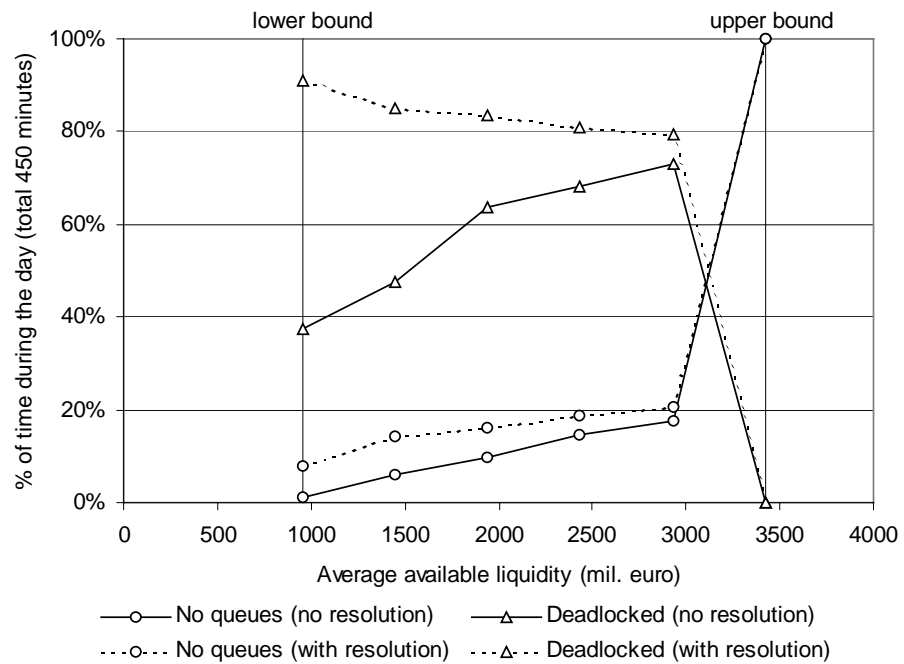
Further more, the simulations show that the fraction of the day when the queue is gridlocked increases as the liquidity available is reduced. This relationship is almost linear in both cases. In the simulations with Danish data a reduction of the available liquidity by 66 million kronor (8.9 million euro) in the system leads to one additional gridlock minute on average. With Finnish data one more minute is gridlocked for every 23 million euro reduction in available liquidity for the system as a whole. The sensitivity of the system to gridlocks was thus much higher with Danish data than with Finnish data. At the lower bound of liquidity the system simulated with Danish data was on average gridlocked 62% of the time during an average day, while with Finnish data gridlocks existed only on some 15% of time in the system.

The smaller degree of queuing and gridlocks in BoF-RTGS can in part be explained by the fact, that in the simulations payments from participants outside Finland (i.e. incoming TARGET payments) were not queued. The rationale for this was that the source of these payments is another RTGS system, with possibly other liquidity optimization and management features than BoF-RTGS. As liquidity by the Finnish banks in the simulations is reduced, the effects are only partly experienced by the banks themselves and some of the effects leak to the other TARGET RTGS systems.

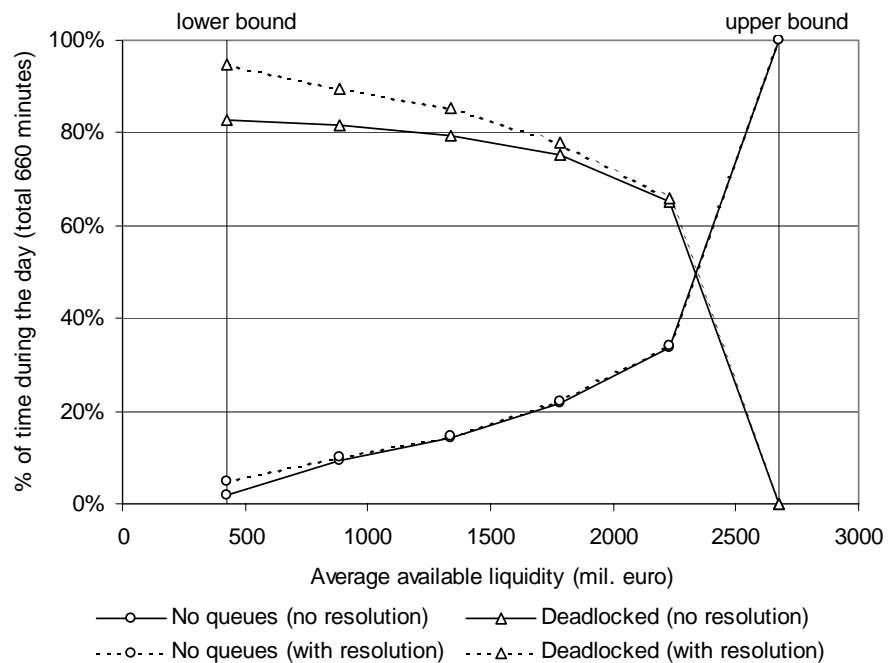
The effect of gridlock resolution on the state of the system can be seen in Figures 5a and 5b. In both cases the implementation of gridlock resolution increases the fraction of the day where no payments are queued. In the Danish case the number of minutes without any queues is considerably increased when the gridlock resolution algorithm is applied. At low levels of liquidity the increase is around half an hour and at higher levels of liquidity between 10 and 20 minutes.

With Finnish data the reduction is not as large, but still noticeable. At the lower bound of liquidity, the reduction was on average 20 minutes, at higher levels of liquidity between 3 and 5 minutes.

**Figure 5a. The effects of resolution on system state (Danish data)**



**Figure 5b. The effects of resolution on system state (Finnish data)**



However, most of the gridlocked queues could only be resolved partially and simultaneous settlement of all queued payments was possible rather rarely. Full resolution was more likely at high levels of liquidity than at low levels of liquidity. At the lower bound of liquidity only some 11% (DK) and 20% (FI) of gridlocks were fully solvable in the respective systems. Full gridlock resolution was successful with Danish data on the average every second day and with Finnish data every third day. In contrary, the partial gridlock resolution feature was successful on the average 5 and 2 times a day, with Danish and Finnish data respectively. The fact that most of the resolutions were only partial explains why the fraction of the day where the systems have a deadlocked queue also increased as a result of implementing a gridlock resolution mechanism. Recall that a deadlocked queue can only be settled by the infusion of additional liquidity by one or more participants with queued payments. When gridlocks are removed, either all queued payments are settled or the remainder of the queue is left deadlocked. For any given amount of liquidity an efficient system is one with no queuing or queues that are deadlocked.

## 6 Conclusions

When banks do not hold sufficient liquidity to settle all obligations immediately the system will experience gridlocks that could be solved by an appropriate algorithm. For both simulations with Danish and Finnish data we found an almost linear relationship of increasing gridlocks as the liquidity in the system was reduced.

With Danish data the system was up to 62% of the time in gridlock, when liquidity was scarce. The potential for a resolution of gridlocks thus clearly exists. By applying the algorithm queuing could be substantially reduced, by up to 50 % at low levels of liquidity with Danish data. The effects were more modest, but still concrete at higher levels of liquidity. With Finnish data the effects were relatively modest, mainly due to the fact that the system experienced much fewer gridlocks. This again could be explained by the fact that payments arriving from other RTGS systems in TARGET were not queued, and represented an extra source of liquidity for the banks. The liquidity received from these payments could be used for settling outgoing payment irrespective of the liquidity used in the simulations. Thus the scope for optimization was smaller in the Finnish case, which was simulated as an ‘open’ system in contrast to the ‘closed’ Danish system.

The proposed algorithm was also found to reduce the risk of severe settlement delays, with Danish data to a larger extent than with Finnish data. This is a further improvement in the system and should enable banks to operate on lower levels of liquidity.

The use of the algorithm can be seen as an improvement to the current system designs as all other aspects of settlement can remain as they are. The banks can continue to be sure that the payments are released in their preferred order as any sequence constraint can be used in solving the Gridlock Resolution Problem, reducing the legal risk involved. Due to this fact, the algorithm is also invariant whether prioritization of payments or multiple payment classes are used. In these cases the underlying set of payment is just organized differently. Further on, the calculation time of the algorithm increases only linearly with the size of the problem, resulting in fast calculation times even with large numbers of queued payments. Thus the ongoing RTGS settlement does not have to be suspended for a significant period of time.

Overall, we find that gridlock resolution, from the central bank perspective, can be seen as a tool to ensure a smoother functioning of the system when liquidity holdings of the participants are low, or when the system is experiencing a temporal shortage of liquidity. From the participants’ perspective, it reduces the costs associated in settlement, either through decreased delay costs or through their ability to hold less liquidity for the settlement of payments without incurring more delays in payments. We conclude that the algorithm explained in this paper is of practical importance, efficient, and easy to implement in the design of existing RTGS systems.

## References

- Angelini, Paolo. "An Analysis of Competitive Externalities in Gross Settlement Systems", *Journal of Banking and Finance* 22, (1998) pp. 1-18
- Bank for International Settlements. "Payment Systems in the Group of Ten countries" prepared by the Committee on Payment and Settlement Systems of the central banks of the Group of Ten countries, Basle, (December 1993).
- Bank for International Settlements. "Real Time Gross Settlement Systems" prepared by the Committee on Payment and Settlement Systems of the central banks of the Group of Ten countries, Basle (March 1997).
- Bech, Morten L. and Rod Garratt. "The Intraday Liquidity Management Game", unpublished manuscript, University of California, Santa Barbara (April 2001).
- Güntzer, Michael M., Dieter Jungnickel and Matthias Leclerc. "Efficient algorithms for the clearing of interbank payments", *European Journal of Operational Research* 106 (1998) 212-219.
- Horowitz, Ellis and Sartaj Sahni. "Fundamentals of Computer Algorithms", Computer Science Press, Inc. (1978)
- Kahn, Charles and William Roberds. "Payment System Settlement and Bank Incentives", *The Review of Financial Studies* 11, (1998) pp. 845-870.
- Koponen, Risto and Kimmo Soramäki. "Intraday Liquidity Needs in a Modern Interbank Payment System - A Simulation Approach". *Studies in Economics and Finance* E:14, Bank of Finland, Helsinki (1998)
- Leinonen, Harry and Kimmo Soramäki. "Optimizing Liquidity Usage and Settlement Speed in Payment Systems", Bank of Finland Discussion Paper 16, (1999).
- McAndrews, James and Samira Rajan. "The Timing and Funding of Fedwire Funds Transfers", *FRBNY Economic Policy Review*, July 2000
- Soramäki, Kimmo. "Alternative liquidity management features in an RTGS environment", Bank of Finland Working Paper, March 2000



## Appendix 1.

$$\begin{aligned} & \max_{0 \leq l \leq m} \sum_{i=1}^n l_i \\ & \text{s.t.} \\ & B_i(\bar{B}_i, l) = \bar{B}_i - S(l_i) + R(l_{-i}) \geq 0, \quad \text{for } i = 1, \dots, n \end{aligned} \tag{5}$$

where

$$\begin{aligned} S(l_i) &= \sum_{i=1}^{l_i} a_{i,k} \\ R(l_{-i}) &= \sum_{j=1}^n \sum_{k=1}^{l_j} a_{j,k} I(r_{j,k} = i) \end{aligned} \tag{6}$$

The properties of the solution and the algorithm can be shown using the following lemma

**Lemma 1:** Suppose that  $y$  and  $z$  are two solution candidates such that  $z$  is feasible,  $y \geq z$  i.e.  $\forall_{i=1}^n y_i \geq z_i$ , and  $B_j(\bar{B}_i, y) = \bar{B}_i - S(y_j) + R(y_{-j}) < 0$  for some bank  $j$

Let

$$\hat{y}_i = \begin{cases} y_i - 1 & \text{if } i = j \\ y_i & \text{if } i \neq j \end{cases} \quad \text{for } i = 1, \dots, n$$

then  $\hat{y} \geq z$ .

**Proof:** We only need to show that  $z_j \leq \hat{y}_j = y_j - 1$  since for  $i \neq j$ ,  $z_i \leq \hat{y}_i = y_i$  is trivially satisfied. We have by hypothesis that  $z$  is feasible and that  $y \geq z$ . This implies that

$$R(y_{-i}) \geq R(z_{-i}), \quad \text{for } i = 1, \dots, n$$

i.e. that the value of incoming payments are going to be at least as large for bank  $i$  when the solution is  $y$  as if the solution is  $z$ . This implies

$$\bar{B}_i - S(z_i) + R(y_{-i}) \geq \bar{B}_i - S(z_i) + R(z_{-i}) \geq 0 \quad \text{for } i = 1, \dots, n$$

and in particular for  $i = j$ . However, by hypothesis we have that

$$B_j(\bar{B}_j, y) = \bar{B}_j - S(y_j) + R(y_{-j}) < 0$$

We have

$$\left. \begin{array}{l} B_j(\bar{B}_j, y) = \bar{B}_j - S(z_j) + R(y_{-j}) \geq 0 \\ B_j(\bar{B}_j, y) = \bar{B}_j - S(y_j) + R(y_{-j}) < 0 \end{array} \right\} \Rightarrow S(y_j) > S(z_j) \Rightarrow y_j > z_j$$

as required since  $\hat{y}_j = y_j - 1$

◆

The lemma stipulates that if we have two possible solutions where one ( $y$ ) is larger is than the other ( $z$ ) and only the smaller solution is feasible then omitting one payment at a time from a deficient bank in the larger solution will result in a new solution that settles at least the same payments as the feasible solution. In others words the algorithm can not bypass the feasible solution.

**Corollary 1:** The solution found by the algorithm is indifferent to which of the deficient banks a payment is removed from.

**Proof:** Assume that the optimal solution is the feasible solution  $z$  and the corollary follows from lemma 1. ◆

**Corollary 2:** The algorithm always finds a unique and optimal solution

**Proof:** We have that the algorithm starts with all the payments in queue and removes one payment at a time By lemma 1 it can not bypass a feasible solution, the first feasible solution is thus the optimal solution and it is unique. ◆

**Corollary 3:** The solution to the problem is invariant to whether the total value or number of payments is being maximized.

**Proof:** If there is only one feasible solution then this solution is going to have both the maximum total value and number of payments. If there are two or more feasible solutions then any other feasible solutions than the first one found by the algorithm will by lemma 1 contain the same or fewer payments for each bank and thus less value. ◆