Author's Accepted Manuscript

www.elsevier.com/locate/physa

Congestion and cascades in payment systems

Walter E. Beyeler, Robert J. Glass, Morten Bech, Kimmo Soramäki

PII: S0378-4371(07)00597-3
DOI: doi:10.1016/j.physa.2007.05.061
Reference: PHYSA 10777

To appear in: *Physica A*

Received date: 28 July 2006
Revised date: 23 February 2007
Accepted date: 15 May 2007

Cite this article as: Walter E. Beyeler, Robert J. Glass, Morten Bech and Kimmo Soramäki, Congestion and cascades in payment systems, *Physica A* (2007), doi:10.1016/j.physa.2007.05.061

Congestion and Cascades in Payment Systems

Walter E. Beyeler

Robert J. Glass

Morten Bech

Kimmo Soramäki

26 July 2006

Revised 23 February 2007

Abstract

We develop a parsimonious model of the interbank payment system. The model incorporates an endogenous instruction arrival process, a scale-free topology of payments between banks, a fixed total liquidity which limits banks' capacity to process arriving instructions, and a global market that distributes liquidity. We find that at low liquidity the system becomes congested and payment settlement loses correlation with payment instruction arrival, becoming coupled across the network. The onset of congestion is evidently related to the relative values of three characteristic times: the time for banks' net position to return to 0, the time for a bank to exhaust its liquidity endowment, and the liquidity market relaxation time. In the congested regime settlement takes place in cascades having a characteristic length scale. A global liquidity market substantially attenuates congestion, requiring only a small fraction of the payment-induced liquidity flow to achieve strong beneficial effects.

Key words: network, topology, interbank, payment, money market, Sandpile model, congestion

The National Infrastructure Simulation and Analysis Center (NISAC) is a program under the Department of Homeland Security's (DHS) Preparedness Directorate. Sandia National Laboratories (SNL) and Los Alamos National Laboratory (LANL) are the prime contractors for NISAC under the programmatic direction of DHS's Infrastructure Protection/Risk Management Division.

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company for the United States Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

Beyeler: Sandia National Laboratories. Glass: Sandia National Laboratories. Bech: Federal Reserve Bank of New York. Soramäki: Helsinki University of Technology. Address correspondence to Walter E. Beyeler (e-mail: webeyel@sandia.gov). The views expressed in this paper are those of the authors and do not necessarily reflect the position of the Federal Reserve Bank of New York or the Federal Reserve System.

1 Introduction

Modern economies depend on efficient and reliable financial markets. Critical to the smooth functioning of these markets are a set of trading, payment, clearing and settlement infrastructures. Financial infrastructures are formed by a large number of technological and institutional components that interact within complex networks.

One core infrastructure is the interbank payment system which allows movement of funds between banks. Funds transfers may be related to transactions originating from money, foreign exchange or securities markets. The Fedwire Funds Service operated by the U.S. Federal Reserve, for example, processes more than five hundred thousand payments daily with a total value exceeding $2 trillion [1]. The European TARGET system handles similar volumes in euros [2]. Funds transfers generally take place on the books of a central bank. In such systems the transfers take place in real time and funds received can immediately be used to effect further payments. This reuse allows the system to settle payments using only a small fraction of the daily turnover. For example the daily flow of roughly $2 trillion in Fedwire is supported by a total daily account balance of approximately $15 billion [1] and an average daily overdraft of approximately $36 billion [3].

Participants have an economic incentive to minimize the funds committed to payment processing because liquidity used for settling payments imposes an opportunity cost on banks. Under-funding can also be costly, especially for bank customers and other banks in the system. Shortfalls of funds can delay a bank's payment processing, and payment systems can even enter gridlock states in which no bank can process a payment [4]. Delayed payments are unavailable to intended recipients: in this way congestion in the payment system can propagate into the economy by restricting money flow among banks and eventually among their customers.

Large-scale simulations of payment systems have been used to evaluate the possible consequences of changes in payment system rules and policies under both normal and disrupted conditions. A number of such studies are available in Leinonen [5]. These simulations have used detailed descriptions of the business rules followed by the diverse participants, including banks and system operators, to anticipate the response of specific systems to potential stresses.

In this paper we develop a parsimonious model of the interbank payment system to study congestion and the role of liquidity markets in alleviating congestion. This model focuses on the essential dynamics of payment processing in order to understand how networks of interacting agents, each following simple rules, can give rise to system-level congestion. The main features are an endogenous instruction arrival process, a scale-free topology of payments between banks as found in real interbank payment systems [6,7], a fixed total liquidity used by banks to process arriving instructions, the ability of banks to build and work off queues of instructions they cannot process, and a global market that distributes liquidity among the banks. Because we focus on the influence of liquidity and liquidity distribution mechanisms on system performance, we assume that all banks follow a cooperative strategy in submitting payments.

Financial relationships among individual decision-makers are increasingly represented using network models. One thread of research, for example [8-11], models price formation in a market made by agents responding to the behavior of their immediate neighbors in an influence network. Another thread [12-16] examines the flow of diverse goods and services between producers and consumers through a set of intermediaries, where the network links model pairwise specifications of cost and information among the individual decision-makers. In these and similar studies, the operation of the payment systems that undergird individual financial transactions is presumed. The purpose of the present study is instead to focus on payment system operations, using a stipulated forcing function to create the payment flows that express underlying economic relationships

Network models of queuing agents have been studied in many contexts, including manufacturing processes and supply chains, computer networks, and transportation infrastructures [13,16,17,18]. In these models, interactions between agents generally represent transfer of workload from agent to agent. In contrast, interactions between banks in payment systems transfer *capacity* rather than workload: a bank that sends many payments and sends them promptly tends to relieve rather than create congestion at receiving banks. Congestion in a payment system is both a cause and a consequence of reduced transfer capacity.

Our model has been developed in the spirit of abstract models used to study critical behavior. Bak *et al.* [19] discovered self organization in systems of locally interacting elements with non-linear dynamics. In their pioneering model random stresses impinge on a lattice of coupled elements, which discharge when their state variable exceeds a threshold. Stress is dissipated at the lattice boundaries. This system is driven to a state characterized by discharge cascades at all length scales. Sachtjen *et al.* [20] studied the effect of several stylized topologies in systems consisting of elements with similar threshold discharge behavior, but that were stressed by random bilateral exchanges between pairs of elements connected by links.. This system undergoes a transition in a tolerance parameter, below which the average cascade size becomes unbounded. It may be tuned to a critical state but, unlike Bak et al., is not driven to one.

We find that at sufficiently low liquidity our system becomes congested and payment settlement loses correlation with payment arrival. At low liquidity banks' payment processing becomes coupled creating periods of congestion and episodic cascades of settlement. Settlement cascades have a characteristic size. We find that a global liquidity market, a feature of many modern payment systems, can effectively compensate for the imbalances created by the payment instructions received by banks. Congestion can be substantially attenuated by the market, and only a small fraction of the payment-induced liquidity flow is required to achieve strong beneficial effects.

2 Model Definition

2.1 Primitive payment system

We consider an economy populated with *productive agents*, *banks*, and a *central bank* administering an interbank payment system. Figure 1 illustrates the model components, state variables, and processes. Productive agents, representing the external economy, hold deposits at banks to settle obligations arising from trades with each other. Banks maintain balances at the

central bank to transfer the funds related to the payment instructions received from their agents and destined to agents banking at other banks[1]. For simplicity we assume that all payments are of equal size. A bank's ability to execute payment instructions depends on the availability of funds on its account at the central bank. We assume that banks are reflexively cooperative: they settle payments whenever they have funds to do so. Otherwise arriving payment instructions are placed on a queue. Whenever funds are received by a bank, these funds are used to immediately settle previously queued instructions.

We model instruction arrival as a random process: instructions of unit size arrive at random real intervals according to a non-stationary Poisson process (Section 2.3 below and Appendix A). Let $I_i(t)$ denote the rate of payment instructions submitted to bank *i* by its productive agent for processing at time *t*. $S_i(t)$ denotes the rate of payments sent and $R_i(t)$ the rate of payments received by bank *i* at time *t*. We define the state of a bank by the value of the deposits $D_i(t)$ held by its agents, the value of its balance at the central bank $B_i(t)$, and the value of payment instructions in its queue $Q_i(t)$.

$$D_i(t) = D_i(0) + \int_0^t \big(R_i(s) - I_i(s)\big)\, ds \tag{1}$$

$$B_i(t) = B_i(0) + \int_0^t \big(R_i(s) - S_i(s)\big)\, ds \tag{2}$$

$$Q_i(t) = \int_0^t \big(I_i(s) - S_i(s)\big)\, ds \tag{3}$$

2.2 Diffusive Liquidity Market

In an elaboration of the primitive model above, banks can lend excess funds and cover a shortfall in funds using a liquidity market. The liquidity market is modeled as a linear diffusive process. We define a *liquidity potential*

$$Z_i(t) \equiv B_i(t) - B_i(0) - Q_i(t) \tag{4}$$

for each bank. The liquidity potential is the the difference between a bank's current net position of obligations ($B_i(t) - Q_i(t)$) and its initial funding. We use this potential to describe the bank's willingness to supply funds to the market. Banks with excess funds will supply them to the market at a rate proportional to their surplus, and banks in deficit will obtain funds in proportion to their deficit. We define a market conductance *c* to capture the effects of transaction costs, information costs, and any other factors constraining liquidity transfers among banks. The aggregate flows between banks and the market are always in balance as given by a simple conservation equation

$$\sum_{i \in A} \big(Z_i(t) - Z_m(t)\big) \cdot c = 0 \tag{5}$$

[1] Transfers between agents within the same bank are not modeled.

where Z_m is the market potential, and A the set of banks active in the market. From (5) it follows that the market potential is

$$Z_m(t) = \frac{1}{n_{\mathrm{A}}} \sum_{i \in \mathrm{A}} Z_i(t) \tag{6}$$

where n_{A} is the number of active banks. A bank participates in the market as a borrower if its potential is lower than the market potential. It participates as a lender if its potential is higher than the market potential, and if it has funds to lend. These conditions can be summarized as

$$i \in \mathrm{A} \quad \textit{iff} \quad Z_i(t) > -B_i(0) \quad \textit{or} \quad Z_i(t) < Z_m(t) \tag{7}$$

Each bank's flow of funds in to or out of the market varies linearly with its liquidity potential $Z_i(t)$. The total net flow of bank *i* to the market until time *t* is given by

$$V_i(t) = \int_{s=0}^{t} c\left(Z_i(s) - Z_m(s)\right) M_i(s) ds \tag{8}$$

where $M_i(t)$ is an indicator of the bank's participation in the market at time *t*: $M_i(t) = 1$ if $i \in A$ at *t*, $M_i(t) = 0$ otherwise. With the diffusive liquidity market, the balance of the bank is affected, in addition to Equation 2**,** also by its funding activities and is given by

$$B_i(t) = B_i(0) + \int_0^t \left(R_i(s) - S_i(s)\right) ds - V_i(t) \tag{9}$$

Equations 4 through 9 define a set of first-order equations for $Z_i(t)$, having a characteristic time of $\tau_m = 1/c$.

2.3 Relationships between environment variables

Payment instructions are modeled as random events of unit size. We assume that payment instructions to a bank are driven by the level of deposits $D_i(t)$ held by its productive agent, which may be converted into a payment instruction with a constant probability per unit time p_e. The expected rate of instruction arrival to a bank is defined as $\langle I_i(t) \rangle = p_e D_i(t)$. This frequency can be expressed in terms of its initial value $\lambda_i = p_e D_i(0)$ as

$$\langle I_i(t) \rangle = \lambda_i \cdot \frac{D_i(t)}{D_i(0)} \tag{10}$$

As defined by Equation 1, instructions received by a bank reduce deposits, while payments arriving at the bank increase deposits, thereby creating new obligations which may later be called upon with the common probability per unit time p_e. Thus payment arrival rate increases as incoming payments add to deposits, and decreases as instructions from the productive agent deplete deposits. Unlike the case of pure Brownian motion the dependence of instruction arrival rate on deposits in Equation 10 creates a finite expected time for $D_i(t)$ to return to $D_i(0)$. Numerical simulations indicate that this return time τ_r is approximately equal to $D_i^{1/2}(0)/\lambda_i$ (see Appendix A).

The initial distribution of deposits among the *n* banks in the network is assumed to follow a power law: $p(D_i(0)=d) \propto d^{-\gamma}$. The assumption is inspired by the power law distribution found for US firm sizes [21]. Thus also the initial instruction arrival rate has a power law distribution among the banks.

We assume that each bank *i* interacts exclusively with k_i other banks. These interaction paths define links in the payment network. The degree of each bank scales with its deposits $k_i \propto D_i(0)$. More specifically we define a deposits-per-link parameter d_0 that ties the network topology to deposits

$$D_i(0) = d_0 k_i \tag{11}$$

Using the algorithm described in Appendix B, we construct a network of payment exchange pathways among banks based on their degree k_i. This gives us a scale-free network topology as found in real interbank payment systems [6,7].

An instruction arriving to bank *i* is assumed to be equally likely to be destined for any of the k_i neighboring banks. We stipulate that on average the flow of instructions for bank *i* to pay some other bank must equal the flow of instructions for other banks to pay bank *i*. For this to hold, the instruction arrival rate must scale with k_i, which follows from the definition of λ_i and Equation 11.

Each bank sets its initial central bank balance $B_i(0)$ to control its risk of exhausting funds due to an imbalance between payment instructions and receipts of funds. If all instructions were executed immediately, a bank's balance would approximately follow a random walk[2]. The standard deviation of a bank's balance will therefore grow over short time scales as $n_i^{1/2}$ where n_i is the number of payments sent or received by bank *i*. This number is proportional to the instruction arrival rate, which in our setup is initially proportional to the bank's starting deposits $D_i(0)$. The initial balance required to provide a given probability of not running short of funds will therefore scale with $D_i(0)^{1/2}$. This formulation is reasonable in that banks with larger

[2] as long as changes in D(t) are small, and the instruction arrival rate from Equation (10) can be assumed constant

deposits maintain larger balances, while the sub-linear dependence reflects economies of scale. We set each bank's initial balance[3] as

$$B_i(0) = l \cdot \left(D_i(0) \Big/ d_0 \right)^{1/2} \tag{12}$$

where l is a global liquidity factor. The initial balance allows a bank to continue to submit payments for a time although its net position is negative.

The operational time "bought" with the initial liquidity can intuitively be defined as the time at which the probability of having a positive balance falls to specific value. For convenience, and because the balance distribution is approximately normal, we use the time at which the standard deviation of the balance grows to equal the initial balance. We can relate the liquidity factor l to this time. The initial rate of instruction arrival at a bank is $\lambda_i = p_e D_i(0) = p_e d_0 k_i$ so that the expected number of payment events, including sent and received payments, in a time τ_l is $n_i = 2\tau_l \lambda_i = 2\tau_l p_e d_0 k_i$. Because, from Equation 11, $D_i(0) \Big/ d_0 = k_i$, equating the balance standard deviation $n_i^{1/2}$ and the initial balance from Equation (12) yields $\tau_l = \dfrac{l^2}{2 p_e d_0}$.

3 Mean-field Approximations

The performance of the primitive model can be approximated under certain limiting conditions. Understanding these bounding cases helps us interpret the results we obtain from the simulations.

In a network of n banks suppose that banks in a subset Λ are liquid, i.e. have a balance that allows execution of an instruction. The remaining banks have a balance of zero. Instructions received by banks having zero balance will accumulate in their queues. An instruction received by a bank in Λ will be executed, allowing the receiving bank to execute a queued payment - if it has one. This in turn allows the recipient of the last executed payment to process one of its queued payments, and so on. The chain continues until the recipient of a processed payment has no queued instructions. We assume that when a payment arrives, any queued instruction is processed immediately, so that the entire chain of queued payments is traversed before any new instruction can arrive. Subsequent instructions will rebuild the system's queued instruction inventory until a liquid bank receives a new instruction. This instruction will again catalyze a chain of payments through the bank network. The length of such *settlement cascades* is the number of queued instructions whose execution is enabled by the execution of an instruction arriving at a liquid bank.

[3] Equation (12) allows banks to have fractional balances. Without a market any fractional part of a bank's balance is permanently unavailable for payment processing and is effectively lost from the system. This artifact would make total system liquidity a discontinuous function of l, as well as subject to sampling error via the network realization. When a market is not included we require integer initial balances and interpret any fractional part of Equation (12) as the probability of having an additional unit balance.

We hypothesize that the primitive system will be driven to a quasi-steady state in which, on average, the accumulation of queues is balanced by the release of queued instructions in settlement cascades. Queues will build within the system until arriving payments create cascades of sufficient average length and frequency to discharge the instructions that become queued between the cascades.

This hypothesis entails relationships among the average values of certain system state variables. An arriving instruction will be immediately executed if it is received by a bank in Λ. We denote the occurrence of this event as E. Assuming deposits are close to their initial value, the expected arrival rate of instructions at bank *i* is, from Equation (10), approximately proportional to the degree of the bank:

$$\langle I_i(t) \rangle \approx \lambda_i = p_e d_o k_i \tag{13}$$

Because the instruction streams are independent across banks the probability that the instruction arrives at a bank with funds is therefore the fraction of edges in the entire network incident on banks in Λ:

$$P(E) = \frac{\sum_{i \in \Lambda} p_e d_0 k_i}{\sum_i p_e d_0 k_i} = \frac{\sum_{i \in \Lambda} k_i}{\sum_i k_i} \tag{14}$$

With a complementary probability, $P(\bar{E}) = 1 - P(E)$, the instruction is queued.

The number of queued payments N_E executed as a result of a new instruction depends on the payment instructions arriving to a bank with funds (event *E* occurs) and on the length of the cascade that the initiating payment releases. If executed instructions induce a settlement cascade of length L, we have $\langle N_E \rangle = P(E)\langle L \rangle$ assuming independence of *E* and *L*. The number of instructions added to queues N_Q is one for each queuing event $\bar{E}$, so that $\langle N_Q \rangle = P(\bar{E}) \cdot 1 = 1 - P(E)$. The equilibrium hypothesis states that the expected number of payments released from queues matches the expected number accumulated in queues.

$$\langle N_E \rangle = \langle N_Q \rangle \rightarrow P(E)\langle L \rangle \approx 1 - P(E) \tag{15}$$

or

$$\langle L \rangle = \frac{1}{P(E)} - 1 \tag{16}$$

.

The chain of release of queued payments begins with receipt of an instruction for a liquid bank to deliver a payment to a bank with queued instructions, and ends when the receiver of a payment has no queued payments. The probability of a particular chain of length h, conditional on the first bank having funds, is then:

$$P(L=h) = P(\bar{E}_1) \cdot P(\bar{E}_2) \cdot \ldots \cdot P(\bar{E}_h) \cdot P(E_{h+1}) \tag{17}$$

where $P(E_{h+1})$ is the probability that the bank receiving the last payment has no queued payments, and $P(\bar{E}_i)$ is the probability that the bank at step i of the cascade has queued payments[4].

The distribution of L can be derived under a mean field approximation. Comparing this approximation with model results will help gauge the influence of local correlations and network structure, which are ignored in the mean field approach. Using the mean field assumption that the probabilities of queuing instructions are equal across banks (i.e. $P(\bar{E}_i) = P(\bar{E}) \quad \forall i$), and assuming that the execution of the chain of payments does not change this probability, the path length has a geometric distribution

$$P(L=h) = P(E)P(\bar{E})^h = P(E)(1-P(E))^h, \quad h > 0 \tag{18}$$

$P(E)$ depends on the arrangement of liquidity within the network. Two factors determine whether a specific bank has a non-zero balance and is therefore able to execute an instruction: the specific sequences of instructions that the bank and its neighbors have received, and whether its neighbors have made or queued their outstanding payments to the bank in question. We can use the analytical expression for the probability distribution of net position (see Appendix A) to describe the first factor, but we cannot estimate $P(E)$ without including the second factor, and an approximation is not available. When we subsequently compare simulation results with Equation 18 we therefore use the *observed* state of the network to estimate $P(E)$, using Equation 14, as the fraction of network links originating at banks with a positive balance.

4 Simulation design and results

We designed simulations to explore model behavior as a function of the liquidity factor (l) and liquidity market conductance (c). The values for the two parameters were chosen to span the transition between non-congested and congested behavior. A d_0 value of 10000 was used in the main simulations, but we explored the impact of alternative levels of deposits to the instruction arrival in the sensitivity analyses in Section 4.3. Simulation durations were chosen by experimentation to allow the system to develop a quasi-steady state, after which state statistics were collected. The simulations use a network of 200 banks and 580 links representing bi-

[4] Note that the observation of a cascade is conditional on the initial bank being in a non-delaying state, so that this probability does not appear in the expression for $P(L=h)$

directional payment relationships among the banks. The average degree was 2.9 and the power law co-efficient of the degree distribution was approximately 2.5. We set $p_e d_0 = 1$, so that $\lambda_i = k_i$. This establishes a time scale for the simulation so that over a unit time interval each link in the network sends a single payment in both directions, on average. The simulations were made using a Java implementation of the algorithm described in Appendix B.

We first explored the onset of congestion in the primitive system as liquidity is reduced. Results from these analyses are discussed in Section 4.1. Simulations that include a liquidity market are presented in Section 4.2**.** In these sections we focus on global system properties such as throughput and cascade statistics. In Section 4.3 we describe sensitivity studies using alternative networks and deposit levels.

4.1 Primitive Payment System

The system begins with no queued payments and with liquidity distributed throughout the system based on Equation 12. Settled payments redistribute the initial liquidity, and queues build as some banks exhaust their funds. Examples of the growth of payment queues and the change in the fraction of instructions executed immediately are shown in Figure 2 for $l = 1$. As time passes, banks begin to queue more payments and also the length of settlement cascades increases. These cascades eventually counteract the accumulation of queued payments and the system settles into a quasi-steady state where the total number of queued payments fluctuates around a fixed level.

When liquidity is increased to a sufficiently high level, instructions can be processed promptly, and system output (in terms of settled payments) is very close to system input (in terms of payment instructions) in each time interval: A scatter-plot showing the number of arriving instructions and the number of settled payments in 10-unit time intervals for four levels of liquidity is presented in Figure 3. At the highest liquidity factor of 250, observations cluster along the diagonal where settled payments equal received instructions: there is a strong correlation between instructions and payments. As l decreases, banks experience temporary liquidity shortages and begin to queue their instructions until they receive funds from other banks.

This coupling of settlement among banks creates episodes of low payment volume (as queues build) and high payment volume (as queues are reduced by settlement cascades). Settlement becomes governed by the internal dynamics of coupled instruction queues rather than being driven directly by input instructions. With reduced liquidity, there is increasing variability in the number of payments settled in each interval, and payment settlement looses correlation with instruction arrival. Figure 4 shows the correlation coefficient for various liquidity factors.

As liquidity is reduced, the number and length of settlement cascades increases. The average length of settlement cascades with different levels of liquidity is shown in Figure 5. Below a liquidity factor of 100 the average cascade length is approximately a power-law function of the liquidity factor, with a coefficient of approximately -5/6. The observed average length in Figure 5 conforms closely to the theoretical approximation which ignores network structure and the resulting local correlations in bank states (Equation 16).

The complementary cumulative probability distribution of observed settlement cascade lengths is shown in Figure 6**.** When liquidity is abundant (l>50) only a few cascades are observed and these are not presented. In general, the majority of cascades are small and large cascades are very rare. The observed settlement cascade length distributions do not conform as closely to the mean field approximation (Equation 18) as does the average cascade length. The observed cascades follow a wider distribution with more small and large events than expected from the mean field analysis. Unlike the approximation for average length, the mean-field approximation for the length distribution depends on the assumption that the states of neighboring banks are uncorrelated.

The probability that a payment released from a queue is received by a liquid bank seems to vary as the cascade progresses. We speculate that liquid banks cluster in the network thus making the probability that a payment released from a queue will be sent to a liquid bank higher than the mean when the released payment comes from a bank "near" the originating (liquid) bank. This effect can be seen in the large number of events of length 1 relative to the mean field approximation: an instruction sent to a liquid bank often causes one of its neighbors to release a queued payment and this payment is often directed back to the original (non-queuing) bank, particularly when this exchange takes place among low-degree banks. For the first bank in the chain, the probability of sending a payment to a non-queuing liquid bank is at least 1/k, which is generally much larger than P(E) for the network as a whole. Conversely, if the chain of induced payments extends "far" from the originating bank the probability of sending to another liquid bank, and hence ending the chain, is lower than the fraction of liquid banks in the network. There are therefore more large cascades than the mean-field analysis predicts. The network sensitivity studies discussed in Section 4.3 below support this explanation.

Congestion in the primitive system with a given topology and instruction arrival process depends exclusively on the liquidity in the system. In our model congestion manifests as a high number of queued instructions, high queuing times, and a degradation of the rate at which productive agents can make payments to each other. We next examine these performance metrics.

The value of queued instructions in the quasi-steady state decreases roughly exponentially as liquidity is increased. The total value of queued instructions in relation to total deposits at different levels of liquidity is shown in Figure 7. At the lowest level of liquidity simulated, around 2.7% of deposits were held in queues. Although the fraction of all deposits in all queues is less than 3%, queuing is widespread: most banks in the system are queuing payments as can be seen in Figure 1, where only 2% of instructions arrive at banks without queues.

To analyze delays we calculate the time spent on the queue in relation to the time between instruction arrivals. We calculated the average queuing time as the ratio of the queue size to the instruction arrival rate. Because arrival rates are similar at all liquidity levels, queue size and delay time are nearly proportional. At a normalized delay time of 1 the average time in queue equals the average time between instruction arrivals. We find that this delay statistic decreases roughly according to a power law when liquidity is reduced (Figure 8). The power law relationship exhibits a roll-off at liquidity levels above 50. A delay time of 1 is achieved only at rather high levels of liquidity, between 100 and 200. The delay statistic climbs by roughly two orders of magnitude as the liquidity factor decreases from 100 to 0.1. We see that the average

delay statistic for a liquidity factor of 250 is less than 1, which is consistent with the strong correlation between total instructions and payments in Figure 4. In contrast the small correlations associated with lower liquidity factors are consistent with the long average delay times we observe.

A characteristic of the model is that congestion in the payment system slows down the instruction arrival rate, as this depends on the level of deposits available to the productive agents (Equation 10). A payment instruction to a bank reduces the rate at which subsequent instructions arrive to the bank, while a received payment increases the rate. Payment queues, however, trap deposits so that the funds are unavailable to depositors, preventing new instructions from being issued against those funds. The instruction arrival rate relative to the rate that would occur in an uncongested system is also shown on Figure 7. The reduction in payment arrival is highest when liquidity is low, at around 97.3% of the uncongested arrival rate. We see that the fractional reduction in instruction arrival rate is approximately equal to the fraction of queued payments, as expected from Equation 10.

4.2 Adding a Diffusive Liquidity Market

The liquidity market re-distributes liquidity from banks with high balances to banks with low balances or queued payments. This redistribution significantly tames the primitive payment system. Figure 9 shows the value of queued payments in a system with $l = 1$ as a function of time for several values of market conductance. The results for the primitive system from Figure 2 are also shown for comparison (Note that the time axes differ in the two figures). The market allows the system to reach a quasi steady-state much more rapidly than the primitive system. Also the value of queued payments is dramatically reduced, dropping by 49% with a conductance of 0.0001, and by 95% at with a market conductance of 0.01.

The liquidity market also reduces the frequency and size of settlement cascades. The average cascade lengths for a range of liquidity factors for three market conductance levels are shown in Figure 10. The data points from the primitive system from Figure 5 are also included for comparison. The apparent parallel shift towards the origin as market conductance increases suggests that the scaling between the liquidity factor and average cascade length is not sensitive to market conductance.

The distributions of observed cascade lengths are shown in Figure 11. The bursty character of payment releases in the primitive system is suppressed as c increases, leading to a much more compact distribution of cascade lengths where long cascades are much less frequent. The distributions appear to be geometric. However, in contrast to the primitive system we do not have an approximate analytical result for comparison.

The market reduces congestion at all values of l. The average value of payments queued relative to total deposits is shown in Figure 12A. The primitive system with a liquidity factor of 5 tends to trap nearly 1% of deposits in queues, while the same performance can be achieved with a market conductance of 0.0001 with ten times less liquidity. By reducing queuing, the liquidity market increases the availability of deposits to customers: low liquidity has a smaller impact on

the instruction arrival rate (Figure 12B**)**. The fractional decrease in instruction arrival rate is, allowing for sampling variability, equal to the fraction of deposits trapped in queues.

Liquidity flow through the market dramatically reduces the payment delay time (Figure 13) as well. For example, the primitive system would have a delay time of 1 with a liquidity factor of approximately 150. A system with a conductance of 0.01 achieves the same performance at a liquidity factor of 8.

The magnitude of liquidity redistribution can be characterized at the system scale by comparing the rate of liquidity flow through the market to the liquidity flow driven by payments. Figure 14 shows this ratio as a function of l for several values of market conductance c. The ratio varies between 9×10^{-4} and 2×10^{-2}, depending on the level of liquidity and the market conductance. Increasing l generally tends to increase market flow rates only slightly, while increasing market conductance leads to a nearly proportional increase in market flow rates.

The relative insensitivity of the market flows to changes in the level of liquidity is due to the interaction of two effects: a higher level of liquidity increases the supply of funds available for exchange in the market, but tends to reduce the demand for liquidity flow across the system. At low conductance values, market flow rates appear to scale directly with conductance. The market is less effective at low conductance values (see Figure 10) and so liquidity gradients are not greatly reduced by market flows. Increasing conductance eventually begins to improve performance and reduce balance variations. There are diminishing returns to increasing conductance beyond this point: increased market flow rate caused by a larger conductance is offset by a decrease in balance variations caused by the equalizing effect of market flows.

4.3 Sensitivity Analyses

We performed several sensitivity analyses to gauge the dependence of the results on the payment network used, and on the level of deposits. The sensitivity analyses focus on the primitive model because it includes only local interactions among banks and the outcome of these interactions may be sensitive to changes in network topology or in the instruction arrival process that stresses the system. The diffusive flux introduced by the global liquidity market dissipates the effects of local interactions: this model will therefore be less sensitive to changes in those interactions.

We explored the effects of sampling error by considering two additional realizations of the 200-node scale-free network, and the effect of network size by deriving selected results for a 1000-node network. While the scale-free character of payment system networks is supported by data

from real systems [6] we include results for a small-world network [22] to explore the effect of network topology.[5]

The differences in cascade lengths between the different realizations of the scale-free networks are rather small. There appears to be more sensitivity to network structure as liquidity decreases, both as regards the different realizations of the scale-free network or the basic network geometry. The small-world network, however, has clearly smaller cascades than the scale-free networks at all liquidity levels. All realizations follow a power law relationship between liquidity and average cascade length for three orders of magnitude. The observed average cascade length as a function of liquidity factor *l* for the alternative networks is presented in Figure 15. The expected value from the mean-field analysis is included for comparison.

The distributions of observed cascade lengths for the alternative networks in the primitive system are presented in Figure 16 for a liquidity factor of 1. Variability among different samples of the 200-node network is evident but is much smaller than the variability due to parameter variations shown in Figures 6 and 11. The frequency of long cascades is somewhat higher in the network with 1000 nodes, however the distribution is quite similar to those from with the 200-node networks, indicating that our results are not an artifact of network size. The small-world network has more small and more extremely large cascades than the scale-free networks. Also the divergence from the mean-field result is more pronounced in the small-world network. This supports the speculation that liquidity clustering contributes to the systematic departure of the cascade distributions from the mean-field distribution, as the small world network has higher clustering than the scale-free networks considered. The degree of liquidity clustering can be observed in Figure 17, which shows snapshots of the state of a scale-free and a small-world network, in which banks with liquidity are highlighted. A liquidity factor of 10 was used so that several banks in each network hold liquidity at any given time.

The relationship between liquidity factor and delay is evidently insensitive to network structure at low values of liquidity, with the maximum variation among the networks occurring at high liquidity factor values. The average instruction delay times for the various networks for a range of liquidity factors are shown in Figure 18.

As a further sensitivity analysis we explored the impact of the level of deposits for a given interbank network size. As expected, a lower level of deposits reduces the average cascade length experienced by banks in the system. The average cascade length as a function of the liquidity factor is shown in Figure 19 for two values of d_0. The smaller d_0 is, the more strongly the banks' net positions are anchored to their initial state, reducing the variability in liquidity across the network and therefore the length of cascades. Decreasing d_0 decreases cascade length

[5] The small-world network is based on a ring topology in which each node is connected to its four nearest neighbors: 2 to the left and 2 to the right. Five percent of the network links are then randomly selected, and one of the endpoints is moved to a randomly-selected node in the network. This process produces a network with strong clustering but small diameter. The simulations were carried out with the same deposits-per-link d_0 as for the scale free network, and the deposits for each bank were scaled according to Equation 11.

across the range of l values in Figure 19, although less than proportionally to the decrease in d_0. The full distribution of cascade lengths for the case of $l = 1$ is presented in Figure 20.

The effect of reducing d_0 on the average value of queued instructions is shown in Figure 21. A decrease in d_0 degrades system performance as measured by the fraction of frozen deposits: although the number of queued payments is smaller with smaller d_0, those payments represent a larger fraction of deposits. At the lowest liquidity nearly 7% of deposits are trapped in payment queues.

Finally, we looked at the delay times for payments with the two deposits levels. The average payment delay times for differently liquidity levels and two levels of d_0 are shown in Figure 22. Reducing d_0 yields smaller average delays at all liquidity levels. A smaller d_0 increases the sensitivity of the instruction arrival rate to deposits (Equation 11) and reduces the variability in the banks' net position (Appendix A). In contrast to the throughput performance measure in Figure 21 the system performance measured by delays is better at smaller d_0 because there is more liquidity per unit deposit.

5 Conclusions

We've defined and analyzed a parsimonious model of a payment system where payment instructions submitted by agents induce a stress to the system. We have used the model to understand how congestion arising from this stress is influenced by two control parameters: the global liquidity level and the conductance of a global liquidity market.

The random instruction stream stresses the system by requiring some banks to be in net deficit for some period of time. Banks will not queue, and the system will not become congested, provided banks can respond to this stress by either drawing down reserves or obtaining adequate liquidity from the market. Our results suggest that the system can remain uncongested if the time constant for applied stresses (i.e. the time for banks to return to a net position of 0) is small compared to the time to exhaust reserves, or large compared to the time to redistribute liquidity through the market. Three parameters control these time constants: d_0 determines the time to return to a net position of zero ($\tau_r \approx \sqrt{d_0}$ from simulations described in Appendix A); l determines the time to deplete initial liquidity ($\tau_l = \frac{l^2}{2p_e d_0}$); c determines the redistribution time for liquidity through the market ($\tau_m = 1/c$). A deposit level of 10000 is associated with a return time of 100: with no market we see congestion for l values of 100 and below (Figure 10). The largest conductance value of 10^{-2} has an associated equilibration time of 100. We would therefore expect congestion to occur only for depletion times smaller than 100, or liquidity factors smaller than 14. With a deposit level of 1000, and an associated return time of 30, we see congestion appearing between l=50 and l=25 (Figure 19).

When the system becomes congested payment settlement loses correlation with payment arrival. Payment settlement takes place in cascades and is governed by the internal dynamics of the

coupled payment queues. The congested state is characterized by a build-up of queued payments at banks that are short of liquidity, with episodic cascades of payment processing as the liquidity transfer from a paying bank enables the receiving bank to submit a payment from its queue. At low liquidity, cascades can affect each bank several times, and payment delays can greatly exceed the time between instruction arrivals.

The settlement cascades have a characteristic length scale. Analytical approximations for the mean size are very close to the values seen in simulations; however the analytical result depends on the liquidity distribution in the network, which must be obtained from the simulation. A mean-field approximation for the distribution of cascade sizes has too few events at both extremes of the distribution. This discrepancy is consistent with liquidity clustering in the network.

While our model is similar to both the models of Bak *et al*. [19] and of Sachtjen *et al*. [20] these models either self-organize (in the case of Bak *et al*.) or can be tuned (in the case of Sachtjen *et al*.) to produce scale-free event distributions, while cascades in our model have a characteristic length scale. The internal dissipation included in our model accounts for this difference. Queues correspond to stored energy, and each step in a cascade event dissipates a unit of stored energy. This internal dissipation induces a length scale on event sizes [23] in contrast to the Bak *et al*. formulation, where energy is only dissipated through the system boundaries, and the Sachtjen *et al*. model, which has no dissipation.

We also find that congestion in the payment system can spill over to the external economy and slow down economic activity if the submission of instructions by customers is dependent on their availability of funds. The congestion can be relieved by either increasing the global liquidity level, or by increasing the conductance of the global liquidity market. We find that less than 2% of the payment-induced liquidity flow through the global market is sufficient to achieve strong beneficial effects. This fraction is lower than the estimated overnight lending value of 30% seen in Fedwire, however the latter is motivated by many factors other than intraday funding payment operations – such as daily transformation of payment-induced liabilities towards the central bank into longer term interbank liabilities. Global liquidity sharing through a liquidity market dramatically reduces cascade lengths and payment delay times as conductance within the market increases. A liquidity market with a high conductance (i.e. low transaction costs) insures against congestion and allows global liquidity levels to be reduced by an order of magnitude for a given performance level. This frees up banks' funds for more profitable investment purposes.

Additional insights can come from further analysis of the behavior of this model, from including additional processes in the model, and from refining the representation of processes already included.

- The liquidity distribution within the network is of interest for several reasons. First, it determines the fraction of liquid banks, for which we have no analytical approximation. Second, it is another possible point of comparison with data. Third, it gives insight into the role of topology.

- We have assumed that bank size correlates with network degree. The case of well-connected banks with small deposits is plausible and may lead to qualitatively different behavior. We assume that the probability of converting a deposit into an instruction is a universal constant, and that the instruction streams are uncorrelated among depositors. The model of depositors could be elaborated to include correlations and reactions to the bank state. Further on the model does not incorporate periodic transformation of payment induced liabilities to longer term interbank liabilities - as is done in real systems through end-of-day settlement of positions. To explore the influence of such management, the current diffusive model of the liquidity market might be replaced by a dynamic network reflecting specific bilateral obligations resulting from interbank loans.

- The current model can be used to explore the effect of certain disruptions, including suspension of payment processing at one or more banks, and degradation or removal of the liquidity market. A lender of last resort can be represented as a node having a fixed liquidity potential, which would create a net source for liquidity during such disruptions.

- Banks in the model manage their liquidity in a reflexive way solely based on the availability of funds. Rather than exploring a set of alternative liquidity levels, the model can be elaborated to include adaptive behavior where banks set their individual levels of liquidity and price their interbank loans. Theoretical work [24] suggests that in a regime of high relative liquidity costs banks will tend to reduce their liquidity to inefficiently low levels.

This simple model of the necessary elements of a payment system has provided insights into the transition between an uncongested state characterized by independent processing at each bank and a congested state characterized by coupled processing across the network of banks. The model and results presented here provide a baseline from which we can explore the effects of other processes occurring in real payment systems, and against which we can assess the implications of disruptions either within the payment system, the liquidity market, or to depositors.

Notation

Table 1 lists the variables used in the model definition and analysis. Variable names obey the following conventions: Lower-case Roman (e.g. l) and Greek (e.g. λ_i) names denote constants; Upper-case Roman names (e.g. $B_i(t)$) denote random variables; Upper-case Greek names (e.g. Λ) denote sets.

Appendix A: Instruction Arrival Model

Taking the limit of Equation 10 as $D_i(0) \to \infty$, i.e. specifying an infinite pool of obligations, corresponds to a homogeneous Poisson process in which the rate of instruction arrival is constant at λ_i. The net flow of instructions to and from a bank is the sum of a series of random perturbations of unit size, which are equally likely to affect the net positively or negatively. The net flow therefore follows a random walk[6]. The probability distribution for the balance of a bank after m payment events (sent plus received) follows a normal distribution with $\mu = 0$ and $\sigma^2 = m$. A random walk process has properties that are unreasonable as a model of long-term conditions at banks. Because the variance will increase linearly with time, a bank will eventually be in an arbitrarily large positive or negative net position with probability approaching 1. The system has no stochastic equilibrium. Second, although the expected balance is 0, the expected *return time* to an initial balance of 0 is infinite. The distribution of return times follows a power law, and its expected value is unbounded [25]. Such behavior is unreasonable for real payment processes.

Assuming that instructions arise from a finite pool of obligations is a more plausible model of the real system. A long-run distribution of the balance exists and has a limiting variance. A *finite* $D_i(0)$ imposes a pull towards a net obligation change of 0: banks with $D_i(t) < D_i(0)$ receive instructions at a somewhat reduced rate (because many of their obligations have been converted into instructions) while banks with $D_i(t) > D_i(0)$ have undertaken additional obligations and so receive instructions at a higher rate. The compensating feedback between $D_i(t)$ and instruction arrival rate constrains the excursions of net position away from 0. In contrast to the homogeneous random-walk model the finite-pool model can reach a stochastic equilibrium. The equilibrium probability distribution for a bank's net position U_i can be derived by solving the detailed balance equations for the possible net positions:

$$P(U_i = k_a - D_i(0)) = e^{-D_i(0)} \frac{D_i(0)^{k_a}}{k_a!} \qquad \text{(A.1)}$$

where the non-negative index k_a defines the possible net positions. Figure A.1 shows the distributions of net position observed at a time of 1000 for the original random walk scenario and for two values of $D_i(0)$.[7] The analytical result for the equilibrium distribution is also shown for the latter two cases. Comparison with the analytical distribution suggests that equilibrium has been reached by a time of 1000. The distribution clusters around a net position of 0 due to the feedback process described above.

[6] A random walk is usually defined to be the sum of n random perturbations. Here we have the realization of independent displacements that occur at a constant frequency. Net position is not strictly a random walk in time because the time between events is random. The distinction makes no difference in our application.

[7] For a system with two banks, both having an initial frequency λ_i of 1.

Dragulescu and Yakovenko [26] describe a similar model, in which randomly-paired agents transfer a unit of money from one to the other, subject to negative balance constraints. In the simplest case where negative balances are disallowed, they find that money distributions follow the (exponential) Boltzmann-Gibbs distribution where the temperature corresponds to the average money per agent. In their model the probability of sending a unit of money is equal for all agents with non-zero balances; here the probability is proportional to the bank's deposits, reflecting the assumption that each deposit is controlled by a customer with a specified probability of issuing an instruction. This difference leads to the deposit distribution being concentrated around $D_i(0)$ rather than being exponentially distributed with an expected value of $D_i(0)$.

With a finite $D_i(0)$ the expected return time to a net position of 0 is also finite. Simulations of return time for various values of $D_i(0)$ exhibit a roll-off in the vicinity of $D_i(0)$, in contrast to the pure power-law behavior of the original random walk model (Figure A.2). Numerical simulations suggest that the return time τ_r is proportional to $D_i^{1/2}(0)$ (Figure A.3).

Making the instruction frequency sensitive to a bank's obligations introduces an automatic adaptation which balances the flow of funds over a time scale that can be adjusted parametrically using $D_i(0)$. This feature also gives us a mechanism to specify model configurations having heterogeneous banks and instruction submittal tendencies (i.e. p_e) without requiring us to enforce net balance constraints on the set of model parameters.

Appendix B: Implementation

We've simulated the evolution of the system state for each combination of parameters using a Java implementation of the following algorithm:

1. Realize a network containing the selected number of banks using the following growth algorithm modified from Barabasi and Albert [27]:
    a. For each bank…
        i. Create a node representing the bank
        ii. Sample a number of initial connections from the new node to existing nodes
        iii. For each connection, select a destination node from the set of existing nodes with a selection probability proportional to the current node degree.
    b. Initialize the state variables for all banks, which generally depend on the banks degree k_i
2. Iterate…
    a. Evaluate the instruction arrival rate at each bank using its current deposits
    b. Sample the arrival time of the next instruction for each bank based on its rate
    c. For market models …
        i. Find the instantaneous market flow rates at the current time
        ii. Set the limiting time equal to the minimum time of the next instruction arrival
        iii. Iterate…
            1. Use a semi-implicit difference approximation to solve Equation (9) at the limiting time
            2. Determine whether any bank's market participation would change before the limiting time based on Equation (7), or whether any bank's balance would increase to permit release of a queued instruction.
            3. If so
                a. estimate the minimum threshold crossing time
                b. reduce the limiting time accordingly
                c. continue iteration from (iii)
        iv. Update bank balances due to market flows
        v. If the balance changes enable release of a queued instruction, process the instruction and any other instructions enabled as a consequence, and proceed from (a)
        vi. If the balance change produces a transition in market participation, proceed from (i)
    d. Send the instruction with the minimum arrival time to the appropriate bank
    e. If a payment is submitted and the receiving bank is queuing payments, submit a queued payment, repeating as long as payments continue to be submitted

In general the state of the queue is defined by a list of instructions to pay specific banks rather than simply the number of instructions in the queue. The computational efficiency is greatly

increased by tracking only the number of queued instructions: the destination for the instruction is sampled from among the possible neighboring banks only when the payment is submitted. Deferring sampling of the payment recipient is valid as long as the probability of a particular bank being the selected does not depend on the state of the sending or receiving bank. The current model satisfies this condition.

References

[1] McAndrews,, James J. and Simon M. Potter; "Liquidity Effects of the Events of September 11, 2001". Federal Reserve Bank of New York Economic Policy Review 8, no. 2, November 2002: pp 59-79.

[2] European Central Bank; *TARGET Annual Report 2005,* European Central Bank, Frankfurt am Main, 2006. ISSN 1725-4876.

[3] Federal Reserve Board, "Peak and Average Daylight Overdrafts of Depository Institutions and Related Fees", download on Feb 17 2007 from http://www.federalreserve.gov/paymentsystems/psr/dlod.htm

[4] Bech, Morten L. and Kimmo Soramäki; "Gridlock resolution and bank failures in interbank payment systems" *in* [5], pp 149-177

[5] Leinonen, Harry (ed.); *Liquidity, risks, and speed in payment and settlemend systems – a simulation approach*. Bank of Finland Studies, 2005, ISBN 952-462-194-0

[6] Soramäki, Kimmo, Morten L. Bech, Jeffrey Arnold, Robert J. Glass, and Walter E. Beyeler; "The Topology of Interbank Payment Flows" forthcoming 2007.

[7] Inaoka, H, T. Ninomiya, K. Taniguchi, T. Shimizu, and H. Takayasu (2004). "Fractal Network derived from banking transaction - An analysis of network structures formed by financial institutions", Bank of Japan Working papers No. 04-E-04.

[8] Ponzi, A and Y. Aizawa, "Evolutionary financial market models", Physica A 287 pp 507-523

[9] Eguíluz, Víctor and Martín G. Zimmermann, "Transmission of Information and Herd Behavior: An Application to Financial Markets", Physical Review Letters, Vol 85 No 26 pp5659-5662

[10] Giardina, I. and J.-P. Bouchaud, "Bubbles, crashes, and intermittency in agent based market models", Eur. Phys. J. B Vol 31 pp 421-437

[11] Zheng, D. F., P. M. Hui, K. F. Yip, and N. F. Johnson, "Herd formation and information transmission in a population: non-universal behavior", Eur. Phys. J. B Vol 27 pp 213-218

[12] Nagurney, Anna, Jon Loo, June Dong, and Ding Zhang. "Supply chain networks and electronic commerce: a theoretical perspective," Netnomics v4 pp 187-220, 2002.

[13] Nagurney, Anna, Ke Ke Cruz, Jose Cruz, Kitty Hancock, and Frank Southworth, "Dynamics of Supply Chains: A Multilevel (Logistical/Informational/Financial) Network Perspective", Environment & Planning B, Vol 29, pp 795-818

[14] Nagurney, Anna, Tina Wakolbinger, and Li Zhao, "The Evolution and Emergence of Intergrated Social and Financial Networks with Electronic Transactions: A Dynamic Supernetwork Theory for the Modeling, Analysis, and Computation of Financial Flows and Relationship Levels", Computational Economics Vol 27 pp 353-393

[15] Nagurney, Anna, and Jose Cruz, "Dynamics of International Financial Networks with Risk Management", Quantitative Finance Vol 4 pp 279-291

[16] Dong, June, Ding Zhang, Hong Yan, and Anna Nagurney, "Multitiered Supply Chain Networks: Multicriteria Decision-Making Under Uncertainty", Annals of Operations Research, Vol 135 pp 155-178

[17] Chang, Cheng-Shang. "Stability, Queue Length, and Delay of Deterministic and Stochastic Queuing Networks," IEEE Transactions on Automatic Control, V39 No 5 pp 913-931, May 1994.

[18] Cheung, Raymond K., Warren B. Powell. "An Algorithm for multistage dynamic networks with random arc capacities, with an application to dynamic fleet management," Operations Research, Vol. 44 Issue 6 pp951-963, 1996.

[19] Bak, P., C.Tang, and K. Wiesenfeld, "Self-Organized Criticality: An explanation of 1/f noise", Physical Review Letters, 59:4:381-384, 1987.

[20] Sachtjen, M.L., B.A. Carrerras and V.E. Lynch, "Disturbances in a power transmission system", Physical Review E, 61:5:4877-4882, 2000.

[21] Axtell, Robert. "Zipf Distribution of U.S. Firm Sizes", Science, Vol. 293, pp. 1818-1820, September 2001.

[22] Watts, D. J. Strogatz, S. H. (1998). "Collective Dynamics of Small-World Networks", Nature 393, pp. 440-442

[23] Jensen, Henrik Jeldtoft; *Self-Organized Criticality: Emergent Complex Behavior in Physical and Biological Systems*. Cambridge University Press, 1998, ISBN 0-521-48371-9

[24] Bech, Morten L. & Garratt, Rod, 2003. "The intraday liquidity management game," Journal of Economic Theory, vol. 109(2), pages 198-219.

[25] Brémaud, Peirre; *Markov Chains, Gibbs Fields, Monte Carlo Simulation, and Queues*. Springer Science+Business Media, Inc. New York, 1999.

[26] Drăgulescu, Adrian and Victor M. Yakovenko. "Statistical mechanics of money," axXiv:cond-mat/0001432 v4 4 Aug 2000.

[27] Barabási, Albert-László and Réka Albert (1999). "Emergence of Scaling in Random Networks", Science, Vol. 286, pp. 509-512.

Figure 1 - Payment system components and state variables considered in the model, and their response to a payment instruction event.

Figure 2 - Average total number of queued instructions (blue symbols, left axis) and fraction of instructions that can be immediately executed (pink symbols, right axis) in successive time intervals of size 10 for a liquidity factor of 1 using the primitive model. Data for the primitive system were collected after a time of 30000 to minimize the influence of start-up transients.

Figure 3 – Total number of instructions received and total number of payments settled in 2000 time intervals of size 10 for four values of the liquidity factor. The inset shows the center region using the same scale for instructions and payments. Payments track instructions at high liquidity; reducing liquidity causes periods of congestive delay and settlement cascades so that settlements are controlled by the dynamics of the coupled instruction queues and lose correlation with instructions. The average number of instructions also decreases at low liquidity as more deposits become trapped in payment queues: the center of the distribution of instructions shifts to the left as l decreases.

Figure 4 - Correlation coefficient between total payments sent and total instructions received in 2000 time intervals of size 10 for a range of liquidity factors. As liquidity decreases payments fail to track instructions as payment processing becomes governed by the coupling among banks.

Figure 5 – Average length of settlement cascades observed in a time interval of size 20000 for several values of the liquidity factor. Line indicates the analytical approximation for the expected value of cascade size from Equation 16. The analytical approximation was only evaluated at the liquidity factor values used in the simulations: the line is a visual aid.

Figure 6 - Frequency distribution of observed cascade lengths in a time period of size 20000 for four values of the liquidity factor: 0.1 (grey symbols), 1 (blue symbols), 10 (red symbols) and 50 (green symbols) . The geometric probability distributions from the mean-field approximation (Equation 18) are shown in light grey lines for each case.

Figure 7 - Fraction of deposits in payment queues vs. liquidity factor (blue symbols. left scale) and instruction arrival rate as a fraction of the rate in an uncongested system (yellow symbols, right scale). The relative instruction arrival rate at a liquidity factor of 250 is plotted as 1 but is 1.00006 due to sampling variability.

Figure 8 – Average delay time between instruction arrival and settlement *vs.* liquidity factor. The average delay time is calculated as the ratio of the number of queued payments to the rate of instruction arrival. A

delay time of 1 means that the average time between an instruction's arrival and the ensuing payment is the same as the average time between instruction arrivals.

Figure 9 - Number of queued payments as a function of time for three values of market conductance c with a liquidity factor of 1. Results for the case with no market (dark blue symbols) are included for comparison. Higher conductance leads to smaller queues and more rapid stabilization

Figure 10 – Average cascade length as a function of liquidity factor for three values of market conductance. Results for the no-market case are included for comparison. The market model was not evaluated for liquidity factors of 2 and 0.1. Lines are a visual aid.

Figure 11 - Frequency distributions of observed cascade length with a liquidity factor of 1 for three values of market conductance. Results for the no-market case (dark blue symbols) are included for comparison. Observations were collected in a time period of 20000 for the no market case and in a period of 10000 for the simulations including the market. Increasing c significantly reduces the size of settlement cascades.

Figure 12 - Fraction of deposits in payment queues vs. liquidity factor (A) and instruction arrival rate as a fraction of the rate in an uncongested system (B) for three values of market conductance. Results for the no-market case are included for comparison. Lines are included as a visual aid. In each case the fractional reduction in instruction arrival is, allowing for sampling error, equal to the fraction of deposits held in queues.

Figure 13 – Average delay time between instruction arrival and settlement vs liquidity factor for three values of market conductance. Increasing market conductance lowers delay time with the most marked reduction seen for c larger than 10^{-4}.

Figure 14 - Liquidity flow rate through the market relative to instruction arrival rate vs. liquidity factor for four values of market conductance. Larger liquidity levels increase the liquidity available for exchange but lead to less queuing and lower gradients across the network. Increasing conductance increases flow rate at all liquidity levels.

Figure 15 - Average size of settlement cascades observed in a time interval of length 20000 for several values of the liquidity factor. Networks are distinguished by symbol. Lines indicate the analytical approximation for the expected value of cascade size from Equation 16. The analytical approximation was only evaluated at the liquidity factor values used in the simulations: the lines are a visual aid. A range of liquidity factors was used in the three realizations of the 200-node network and the small world network; the 1000-node network was only simulated for l=1.

Figure 16 - Frequency distribution of observed cascade sizes for a liquidity factor of 1 for three realizations of the 200-node scale-free network (blue symbols), one realization of a 1000-node scale-free network (orange symbols), and one realization of a 200-node small-world network (pink symbols). The geometric distribution expected from the mean-field analysis for the small-world network is shown as a light grey line.

Figure 17 – Snapshot of liquidity distribution in a scale-free network (A) and a small-world network (B) having a liquidity factor of 10. Banks with liquidity are shown as large black nodes. Liquidity appears to cluster in both network types. The small-world network has many banks far from a bank with liquidity, which tends to foster longer cascades.

Figure 18 - Average delay time between instruction arrival and payment *vs.* liquidity factor for the alternative networks considered. Only small delay times, associated with large liquidity factors, appear to be sensitive to network structure.

Figure 19 - Average size of settlement cascades observed in a time intervals of length 20000 *vs.* liquidity factor for two deposit level values. Decreasing deposits decreases cascade size.

Figure 20 – Frequency distribution of observed settlement cascades for a liquidity factor of 1 and two values of deposit level. Analytical approximations for each distribution based on Equation 18 are shown in light grey lines.

Figure 21 - Fraction of deposits in payment queues *vs.* liquidity factor for two deposit level values. Reducing deposits tends to increase the fraction of deposits queued. Although the absolute queue size is smaller with fewer deposits, the size relative to deposits is larger.

Figure 22 - Average delay time between instruction arrival and payment *vs.* liquidity factor for two deposit level values. At all liquidity levels delays are smaller with the lower deposit level.

Figure A.1 - Observed frequency distributions of 1000 observations of net position at time 1000 for the random walk (grey symbols), and the finite deposit model with $D_i(0)$ = 1000 (blue symbols) and $D_i(0)$ = 100 (orange symbols). Analytical curves for the steady-state distribution (Equation (A.1)) are included for the finite deposit model.

Figure A.2 - Frequency distributions of 1000 observations of return time τ_r for four values of the deposit size $D_i(0)$. Including a finite deposit size introduces a roll-off in the return time distribution in the vicinity of $D_i(0)$

Figure A.3 - Averages of 1000 observations of return time τ_r using various values of the deposit size $D_i(0)$. Ten averages from independent simulations are shown for each value of $D_i(0)$. The return time appears to scale with $D_i^{1/2}(0)$

Table 1 – Model Variable Definitions		
Variable	*Dimension*	*Description*
$B_i(t)$	money	Payment account balance of Bank *i*
c	1/time	System market conductance parameter
d_0	money	System deposit size parameter
$D_i(t)$	money	Deposits held by Bank *i* on behalf of its customers at time *t*
E	•	Arrival of an instruction at a bank that is able to execute the payment
$I_i(t)$	money/time	Rate of instruction arrival at Bank *i*
k_i	•	Degree of Bank *i* in the payment network
l	money	System liquidity factor parameter
L	•	Size of a settlement cascade measured by the length of chain of released instructions
$M_i(t)$	•	Indicator of market participation by bank *i* at time *t*
n_{A}	•	Number of banks in the market
n	•	Number of banks in the network
N_E	•	Number of queued payments executed
N_Q	•	Number of instructions queued
U_i	money	Net position of Bank *i*
p_e	1/time	Probability per unit time that a payment instruction will be issued against a given deposit
$Q_i(t)$	money	Payment instructions queued by Bank *i* at time *t*.
$R_i(t)$	money/time	Rate of receiving payments by Bank *i* at time *t*
$S_i(t)$	money/time	Rate of sending payments sent by Bank *i* at time *t*
$V_i(t)$	money	Total net lending by Bank *i* up to time *t*
$Z_i(t)$	money	Liquidity potential of Bank *i* at time *t*
$Z_m(t)$	money	Liquidity potential of the market at time *t*
A	•	Subset of banks participating in the liquidity market
γ	•	Power-law exponent of initial deposit distribution
λ_i	money/time	Initial frequency of instruction arrival at Bank *i*
Λ	•	Subset of banks which can execute their next instruction
τ_l	time	Time constant for depleting initial liquidity
τ_m	time	Time constant for balancing liquidity through the market
τ_r	time	Expected return time of a bank's net position

Figure 1

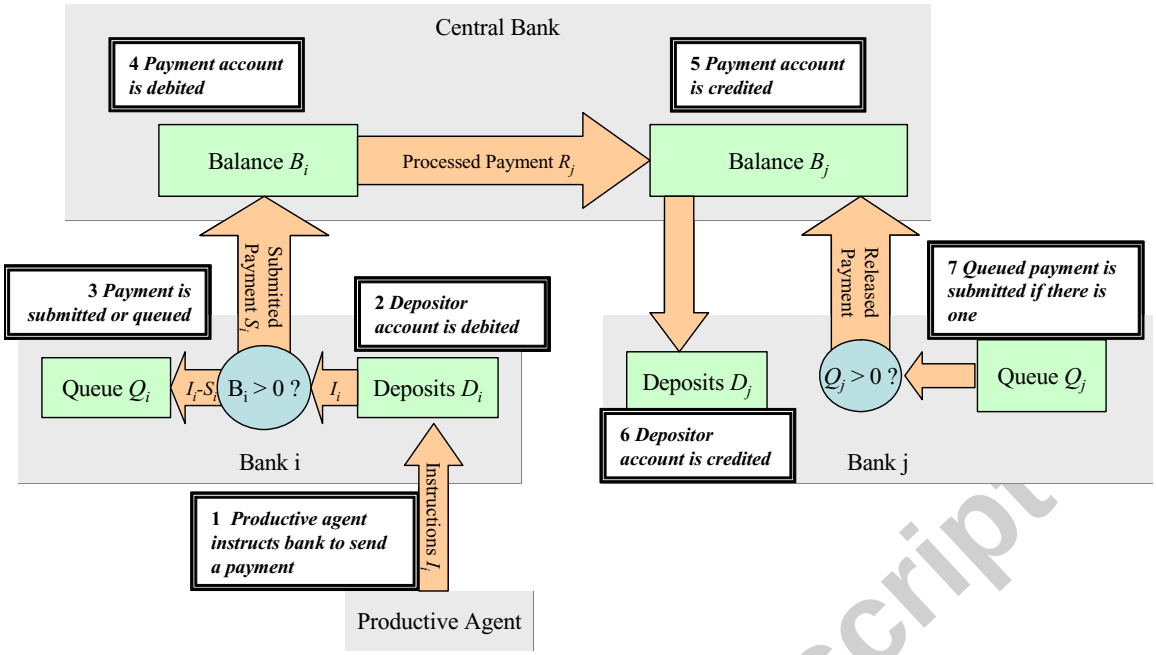


Figure 2

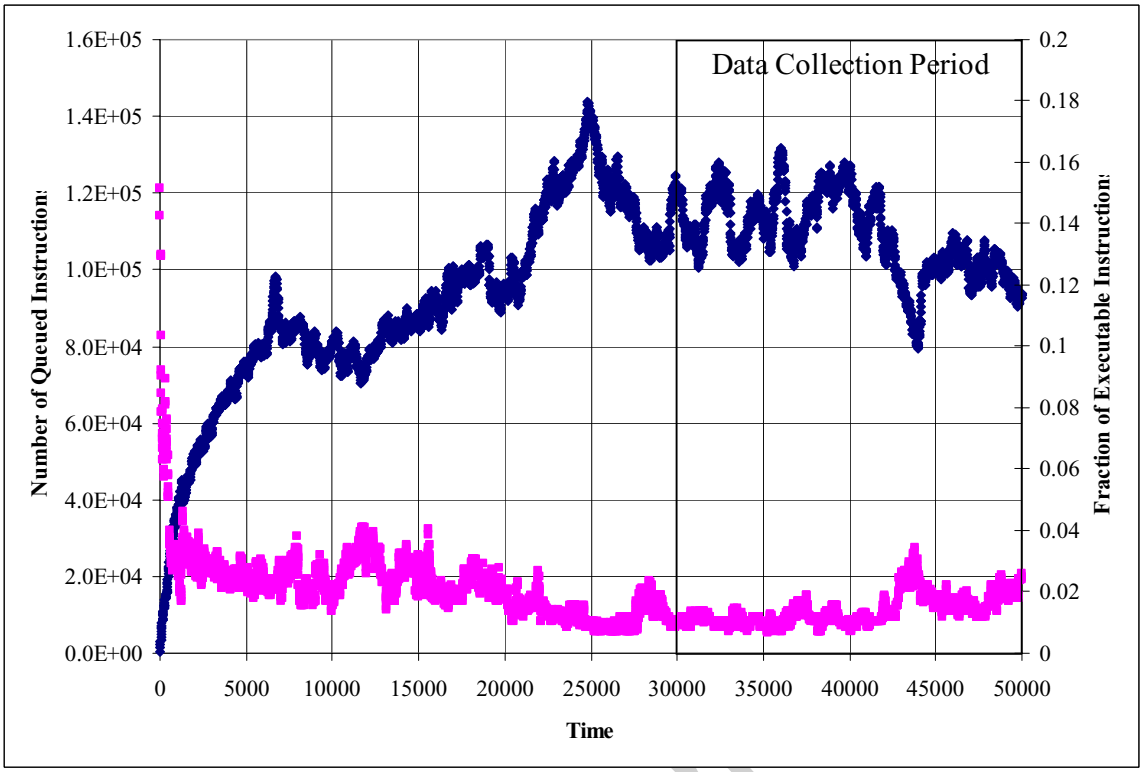


Figure 3

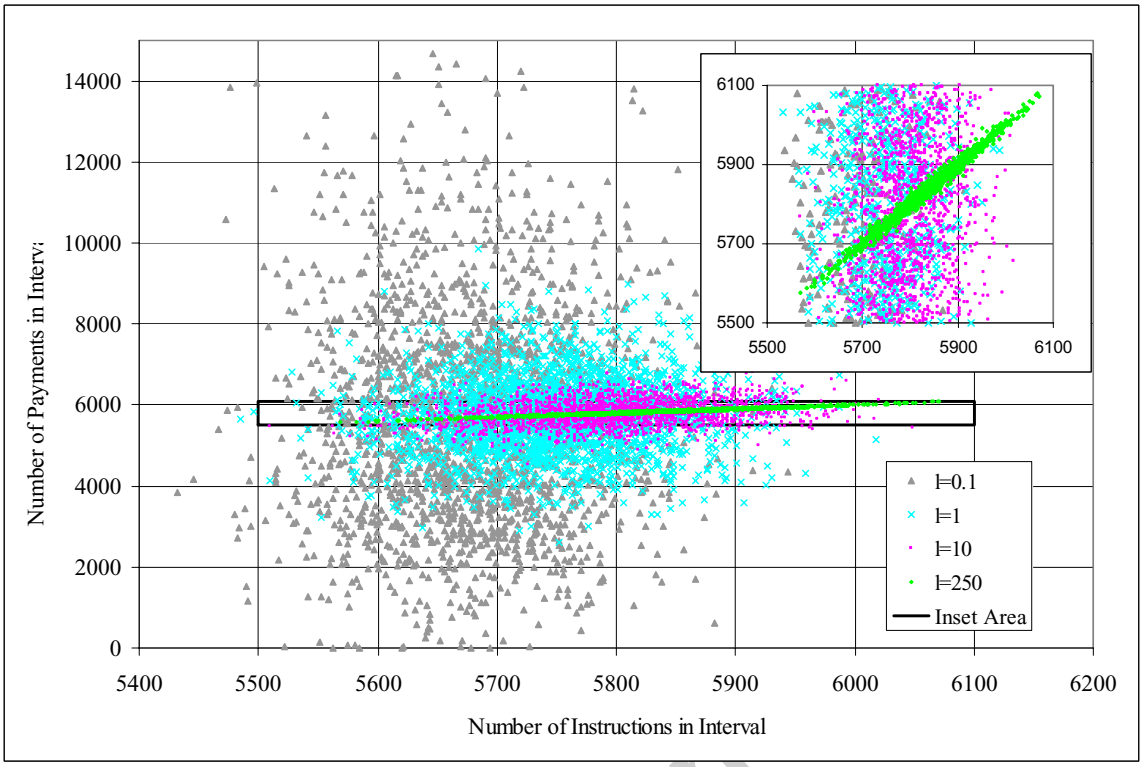


Figure 4

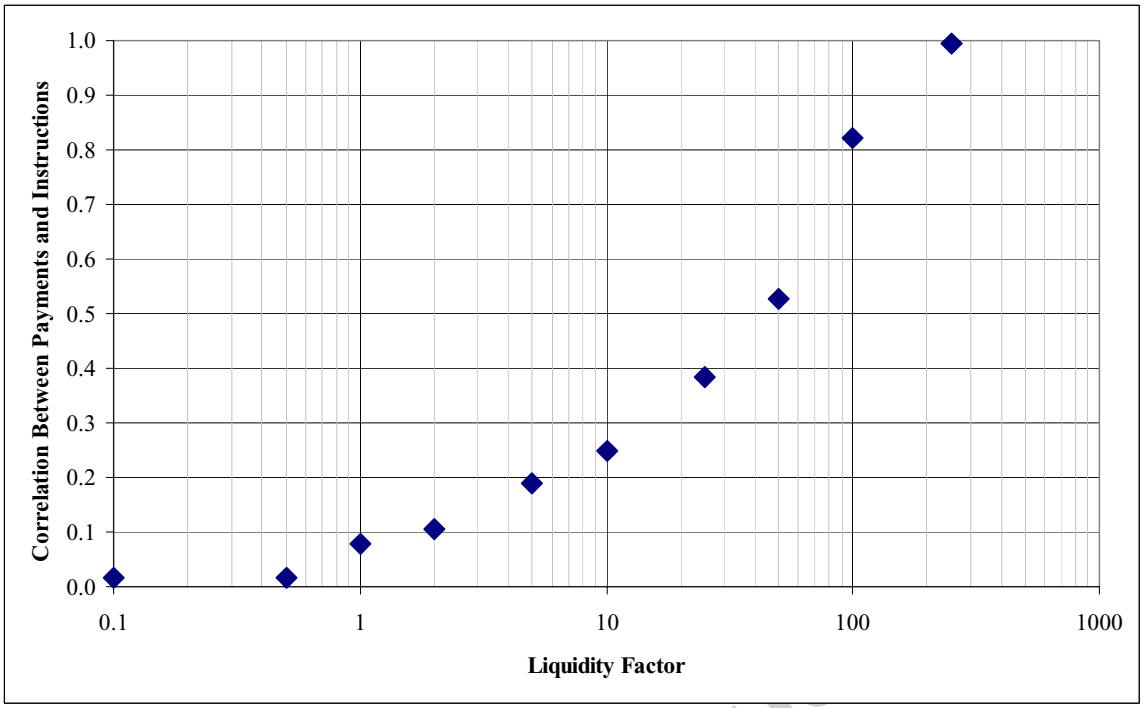


Figure 5

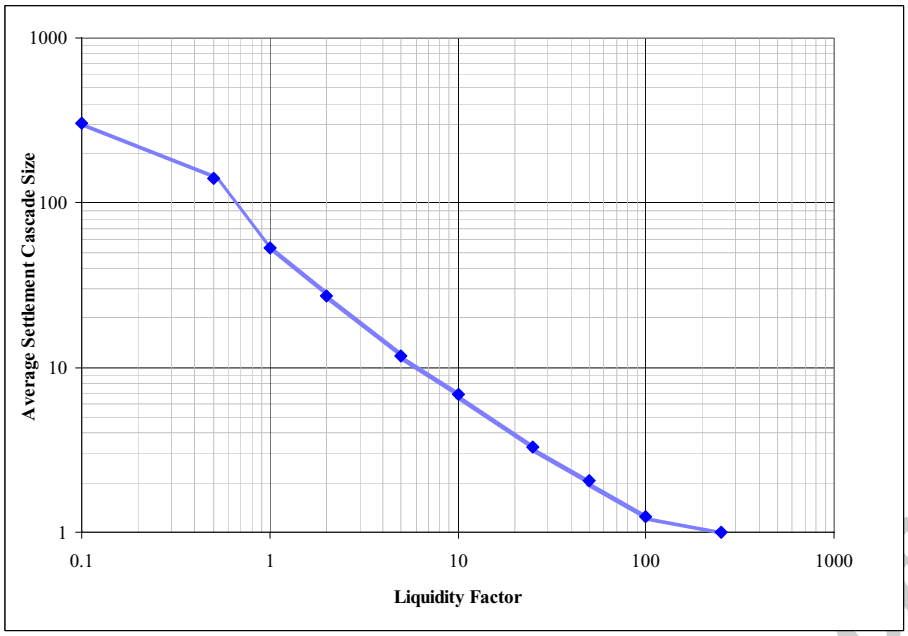


Figure 6

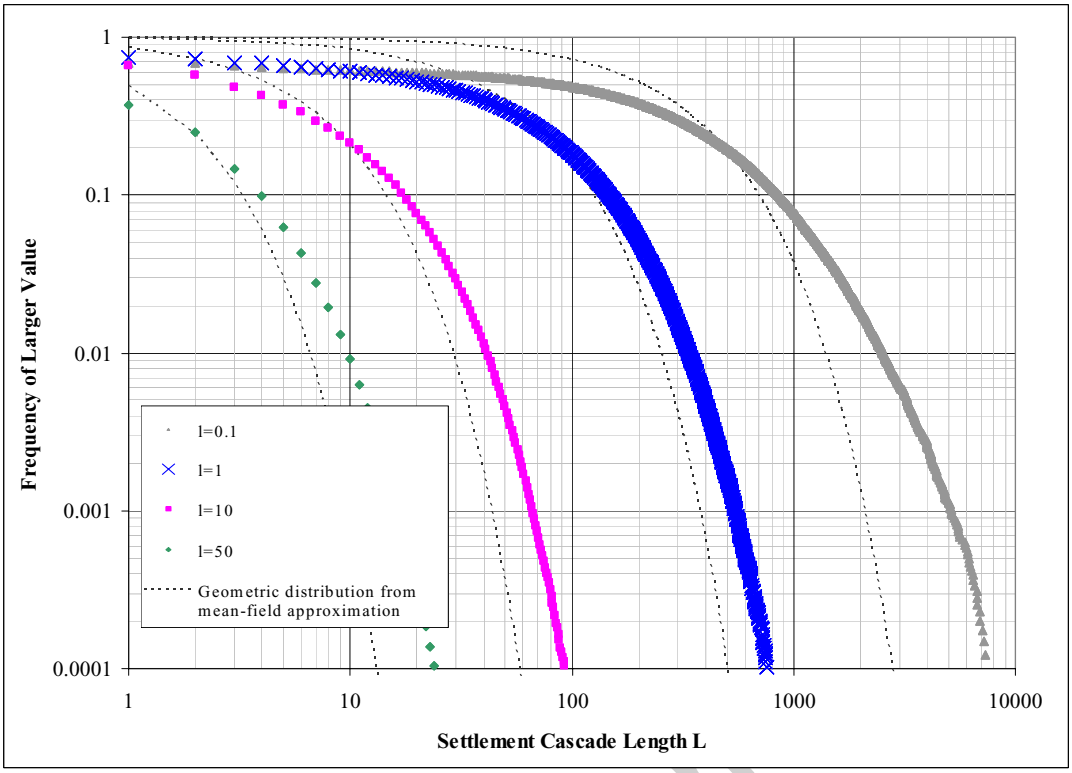


Figure 7

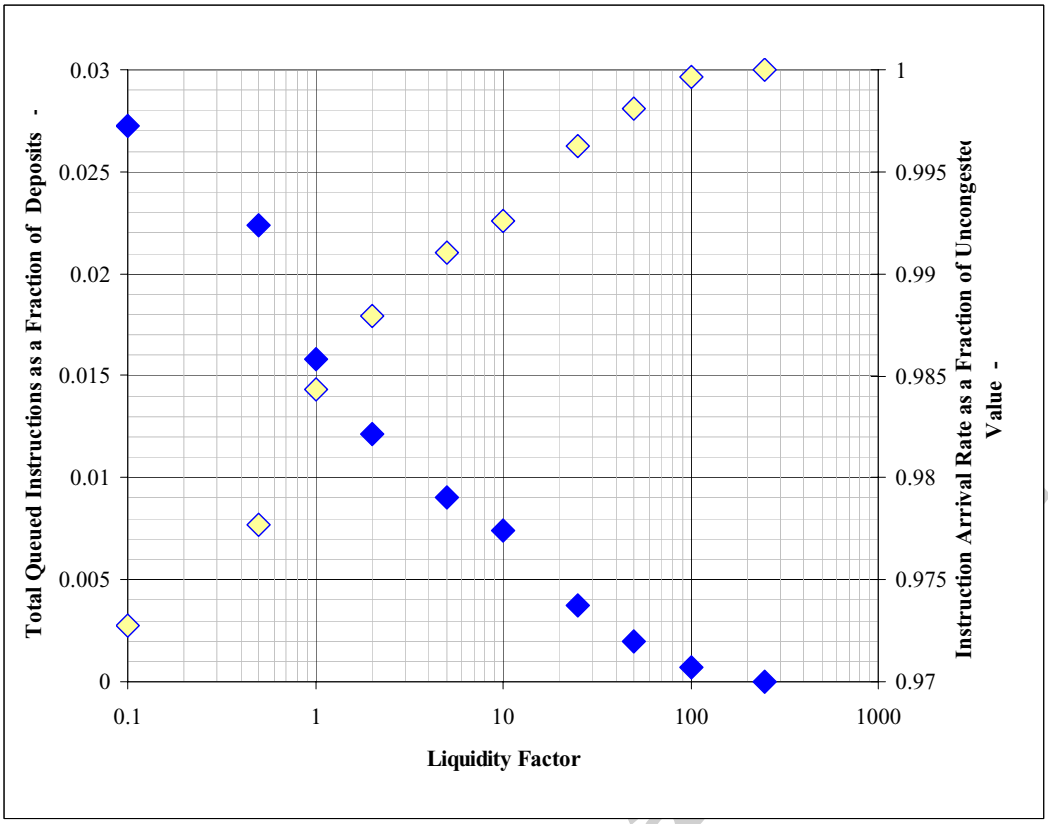


Figure 8

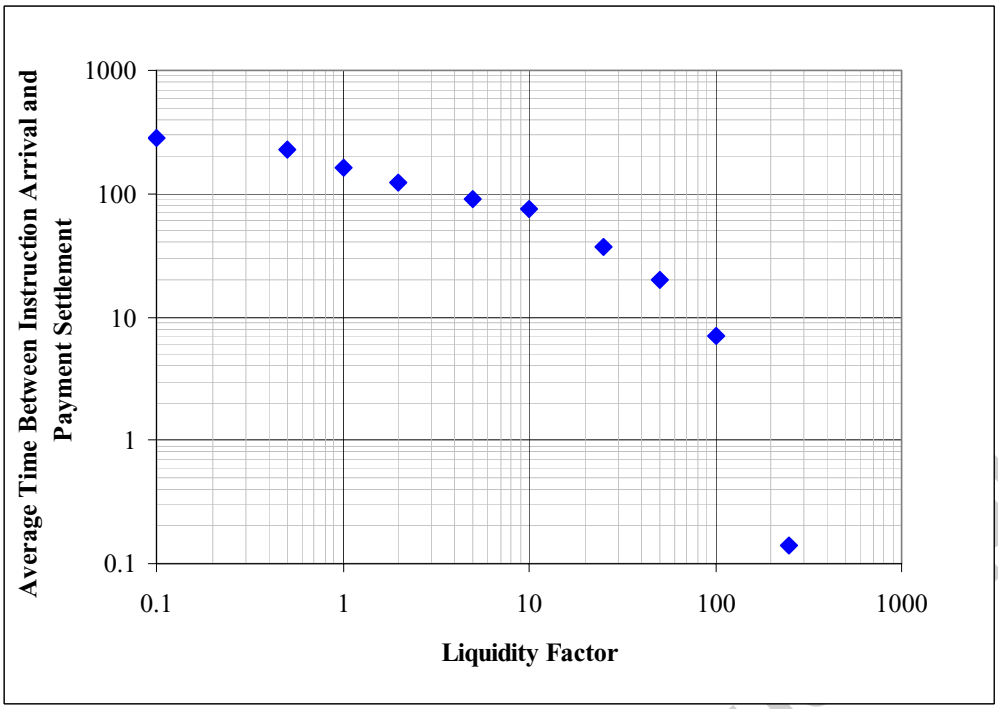


Figure 9

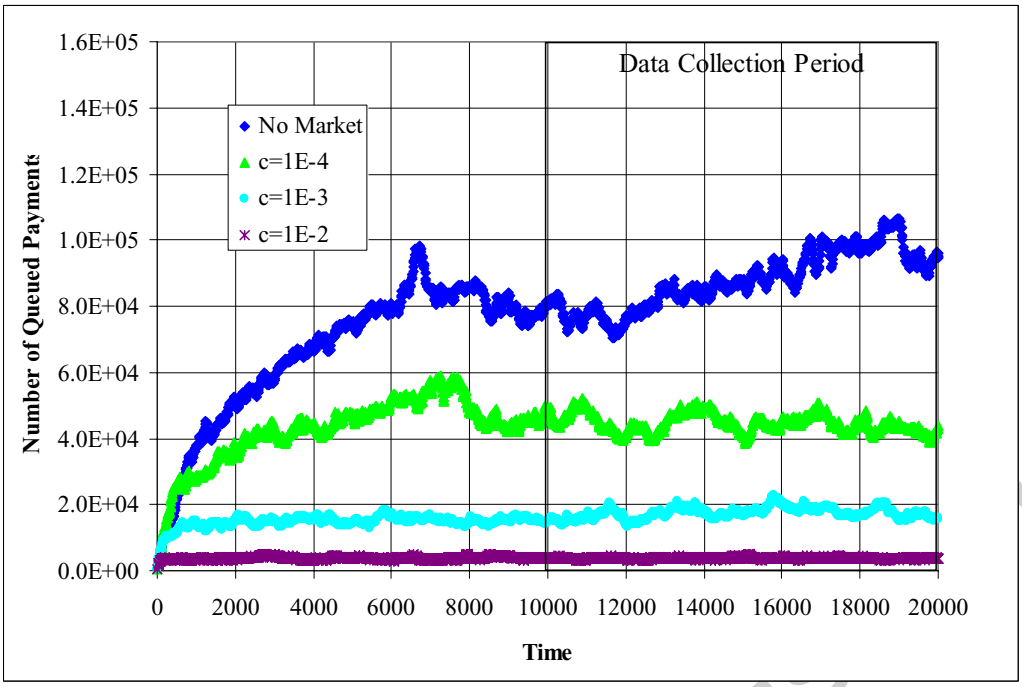


Figure 10

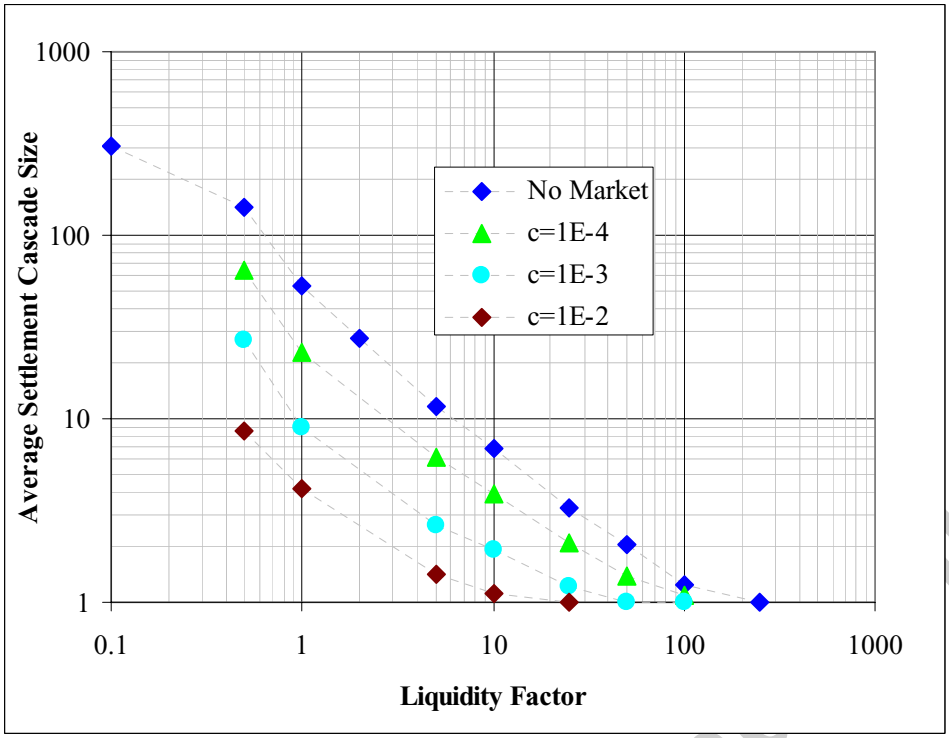


Figure 11

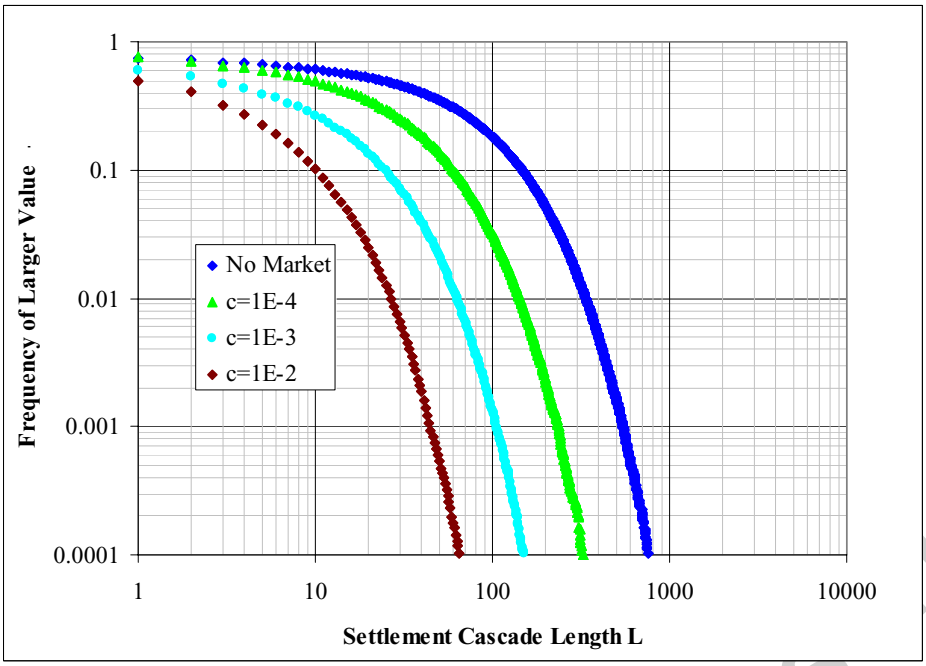
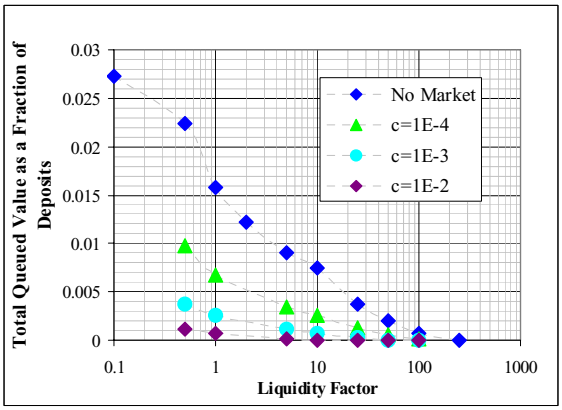
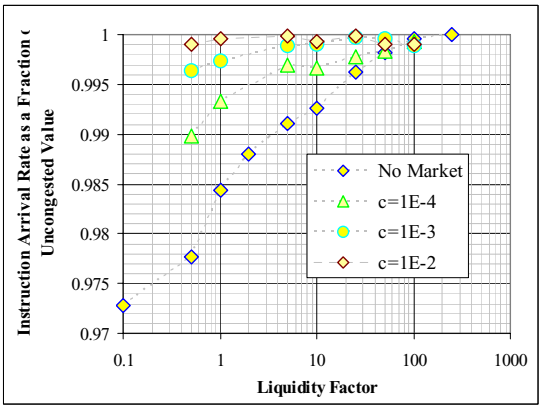


Figure 12



(A)



(B)

Figure 13

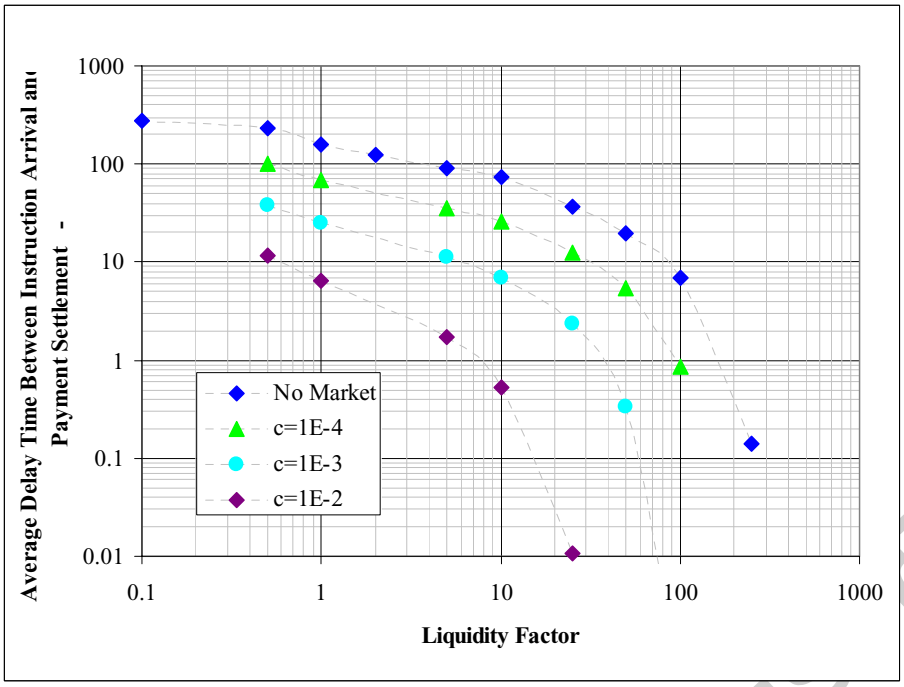


Figure 14

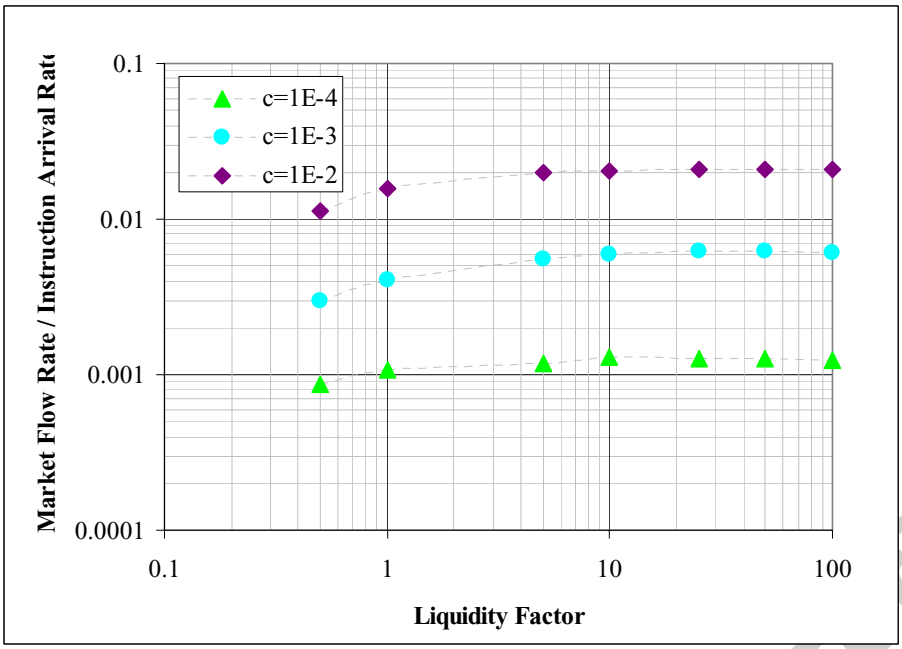


Figure 15

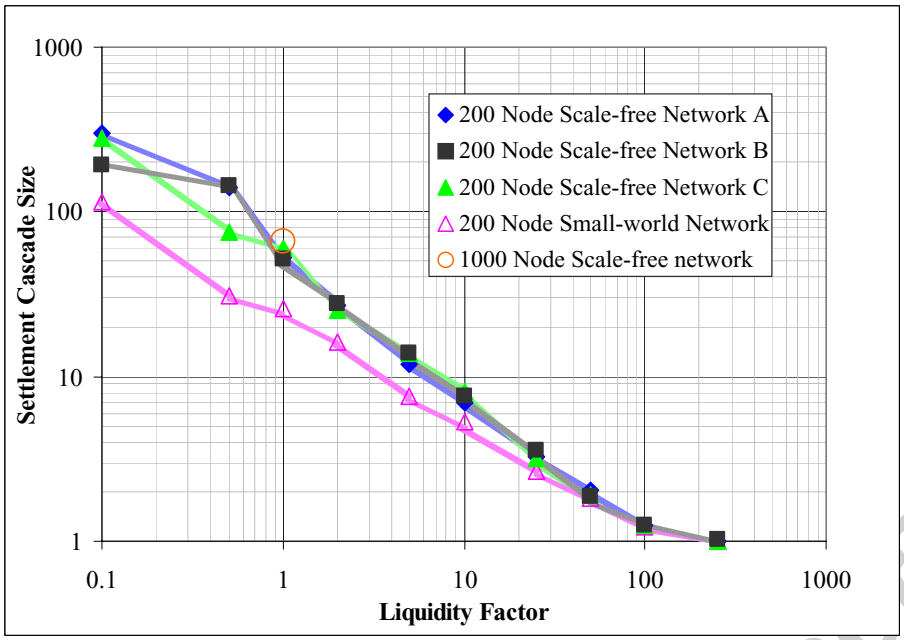
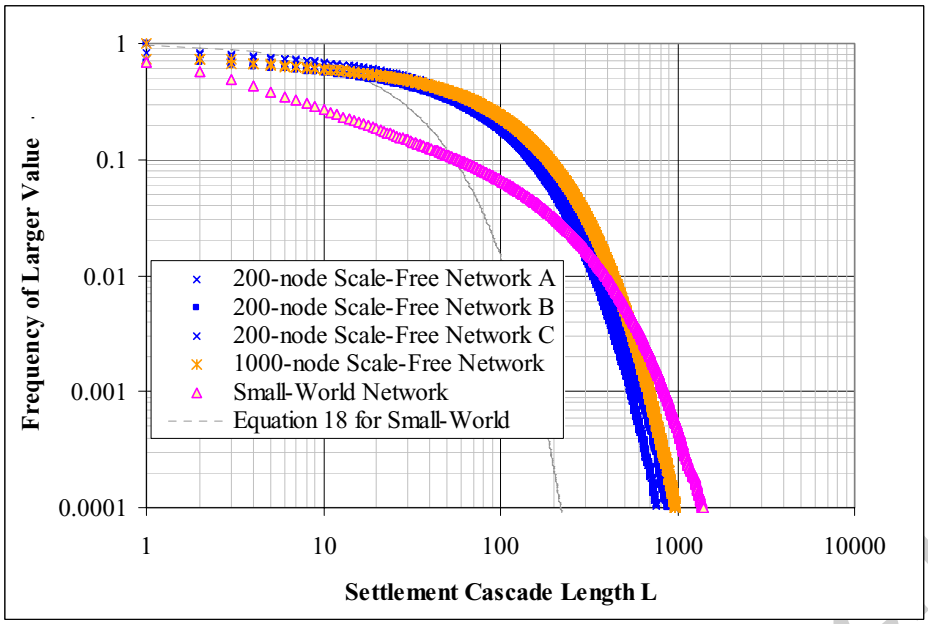
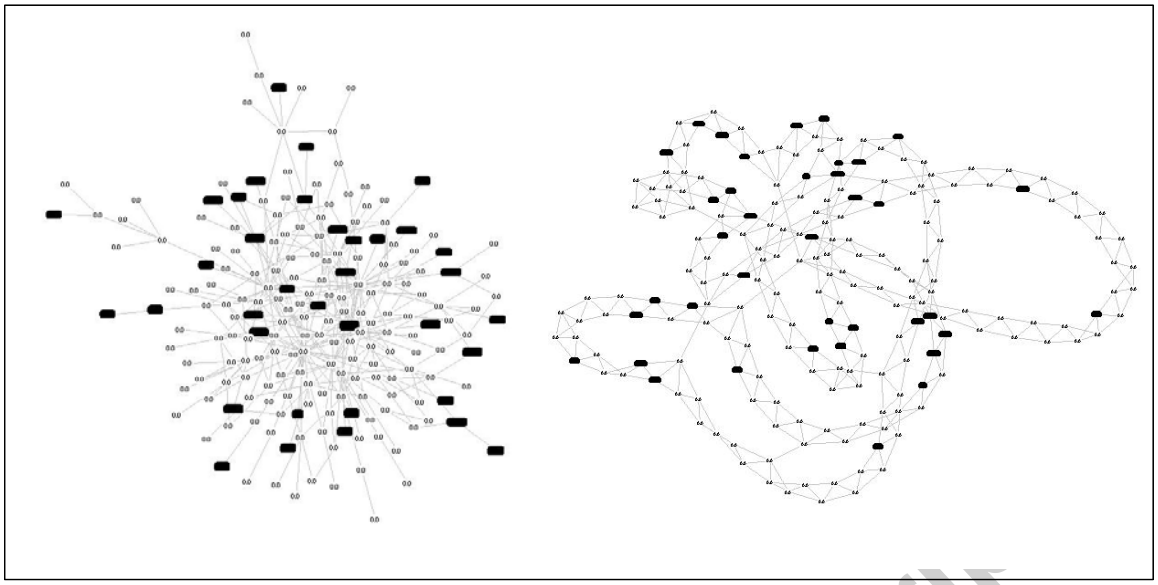


Figure 16





(A) (B)

Figure 18

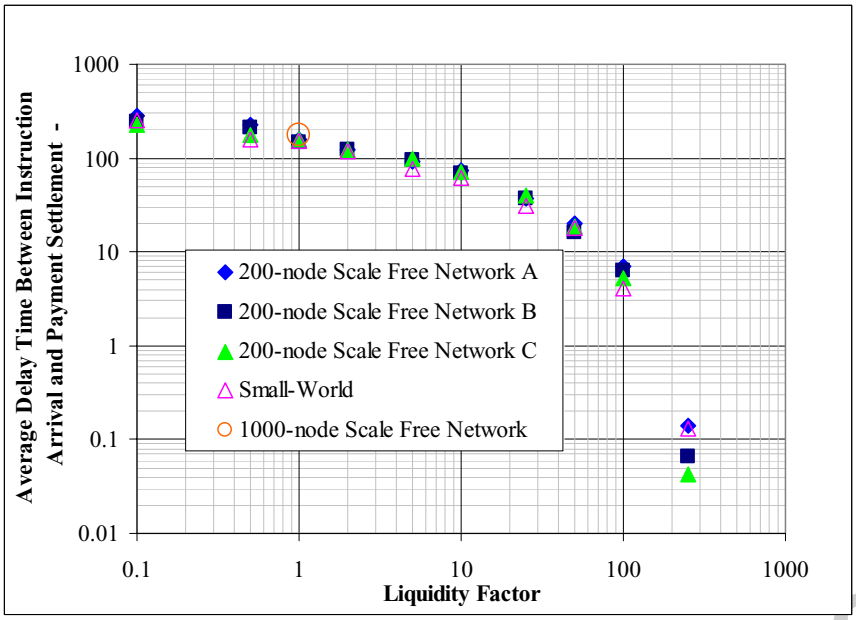


Figure 19

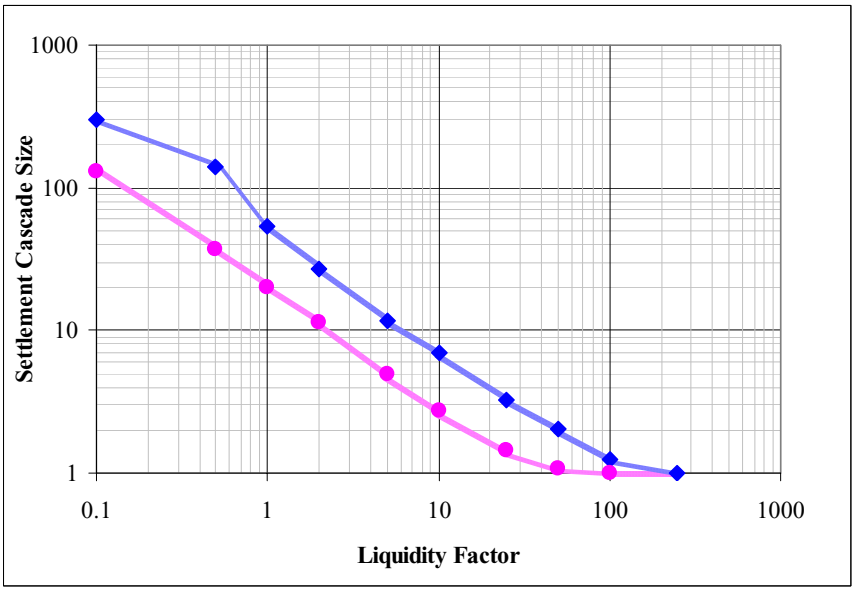


Figure 20

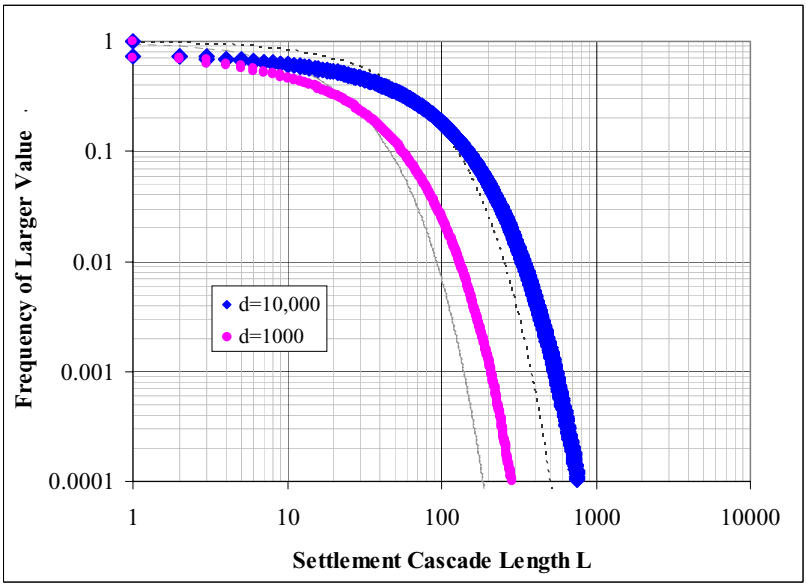


Figure 21

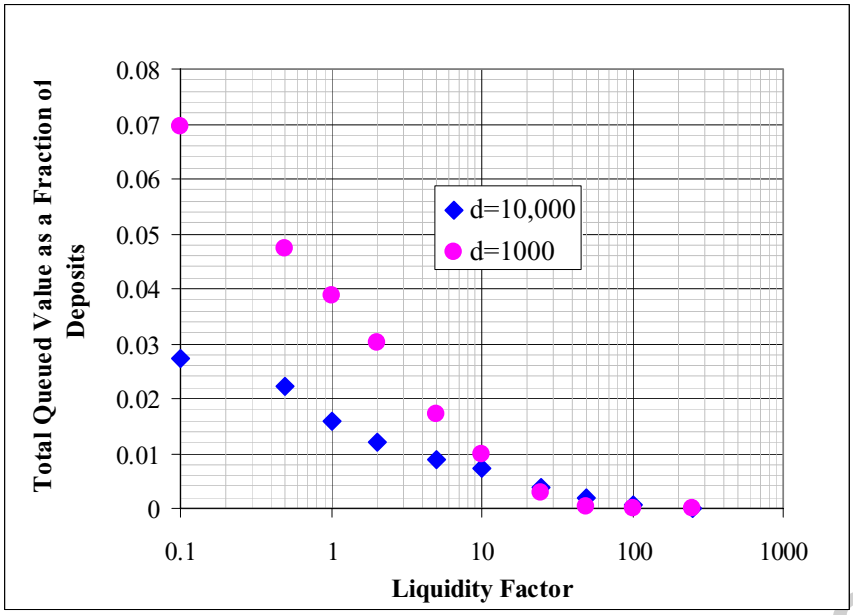


Figure 22

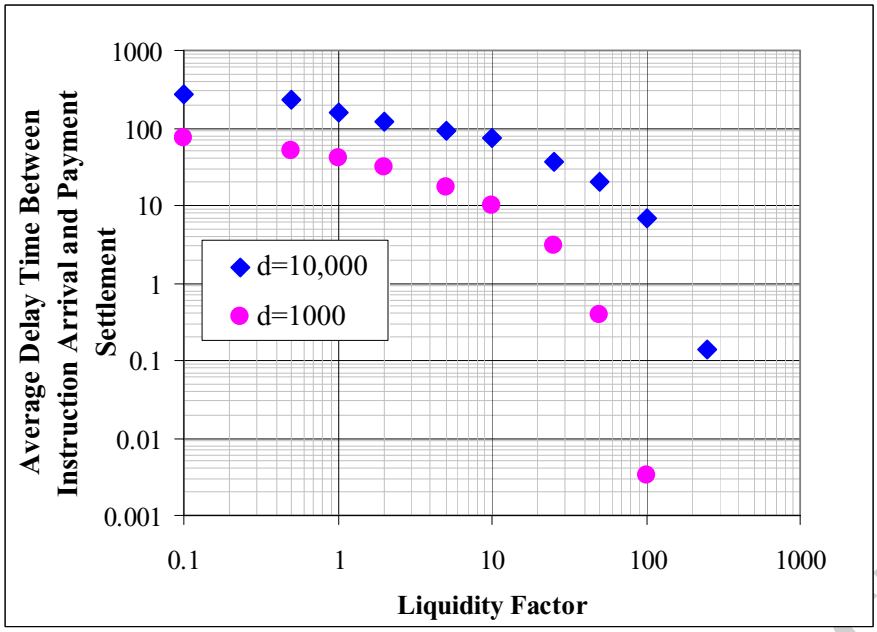


Figure A1

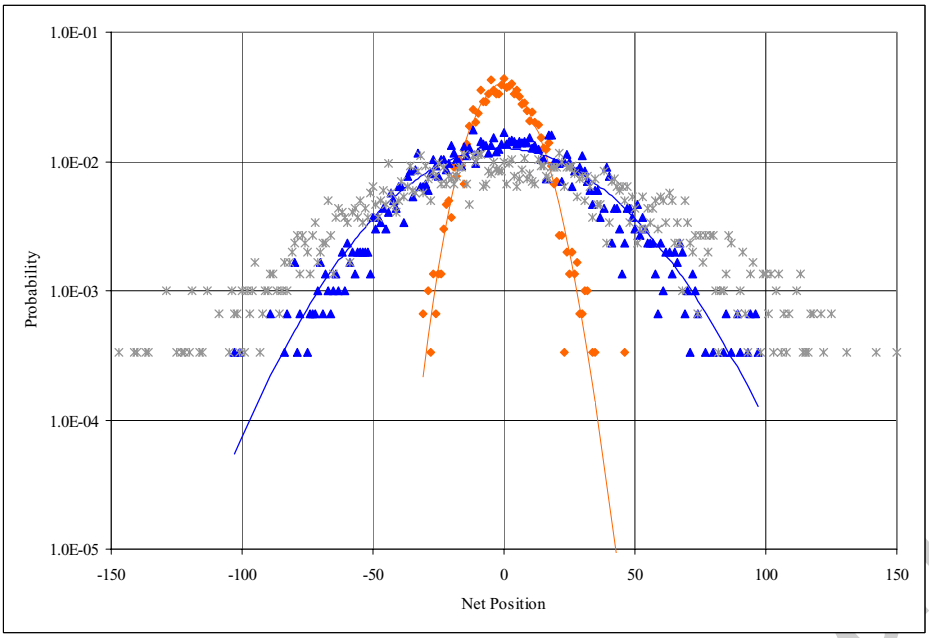


Figure A2

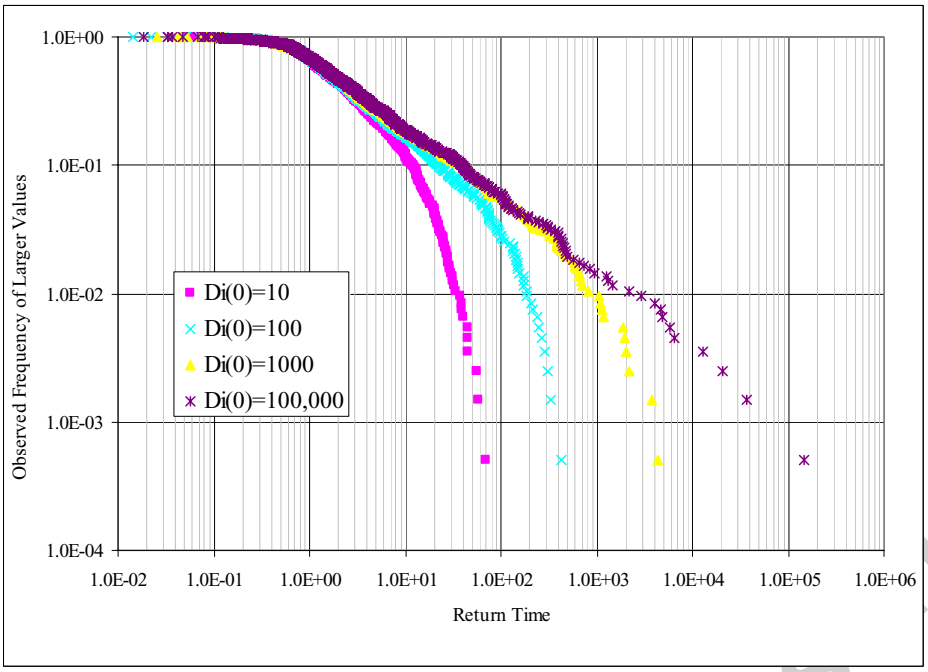


Figure A3