Paying for Payments

Rod Garratt

UCSB

16th Payment and Settlement System Simulation Seminar
Bank of Finland
August 30, 2018

The Diffusion of RTGS

- Historically, interbank payments were settled via end of day netting systems, but as volume increased central banks became worried about the risks.
- Most central banks opted for the implementation of a Real Time Gross Settlement (RTGS) system.
  - eliminates settlement risk (unwinding)
  - increased need for liquidity

Role for LSMs

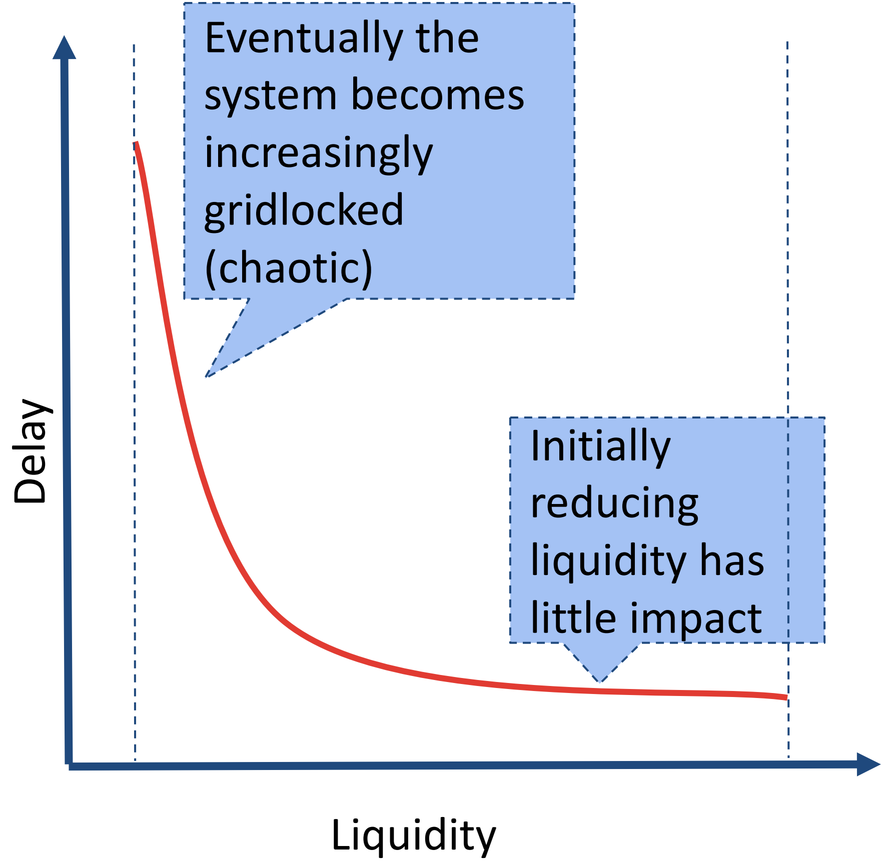
- LSMs work by either encouraging greater liquidity recycling or allowing banks to settle net obligations intraday.
- Policies that incentivize timely payment processing or punish delayers achieve the former.

Queues & Netting Algorithms

- Operate intraday
- Many design issues
  - How often to clear the queue (time or event driven)
  - Which payments to clear (if liquidity constrained)
  - Rules for managing payments in queue
  - Visibility
- Martin and McAndrews (2008), Jurgilas and Martin (2013)
- Liquidity savings are decreasing in the frequency with which net obligations are calculated
- Trade-off

A convex shape for trade-off

Bayeler, Glass, Bech, Soramäki (2007). Physica A.



Resolving the trade-off

- Operator has to make design decisions with imperfect and incomplete information
  - does not know what payments individual banks have or if they will choose to enter them into the queue
  - does not know benefits and costs

Maximize value or volume of payments settled

- Algorithms look for sets of imperfectly offsetting payments that can be settled with the available liquidity
- Finding the optimal netting solution can be very complex
  - constrained integer programming problem
  - if there are n payments in a payment file the number of netting combinations to consider is 2^n-1
  - NP-hard

“Second-best” solutions

- Relatively simple bilateral algorithms can be applied in real time, while more complex multilateral algorithms are employed intermittently at short intervals

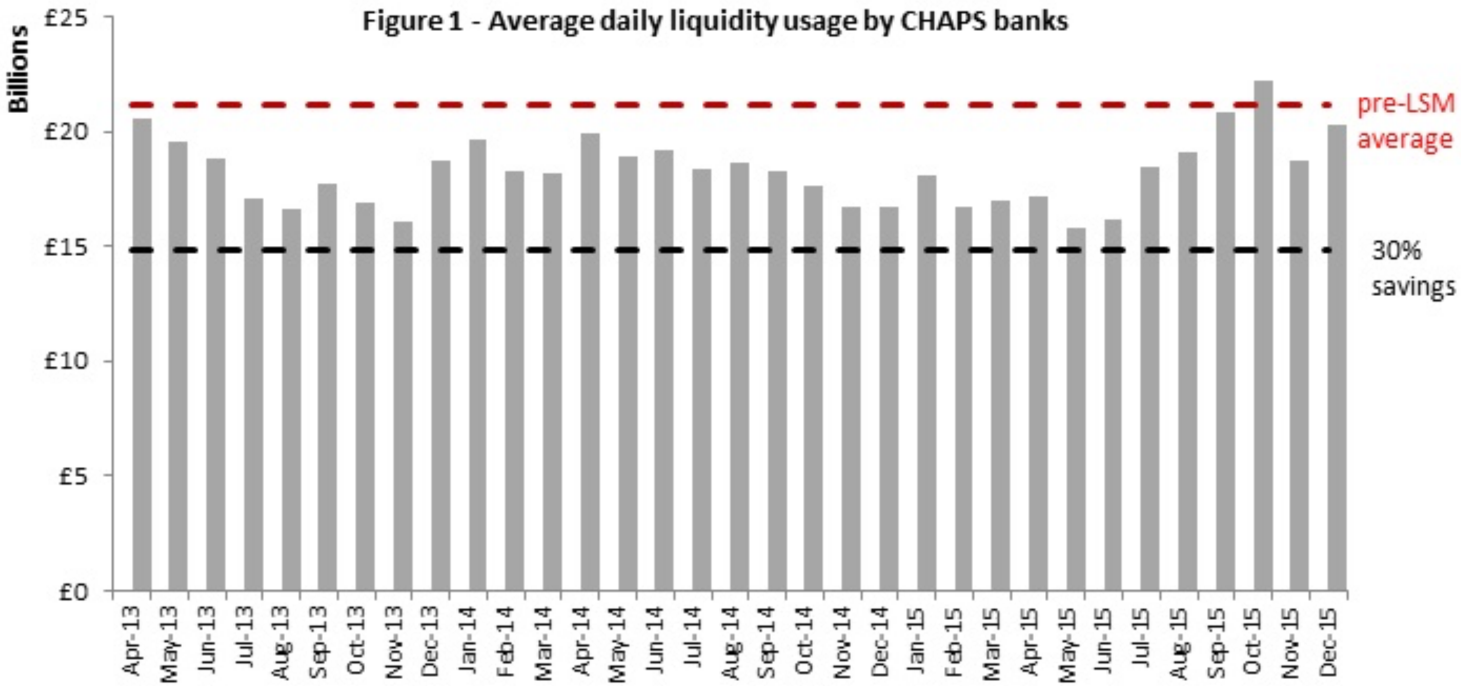
Multilateral algorithms

- FIFO-based Algorithm
  1. Calculate net debit position of each participant
  2. Identify the participant with the largest uncovered debit position
  3. Remove the latest payments of the participant identified in step 2 until its balance is no longer negative.
  4. Repeat step 3 until all participants' balances become non-negative
- Sorted-Queue Algorithm
  – Begin step 3 by sorting the payments according to some criterion (eg ascending order based on amount)

Table 1: Properties of selected centralized queuing mechanisms.

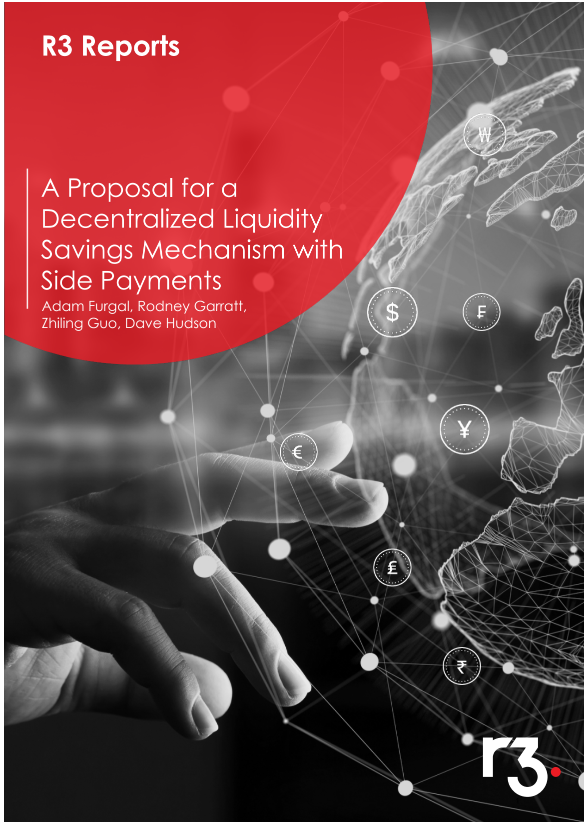
Country	System	Year of LSM	Centralized Queueing Functions				Dedicated LSM Account
			Priority of payments	Queue release method	Bilateral offsetting	Multilateral offsetting	
Australia	RITS	2009	Priority Active Deferred	FIFO Bypass	Continuous	NA	N
Eurosystem	TARGE T2	2007	Highly urgent Urgent Normal	FIFO Bypass	Continuous	Continuous	N
Korea	BOK-Wire+	2009	Urgent Normal	FIFO Bypass	Continuous	Runs every 30 minutes	N
Japan	BOJ-NET	2008	Urgent Non-urgent	FIFO Bypass	Continuous	Runs 4 times a day	Y
Mexico	SPEI	2004	High priority Normal	FIFO	Continuous	Every few seconds	N
Singapore	MEPS+	2006	5 levels	FIFO Sorted Queue	Continuous	Continuous	N
Sweden	RIX	2009	9 levels	FIFO Bypass	At certain specified intervals	At certain specified intervals	Y
Switzerland	SIC	2008	High Norm	FIFO	Every few seconds	NA	N
U.K.	CHAPS	2013	High priority Non-urgent	FIFO Sorted Queue	Every 2 minutes	Every 2 minutes	N

Assessing performance: Eg CHAPS



Source: Seaward (2016, The OTC Space)

Decentralized approach

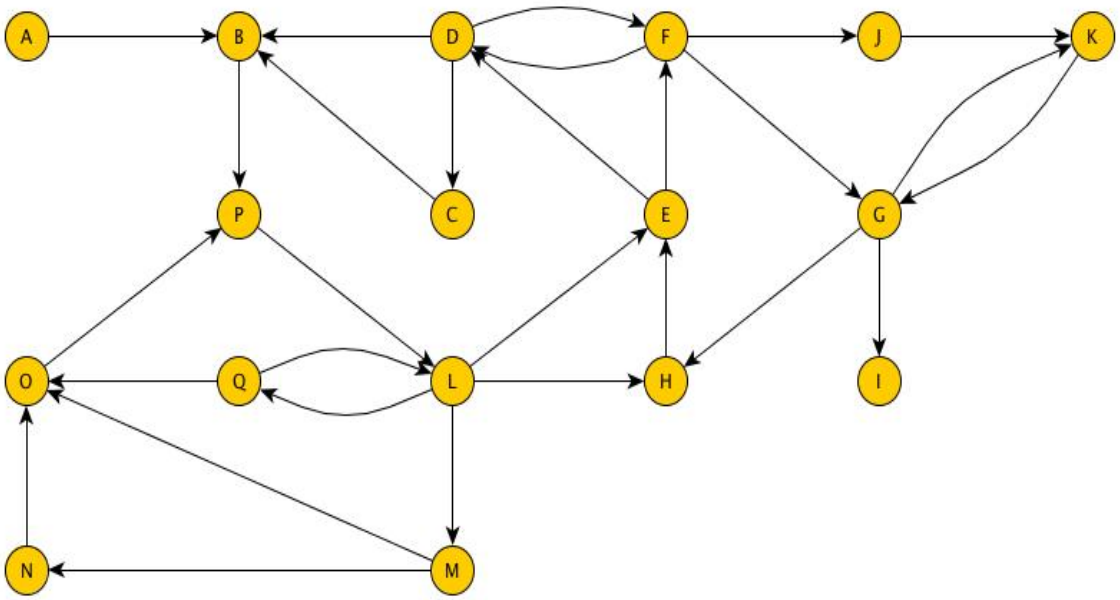


- avoid NP-hard problem or arbitrary “second-best” solutions
- banks make proposals that reflect current conditions
- expost cost-benefit assessment
- competitive proposals that reflect market conditions likely to be welfare enhancing

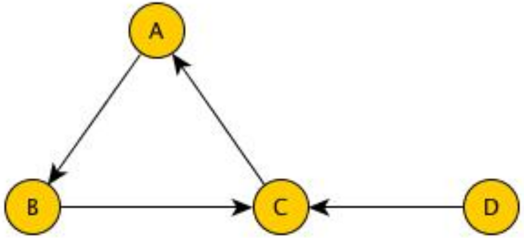
2-stage process

1. Detect obligations between various participating nodes
2. Plan and execute a strategy to successfully meet payment obligations on a net basis
   - Includes proposal for liquidity provision
   - May involve side payments

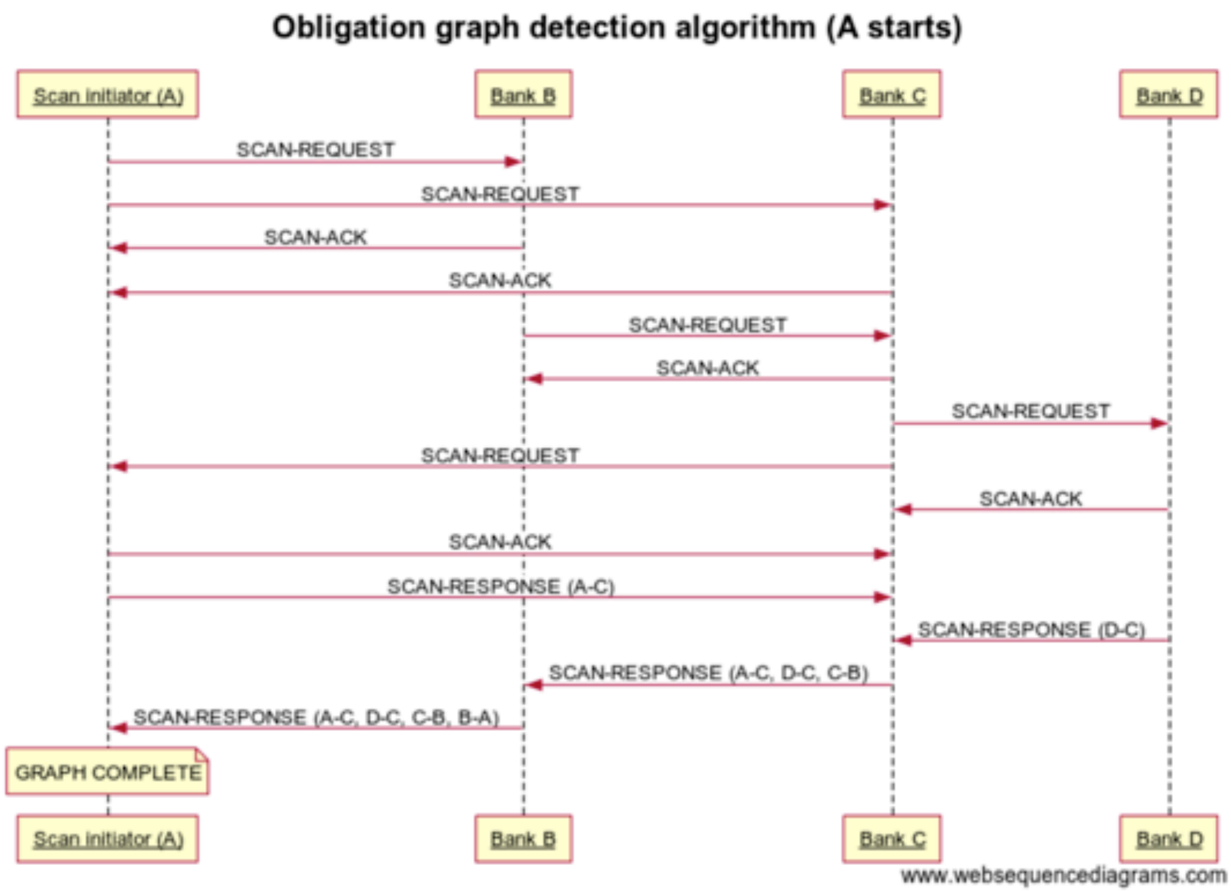
Stage 1: Detecting the obligation network



Recursive graph scanning algorithm: simpler example



- A initiates the scan and request nodes B and C continue it.
- Node B will subsequently request that node C continue the scan, while node C will request both nodes B and D continue.
- Node D will respond with the details of the edge DC.
- Either node B, C or A may take different actions, dependent on the timing of the scan, but node C will bundle the reply from D regarding edge DC, while adding details of the edge CA. Node B will offer details of the edge BC
- Node A will collate the replies from B and C, forming a view of a graph with edges AB, CA, BC and DC.


Obligation graph detection algorithm (A starts)
Scan initiator (A)
Bank B
Bank C
Bank D
SCAN-REQUEST
SCAN-REQUEST
SCAN-ACK
SCAN-ACK
SCAN-REQUEST
SCAN-ACK
SCAN-REQUEST
SCAN-REQUEST
SCAN-ACK
SCAN-ACK
SCAN-RESPONSE (A-C)
SCAN-RESPONSE (D-C)
SCAN-RESPONSE (A-C, D-C, C-B)
SCAN-RESPONSE (A-C, D-C, C-B, B-A)
GRAPH COMPLETE
Scan initiator (A)
Bank B
Bank C
Bank D
www.websequencediagrams.com

Scan complexity

- The scan complexity is a function of the worst-case graph walk
- Worst-case scenario is a purely linear graph, initiated by one of the least-connected nodes
  - complexity is O(n) (twice linear), where n is the number of nodes in the graph
- More typically the complexity is O(log(n)) as we are performing a tree scan

Stage 2: Plan and execute

- Given knowledge of the network a node can make a netting proposal
  - payments to be settled
  - liquidity provision
- Each node signs off that the payments settled are legitimate and that it is willing to make/receive associated payments
- Need unanimous agreement

Liquidity provision

- Constrained integer programing problem is replaced by take-it-or-leave-it offers
- Only make netting proposals that have positive surplus
  - participants in best position to recognize these opportunities
- Side payments increase fairness and are <u>needed</u> to get all mutually beneficial proposals accepted

No side payments

- Suppose Bank A owes Bank B $100 and Bank B owes Bank A $80
- Bank B has observed this netting opportunity and makes a proposal to Bank A

Default Proposal

Bank A provides $20 in liquidity to settle both payments.

Is that fair? Will Bank A accept?

Cost allocation problem

- Lloyd Shapley and Al Roth shared 2012 Nobel prize in economics for their work on matching
- Also made significant contributions to theory of cost sharing
  - Shapley value is a method of joint-cost allocation
  - Roth and Verrecchia (1979) provide a justification in terms of bargaining

Cost allocation problem cont…

- Suppose each bank has a cost of providing liquidity that is equal to some parameter *c* times the amount of liquidity provided.

$$c = .1$$

- Benefit is the bank's gain from timely settlement and call this *b*.

$$b = .05.$$

Cost allocation problem cont…

- Let the *characteristic function* v describe the maximum total surplus any group of banks can achieve by netting payments.

$$v(A) = 0, v(B) = 0$$

$$v(A, B) = .05 * 180 - .1 * 20 = 7$$

- Shapley value for each bank is the average over all orderings of its marginal contributions to the total surplus.

Solution

Order	Marg. Contrib. A	Marg. Contrib. B
A,B	0	7
B,A	7	0
Shapley Value	**3.5**	**3.5**

The cost share of each bank is equal to the benefit the bank receives from having its individual payments settled minus its Shapley Value.

Cost Allocation = benefit minus Shapely value

Bank A	**Bank B**
$.05 * 100 - 3.5 = 1.5$	$.05 * 80 - 3.5 = .5$

Achieved by having bank A provide \$20 in liquidity at cost \$2 and then having bank B make \$0.5 side payment to bank A

Remark

- Here side payments make solution fair

Net benefits with and without side payments

	Bank A	**Bank B**
With	$.05 * 100 - .1 * 20 + \mathbf{.5} = 3.5$	$.05 * 80 \mathbf{- .5} = 3.5$
Without	$.05 * 100 - .1 * 20 = 3$	$.05 * 80 = 4$

b=.025, c=.15

Net benefits with and without side payments

	Bank A	**Bank B**
With	$.025 * 100 - .15 * 20 + \mathbf{1.25} = .75$	$.025 * 80 - \mathbf{1.25} = .75$
Without	$.025 * 100 - .15 * 20 = -.5$	$.025 * 80 = 2$

- Here side payments are <u>essential for acceptance</u>.

3 banks, multiple payments

A owes B \$70 and C \$110, B owes A \$10 and C \$30, C owes A \$60 and B \$30.

$v(j) = 0$, $\mathrm{v}(\mathrm{A}, \mathrm{B}) = 0$, $v(B, C) = 3$, $v(A, C) = 3.5$,
$v(A, B, C) = 6.5$ (excludes payments of \$70 and \$10)

Order	**Marg. Cont. A**	**Marg. Cont. B**	**Marg. Cont. C**
A,B,C	0	0	6.5
A,C,B	0	3	3.5
B,A,C	0	0	6.5
B,C,A	3.5	0	3
C,A,B	3.5	3	0
C,B,A	3.5	3	0
Shapley Value	**1.75**	**1.5**	**3.25**

Solution: 3 banks

Cost Allocation

Bank A	Bank B	Bank C
$.05 * 110 - 1.75 = 3.75$	$.05 * 30 - 1.5 = 0$	$.05 * (60 + 30) - 3.25 = 1.25$

Achieved by having bank A provide \$50 in liquidity at cost \$5 and then having bank C make \$1.25 side payment to bank A .

n banks, etc.

- n banks
- different benefits and costs
- payments to multiple banks
- multiple payments to same bank

That is, $v(S)$ is computed as the sum of the accrued benefits from payments settled minus the combined costs of supplying the liquidity required to settle them. Then, following Shapley (1953) the Shapley value for bank i is given by

$$w_i = \sum_{S \subset N} \frac{(s-1)!(n-s)!}{n!} \left[v(S) - v(S-i) \right], \tag{4}$$

where s is the number banks in group S. The Shapley value of bank i is the weighted sum of the terms $[v(S) - v(S-i)]$, which represent bank i's marginal contribution to coalition S. It can therefore be interpreted as a bank's expected marginal contribution to a coalition of banks that seek to net payments, based on the assumption that each bank's sequential arrival to the coalition is determined randomly.

Each bank i's share of the cost burden of providing liquidity to settle the payments in P, denote this by C_i^P, is then defined as the gross benefit to bank i of having its own payments in P settled minus its Shapley value:

$$C_i^P = \sum_{j \neq i} \sum_{k,h} b^h \times p_{ijk}^h - w_i. \tag{5}$$

A property of the Shapley value is that it assigns the total value of the coalition to its members (efficiency), so that the sum over all banks of the terms in (4) equals $v(N)$. It follows that the sum of the cost shares in (5) over all of the banks cover the actual cost of the proposal: ie $\sum_{i=1}^n C_i^P = \sum_{i=1}^n c^i d_i$.

To actually implement these cost shares we propose the use of side payments. That is, each bank provides liquidity equal to its net debit position and then side payments are made to so that the final cost share to each bank equals the amount specified by (5). Formally, side payments from bank i to bank j associated with any solution of the type given by (5) can be defined as

$$SP_{ij} = \begin{cases} 0 & \text{if } d_i > 0 \\ (c_j d_j - C_j^P) \frac{C_i^P}{\sum_{s:d_s=0} C_s^P} & \text{otherwise.} \end{cases} \tag{6}$$

As a simple (but not overly trivial) illustration let us consider three banks: A, B and C. Assume bank A owes bank B \$100, bank B owes bank C \$80 and bank C owes bank A \$70. Bank A has a net debit position of \$30 and both bank B and bank C have net credit positions of \$20 and \$10, respectively. $V(S) = 0$ for all $S \subseteq N$ except $S = N$. When all three join together there is a netting cycle that clears \$250 worth of payments with \$30 in liquidity. Assume that all the payments are the same priority and that the benefit per dollar to each bank of settling those payments at the current instant is $b = .05$. In addition, assume that the instantaneous, per-dollar cost of liquidity provision is the same for all banks and is equal to $c = .1$.[18] Then the total net benefit that can be obtained by the three banks by settling their combined payment obligations on a net basis is equal to $v(A, B, C) = .05 \times \$250 - .1 \times \$30 = \$9.5$.

To compute the Shapley value for each bank, we list all the possible orderings of the banks and then take the average over all orderings of the marginal contributions of each bank to the total net benefit.[19] Since there are three banks, there are six possible orderings as shown in Table 2.

The calculation of the marginal contribution can be understood as follows. Let us take the order A,B,C as an example. If bank A pays \$100, the benefit is \$5 and the cost is \$10. So bank A would not choose to clear its payment obligation to bank B. The marginal contribution of bank A under this ordering is therefore \$0. Next, if both bank A and bank B form a coalition, the total settlement benefit is \$9 and the total liquidity cost is \$10. So they would choose not to clear their payment obligations. The marginal contribution of bank B is therefore also \$0. Finally, if all three banks form a coalition, the total benefit of clearing their obligations is \$12.5 and the total liquidity cost is \$3. With a net gain of \$9.5, all three banks would choose to clear their obligations. The marginal contribution of bank C is thus \$9.5.

[18] Both b and c should be small relative to magnitudes of liquidity provided. They only reflect the costs and benefits of settlement now versus an uncertain point in the near future. It is also reasonable to set b smaller than c. This is consistent with assumptions made in Bech and Garratt (2003, 2012). If it were not, then one might expect the bank to settle the payment via the standard RTGS stream.

[19] This is the same computation as (4).

Improving centralized solutions

- Suppose proposals come from system operator (eg central bank)
  - Eliminates “Stage 1” of decentralized process
  - Keep “Stage 2”: Center makes netting proposals to sets of participants that include side payments
  - Continue to focus on system welfare
    - have to estimate benefits and costs (with learning) or require banks to report them (mechanism design problem)
    - do not take initial liquidity positions as given

Simulations

- 2 banks, A and B
- costs: $c_i \sim U[0.05,0.15]$, $i = A, B$
- benefits: $b_i \sim U[0.025,0.075]$, $i = A, B$
- payments: $p_i \sim U[1,100]$, $i = A, B$
- c_i and b_i, once drawn, are fixed throughout the simulation (1000 periods)
- payments p_i are drawn each period and are common knowledge

Scenario 1: Decentralized with side payments

- each bank knows its own cost and benefit
- assumes the mean values for the other bank: eg A assumes $c_B = 0.1$ and $b_B = 0.05$.
- banks take turns making proposals
- fair proposal
- payments settle only if recipient accepts

Eg. A proposes, $p_A > p_B$

$$v(A) = \max\{(b_A - c_A)p_A, 0\}$$

$$v(B) = \max\{(.05 - .1)p_B, 0\} = 0$$

$$v(A, B) = \max\{b_A p_A + .05 p_B - c_A(p_A - p_B), 0\}$$

Order	**Marg. Contrib. A**	**Marg. Contrib. B**
A,B	$v(A)$	$v(A,B)$-$v(A)$
B,A	$v(A,B)$	0
Shapley Value	$.5v(A,B) + .5v(A)$	$.5v(A,B) - .5v(A)$

Scenario 2: Centralized with side payments

- Center assumes the mean values; that is, $c_A = c_B = 0.1$ and $b_A = b_B = 0.05$.
- fair proposal
- payments settle only if both banks accept

Eg. $p_A > p_B$

$$v(A) = \max\{(.05 - .1)p_A, 0\} = 0$$

$$v(B) = \max\{(.05 - .1)p_B, 0\} = 0$$

$$v(A,B) = \max\{.05(p_A + p_B) . 1\ (p_A - p_B), 0\}$$

Order	**Marg. Contrib. A**	**Marg. Contrib. B**
A,B	0	$v(A,B)$
B,A	$v(A,B)$	0
Shapley Value	$.5v(A,B)$	$.5v(A,B)$

Benchmark Scenarios

- **Scenario 3: No side payments**
  - Bank in debit position provides needed liquidity without assistance

- **Scenario 4: Complete information with side payments**
  - Proposals reflect true costs and benefits (doesn't matter who makes the proposal)

Helpful Fact

Cost allocation defined using Shapley value will be accepted by all banks if v is superadditive.

- Super-additive v implies Shapley value allocation is an imputation.
- Scenario 4 is the best case scenario for welfare enhancing netting

Results

- Averages over 1000 simulations of 1000 periods

Mechanism	**Welfare**	**Rank**
Decentralized	3.239	2
Centralized	3.190	3
No side payments	3.099	4
Complete info with side payments	3.315	1

Paired two sample for means tests significant at 1% level.

Results

- Averages over 1000 simulations of 1000 periods

Mechanism	Welfare	Accept	Reject	No proposal
Decentralized	3.239	62.4%	6.0%	31.5%
Centralized	3.190	55.6%	11.4%	33.0%
No side payments	3.099	57.6%	13.7%	28.7%
Complete Info with side payments	3.315	71.3%	0.0%	28.7%

Paired two sample for means tests significant at 1% level.

Results

- Averages over 1000 simulations of 1000 periods

More rejections when no private information is used

Mechanism	**Welfare**	**Accept**	**Reject**	**No proposal**
Decentralized	3.239	62.4%	6.0%	31.5%
Centralized	3.190	55.6%	11.4%	33.0%
No Side Payments	3.099	57.6%	13.7%	28.7%
Complete Info with Side Payments	3.315	71.3%	0.0%	28.7%

Paired two sample for means tests significant at 1% level.

Results

- Averages over 1000 simulations of 1000 periods

Mechanism	Welfare	Accept	Reject	No proposal
Decentralized	3.239	62.4%	6.0%	31.5%
Centralized	3.190	55.6%	11.4%	33.0%
No Side Payments	3.099	57.6%	13.7%	28.7%
Complete Info with Side Payments	3.315	71.3%	0.0%	28.7%

Cases where total surplus is positive but liquidity provider requires side payment

Paired two sample for means tests significant at 1% level.

Welfare comparison under different degrees of uncertainty

- Averages over 1000 simulations of 1000 periods

	$c_i \sim U[0.05,0.15]$ $b_i \sim U[0.025,0.075]$		$c_i \sim U[0,0.2]$ $b_i \sim U[0,0.1]$	
Mechanism	**Welfare**	**Rank**	**Welfare**	**Rank**
Decentralized	3.239	2	3.110	3
Centralized	3.190	3	2.869	4
No Side Payments	3.099	4	3.148	2
Complete Info with Side Payments	3.315	1	3.400	1

Summary

- Decentralized LSMs have potential to be welfare maximizing
- Centralized version just as good if no private information
  - perhaps better for a variety of reasons
- Several things to (re)consider
  - Best guess under incomplete information
  - Initial liquidity provision
- Several things to explore further
  - Machine learning
  - Mechanism design approach
- Welcome comments and suggestions!

Thank You