Scaling and Consenus in Monetary Systems.

A Preprint

Jacky Mallett[*]
Assistant Professor,
Department of Computer Science
Reykjavik University
Iceland
jacky@ru.is

February 10, 2020

Abstract

Limits on the information capacity of communicating networks present significant scaling challenges when attempting to architect stable and reliable monetary systems, especially since these systems also have to resolve Fisher consensus issues as money is moved around the financial system. Theoretical and unproven assumptions from economic theory on the advantages of monetary stability, replacing the existing banking system, and the desirability or otherwise of a single global monolithic currency must be carefully examined against these constraints in order to understand both their limitations and the consequent impact on economic activity. In this paper we will review some key results on scaling from computer networks, and discuss the scaling properties of the existing financial infrastructure, with those of some of the newer cryptocurrency alternatives.

1 Introduction

Distributed computer systems, applications which rely on co-ordinating the activities of large numbers of communicating nodes to cooperate on a shared purpose are increasingly commonplace. Examples range from highly centralised online database systems, to more distributed peer-to-peer applications, such as Skype, torrent file sharing, or the blockledger cryptocurrencies originating with Bitcoin. Architecting these systems to scale to the numbers required to operate at national or international levels is extremely challenging, as they must operate not only with respect to whatever their application requirements are, but also against network limits of available bandwidth and processing power for message communication. This is a particularly problematic when hard consensus requirements are imposed by the application, as is the case with monetary applications where uncertainty about the precise value of monetary holdings is rarely well receieved.

Scaling large scale distributed applications is often a complex operation, since the root cause of scaling issues may not be apparent. The symptoms of scaling issues, typically either network congestion and/or CPU overload can have a great many causes, only some of which are necessarily attributable to scaling. It is common for problems to be misattributed. Bitcoin's scalability problem for example, is currently attributed to constraints on "the maximum block size and the inter-block time"[1]. As we will show, the actual problem rests a little deeper in the information exchange needed to compute the proof of work, and maintain the blockledger, in conjunction with the reliance on a peer-to-peer architecture between the nodes. Changes to the block size and inter-block time may provide short term improvements, but in the limit, Bitcoin cannot scale with its current architecture.

Adding to the scaling problem is the peculiar difficulty of providing consensus in distributed systems; that is guaranteeing agreement on the same value as it is being modified between two or more nodes, for example the transfer of money between accounts. The root of this difficulty is the Fisher consensus problem[2]. This simply states that it is impossible to *guarantee* that any number of asynchronously connected nodes will agree on even a single bit value. There is no

[*] Views expressed in this paper should not be taken to represent those Supervisory Board of Directors of the Central Bank of Iceland of which the author is a member.

protocol that asynchronously connected nodes communicating by message passing can implement that will provide an absolute guarantee that at any instant the value held at both nodes will be the same. This is not to say that that agreement will not occur, usually it will, rather that no protocol exists that can guarantee it.

There are in essence two approaches for "solving" the consensus problem in distributed applications. Either the application must relax its requirement for consensus, or it must find a way to synchronise its nodes. Synchronisation, since it typically requires centralising all activity around the value in question on a single node, brings with it equally intractable scaling limits. The break through Nakamoto et al. made with Bitcoin was to develop an elegant distributed algorithm, the proof of work, which allows nodes to perform this synchronisation by engaging in a distributed race for temporary ownership of the synchronisation point. This in turn allowed Bitcoin to use a peer-to-peer relationship between the nodes in its network, which allowed it to signifcantly exceed the limits imposed on a more centralised system.

Scaling limits in all distributed systems can be traced to this combination of the communication limits on bandwidth between their individual nodes, which vary with both the application's message processing requirements, and the technology supporting them, and the arrangement of nodes (their topology) within the application that they are operating. This latter is subject to the needs of the application for consensus on particular values, and any required response times. This interaction between the requirements of the application performing as a network to support its own activities, and the activities themselves presents a complex set of challenges when designing these systems to scale, as the presence of an issue as fundamental and as common as the Fisher consensus problem may well suggest.

That there are limits on communication within network based systems, can be traced back to the work of Shannon[3], who showed that there were hard limits on the communication of information between two nodes based on available bandwidth and error rate. Extending this to large scale networked systems would require another 50 years of computer network development, and a complete understanding would not emerge until the early 21st century, as large scale distributed software applications, such as Skype and other peer-to-peer programs were developed to facilitate information exchange, and the realities facing large scale systems that also had to function as a network to support their operations became apparent.

The "scaling limit" of a system is generally regarded as the maximum number of users or nodes that it can support. How it supports them, is typically a set of tradeoffs that have to be made within that limit on the level of service that can be supported to each node that can manifest themelves in different ways. The Bitcoin network for example, is limited by the number of transactions per second it can perform, and has reached the size where due to the limits on network connectivity which we will discuss in this paper, it cannot add more nodes to help address this.

When scaling limits that are due to network limits within the application are reached, there are relatively few options. Incremental improvements can usually be achieved by improving performance. Beyond that the solution is either to completely re-architect the system (if possible), or to shard it into copies of itself. Sharding is a fairly widely used method of scaling distributed systems, where copies of the system are made, and users distributed between the copies. It works, as long as there are mechanisms to allow communication between the shards as needed, and this communication is relatively limited. An example would be the proliferation of cryptocurrencies, where each one effectively provides an additional sharded blockledger, with market price setting being used to exchange cryptocurrencies. Unfortunately, although each cryptocurrency is algorithmically designed to converge to a fixed number of coins, the ability to arbitrarily create alternate cryptocurrencies, even if it solves the scaling problem, has resulted in a proliferation of cryptocurrencies unmatched since the ill fated American experiment in the 1840's with unrestricted fractional reserve banking[4]. This may not be exactly what Bitcoin's originator intended, with respect to quantitative monetary stability.

To provide a contrast with the cryptocurrency approach to providing monetary transmission services, we will also discuss the scaling properties of the existing banking system. From a distributed systems perspective, fractional reserve banking provides an interesting example of an emergent medieval networked system, solving many of the same problems with respect to consensus and networked based transfer that cryptocurrency is currently attempting to solve. If we concentrate on its role as a monetary transmission network, with information being communicated (in its Shannon sense) by economically active agents performing monetary transfers, then we can analyse it with exactly the same tools. In this role the banking system performs exactly the same role as cryptocurrencies, facilitating the exchange of values in ledger based deposits, and has had to solve exactly the same problems of scaling, and resolving consensus issues to guarantee the integrity of the values recorded in those ledgers. Exactly how it does this is of both technical and economic interest.

Although the existing banking system is often portrayed by economists as a monolithic and highly centralised system, this is not in fact the case. Fractional reserve banking is highly distributed, and has excellent scaling properties. Considering the period of time it was developed over, effectively as a manually operated network, it is a remarkable construct. It provides robust solutions for consensus, and uses an interesting loan based netting mechanism to provide international transfers between countries and currencies. From a stability perspective though it commits the unfortunate sin of being a critical system that relies on recursive regulatory mechanisms, and on examination there are considerable architectural and economic issues caused by the system's extremely poor levels of fault tolerance with respect to loan defaults, which are the proximate cause of most credit crises. Well run banks are typically only able to absorb losses of around 0.75% of their total loan capital per year, and when losses go above this limit, a series of interactions with the regulatory framework will typically trigger the cascade failure that is otherwise known as a credit crisis, recession or depression, depending on severity and time period.

Regulating the system's monetary expansion has also presented considerable challenges over time, and failure to do so has caused a number of economically and socially catastrophic hyperinflations. This was also the proximate reason for the development of Bitcoin provided in the Nakamoto paper[5].

In this paper we will review the origins of the scaling limits, and describe how they apply to the different organisational topologies used by distributed systems. We will compare and contrast the peer-to-peer network organisation of the current cryptocurrencies, with the organised and sharded peer-to-peer organisation of the banking system, and discuss the advantages and disadvantages accompanying them. We will also discuss the scaling solutions that have so far been applied to Bitcoin and some of the issues with them.

2 Scaling Limits in Networked Systems

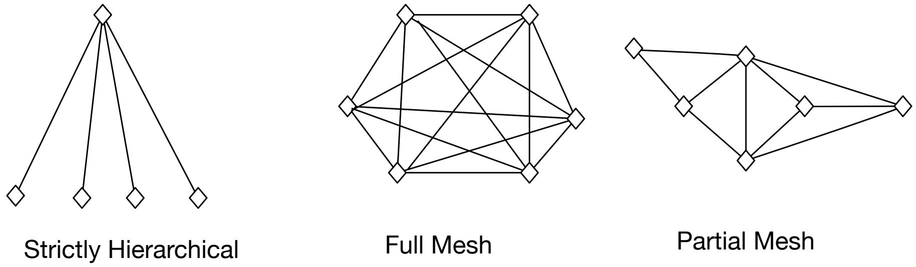

Figure 1: Simple Network Topologies

Concrete bounds on the total amount of information that can be transmitted within a large scale network of nodes, were not worked out until the end of the 20th century. While it is straightforward to estimate the limits within a hierarchical system, typically a simple function of the capacity of a central server, mesh networks are more complex. Gupta and Kamal[6][2] showed in 2000 that the limit on the instantaneous transmission of information within an unorganised peer-to-peer(p2p) network could be expressed as $L\sqrt{N}$, where L is the link capacity of each node, expressed as the number of messages it can send or process, and N is the total number of nodes in the network. This assumes an homogenous network where all nodes have equal link capacity, which was the original basis for the development of the peer-to-peer networks. In reality, there can be a large range of performance in nodes within an application, and hierarchical systems in particular, are usually engineered to have extremely powerful servers with simple end devices.

This followed from Claude Shannon's[3] work, which showed that there are hard limits on the maximum amount of information that can be transmitted through a given channel between two nodes, and also introduced the concept of Information, a unique message transmitted between two or more nodes. The concept of message uniqueness is important in order to distinguish between a broadcast message, where the same message is sent to many nodes, and when the same set of nodes exchange messages with different content. [3] This can be critical in applications where a broadcast message is used to perform some form of synchronisation, and create a shared view of data amongst its nodes, such as as the broadcast of transactions in the Bitcoin network. The resulting synchonisation comes at the price of a significant reduction in the information content being shared, and conseqently affects scalability. The result is that the theoretical maximum information capacity of a distributed system when calculated on the basis of its message throughput rate, can often be significantly greater than its actual information capacity.

This limit on p2p network connectivity becomes apparent as the network scales, and nodes within the network have to act as relay points for messages between nodes that are not directly connected. As more nodes are added to the unorganised p2p network, the number of messages that have to be relayed increases the load on intermediary nodes, and

[2] A considerably simplified and elegant proof was subsequently published by Scaglione[7].

[3] Certain technologies, such as radio consequently have an inherently low information content as they broadcast large numbers of identical messages.

eventually a point is reached where the additional load induced is greater than the additional message relaying capacity introduced by new nodes.

If we add to this the limits for a single master-server, hierarchical network (L), and a full mesh network where all nodes are directly connected to each other, $L(L-1)$, we have the scaling limits for the 3 simple topologies shown in Figure 1. Although unorganised peer-to-peer networks are able to scale significantly better, in the limit, they are asymptopically bound, and reach a point where the addition of additional nodes to the network does not bring with it a matching increase in the overall network's information capacity.

Scaling in any topology can of course be improved if additional resources are allocated in order to improve link capacity between nodes, increasing the value of L. Depending on the application's requirements, this can be larger computers to provide increased messge processing power and/or higher bandwidth. During the period when Moore's law governed, and processing power was doubling every 18 months, this was often enough for medium sized systems. Similarly to rewriting the software to improve its efficiency this only postpones the inevitable in large systems. Unless very careful decisions with respect to architecture and function are made, scaling problems are often intractable for large systems with significant needs for internal communication between their nodes.

2.1 Arbitrary Scaling

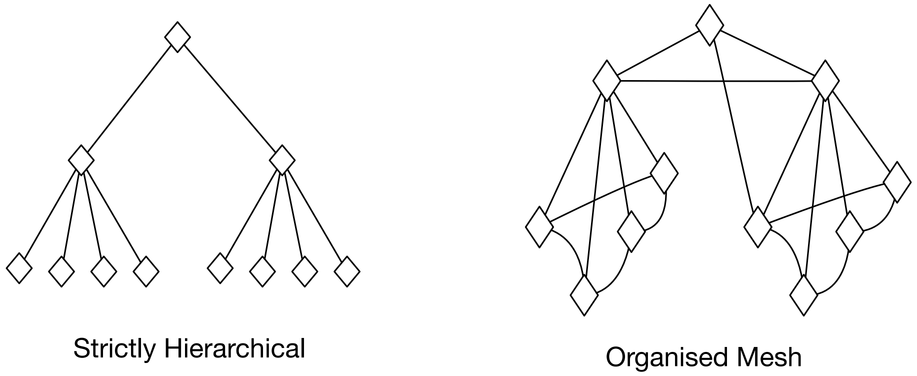

Figure 2: Scalable Topologies

The holy grail of distributed systems is the development algorithms that allow systems to scale arbitrarily with the number of users, and to scale to an arbitrary number of nodes. This is the declared goal of Bitcoin and other cryptocurrencies, but to achieve this at national or planetary levels, is extremely demanding.

Figure 3 shows the scaling limits of an unorganised peer-to-peer network as a function of different link capacity levels (L), and the number of nodes, against the amount of communication that could be performed if all nodes in the system were able to send messages to each other at their full rate $O(N(N-1))$. On the left hand side of the curve we can see that the limits do not apply to small systems which are operating in a communication regime where there is more communication capacity available than they can saturate. This is quite often the case under laboratory conditions, and scaling problems are often only revealed after the system is deployed.

There are effectively three regimes that peer to peer communicating systems operate in: unrestricted, restricted but able to grow, and unable to grow as adding additional nodes imposes a greater burden on the system's communication capacity than is contributed.

One way to model the operation of these limits, is the concept of group size: the maximum number of nodes any given node can connect to directly (1-hop), and by extension the nodes that can be communicated with without additional communication overhead[8]. If the requirements of the application are such that its nodes can be organised into groups

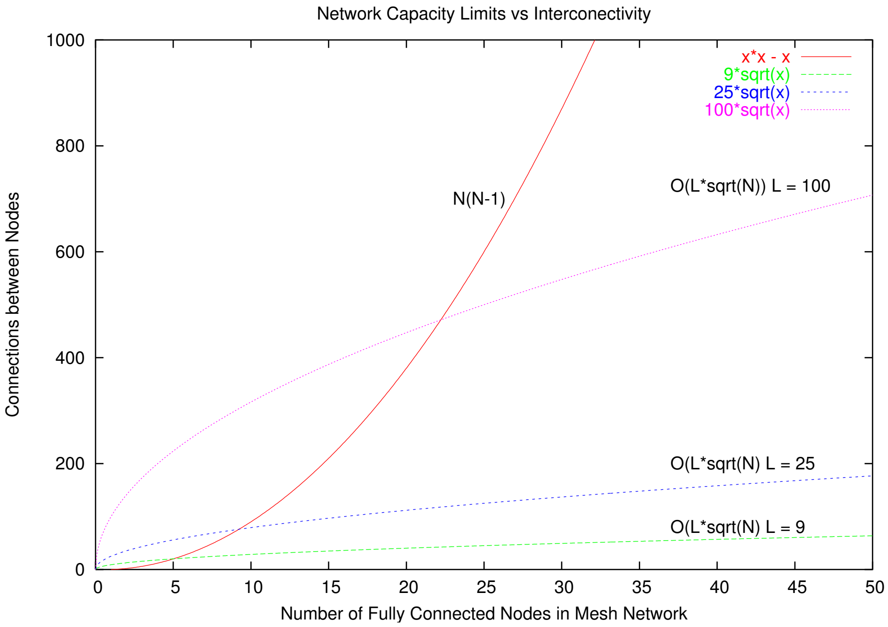

Figure 3: Scaling Limits for p2p topology

that signficantly localise communication within their group, then it is possible to have peer-to-peer networks that scale arbitrarily with N, with each group contributing additional information capacity to the larger system. This offers an explanation for the structure that has been seen to emerge in the Internet[9], despite the original design goal of a fully distributed system, and in many other large scaled networked systems, including emergent systems such as the existing banking system.

By extension of the same argument, the rigidily hierarchical network can also be said to scale arbitrarily, where levels are added duplicating the master-client topology. But it scales within a markedly reduced information capacity,[4] in comparison to an organised p2p network. It also has identifiable single points of failure, so is typically less fault tolerant. We can consequently observe that there are two topologies that can in principle scale to arbitrarily large numbers of nodes and maintain network connectivity: organised peer to peer, and strictly hierarchical networks, with it should be said a large number of hybrids. A large part of the design problem lies in determining which one is most appropriate for any given application's requirements, and the network conditions it is operating within. This typically involves a delicate set of tradeoffs. While organised peer-to-peer topologies have access to a far larger information capacity, and so have much better inherent scaling properties on this dimension, they have significant issues when consensus is required since they do not have the natural synchronisation points that exist in hierarchical networks. However these are also both single points of failure, and points of congestion, which the p2p networks are more robust against.

As a side note, communication latency - the time taken to transmit, receive, and process messages, also plays an important role in topology choices. The information advantages of a mesh network depend on there being time to perform wide scale exchanges of information, and with a high communication latency relative to decision time, this is not always available, and hierarchical topologies must be used.

Whilst the scaling limits around topology are relatively straightforward, the consequent design choices for any distributed system, and this extends to economic and financial systems, which also rely on communication between nodes co-operating on shared tasks, are anything but. Most large scale distributed system design is effectively a delicate dance around scaling limits, communication latency and the Fisher consensus problem.

2.2 Application Design and Consensus

A central, and fundamental conflict occurs in any distributed system that requires both good scaling properties, and consensus between nodes on particular values. The Fisher consensus problem states that it is impossible to guarantee

[4] Intuitively this follows from observing the number of wasted links with the clients at the bottom of the hierarchical network which are utilised by p2p networks2.

consensus between any group of asynchronously connected nodes on even a single bit value. The problem arises from the issue that there is no provably no protocol that can reliably determine between a lost and a delayed message. Although this may seen a relatively simple observation, Fisher consensus is the origin of the issues around the CAP theorem[10], and the fundamental issues with scaling traditional databases.

There are two "solutions" to the unsolvable consensus problem available to architects of these systems: find a way to synchronise the nodes that require consensus, and deal with the resulting scaling issues, or relax the requirement for consensus. The latter is obviously not available to financial systems that are required to be able to guarantee consensus on ownership of financial instruments, or the value held in accounts. Consequently they must turn to synchronisation, and both Bitcoin and the fractional reserve banking system have found elegant ways to address this problem in their respective domains.

2.3 Scaling strategies

When scaling problems occur in a distributed application, there are essentially three strategies availble to try and resolve them: increase the value of L(node connectivity in its various forms), decrease the value of N either by making nodes perform more efficiently, upgrading their hardware, or attempt to rearchitect the topology of the application to improve scalability. The first two of these strategies effectively attempt to increase the upper bound on communication, and while this will allow the system to continue to grow, inevitably if growth continues the system will simply hit a new scaling limit within its topological constraints. This is not to discount these approaches. The combination of Moore's law on the computation side, and the introduction of fiber-optic communication on network communication in the last 20 years, has resulted in massive increases in L, and an accompanying increase in the group size of many distributed computer systems.

Rearchitecting the system to use a different, and more scalable topology can also be attempted, but typically topology choices are so deeply embedded in the system that this is usually impractical, and new systems have to be developed instead. With more advanced distributed systems, such as the p2p networks, the best approaches to this are also an open research problem. Designing a system to be able to take advantage of an organised p2p topology is non-trivial, especially in the presence of consensus requirements, and there has been a clear preference for fundamentally hierarchical systems as a result, simply because they are easier to build. Paradoxically, networked systms that rely on some form of organic growth, such as the Internet, are more likely to find some form of organised p2p topology, as a result of the preferential attachment mechanisms described by Barabasi and Bonabeau[11], in their work on scale-free networks.

3 Scaling Monetary Systems

3.1 Cryptocurrencies

The original cryptocurrency, Bitcoin, is based on a pure p2p topology, putting it in the first class of scalable topologies, with a scaling limit of $L\sqrt{N}$. Within the application transactions are broadcast across the p2p network using a flooding protocol, which creates an exponential load on the system as transactions increase. Computation is distributed across miner nodes in the Bitcoin network, which compete with each other calculating hashes (the proof of work) in order to to find a block with a predefined prefix[5]. The hash is derived from an arbitrary number of Bitcoin transactions, the previous block's hash, and a nonce, hence the need for broadcast. Transactions must have a timestamp within a certain range to be accepted into a block, but this range is quite wide, and can adjust between blocks to reflect the rate of transactions being processed. There is however, no requirement for synchronisation between nodes on the list of transactions being processed, or on their order, and it is possible for a block to contain no transactions at all[12]. Miners participate in an independent race to calculate a hash based on their inputs, with a predefined prefix (a sequence of 0's), and the first miner to find this becomes the synchronisation point for providing that block to be committed to the blockchain, and is rewarded for their discovery. This provides an elegant mechanism that is used to resolve the fisher consensus issue for the Bitcoin blockchain: the criteria for agreement are synchronised across the network by their incorporation in the mining algorithm, and the first miner to create a block and hash that meets them is then self-selected as the synchronisation point to broadcast the new block to be committed to all other nodes, without need for further communication.[6]

In order to create a reasonable throughput in the system, Bitcoin blocks were limited to be no bigger than 1MB. There is a tradeoff operating with the number of transactions in a block, and the time to commit and propagate the block across the network, since if the block exceeds 4MB a sizable number of nodes in the network will not receive the committed

[5] A slowly increasing number of 0's

[6] Trying to solve consensus problems using more communication, typically tends to just create more consensus problems.

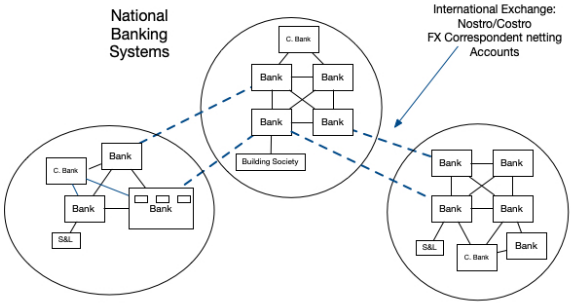

Figure 4: Network Topology of the International Banking System

block before the next block has been mined[13]. Blocks are mined on average every 10 minutes, so the propagation time across the network must be significantly less, otherwise the groups of nodes closest to the block solving the proof of work will operate at an advantage[14]. Attempts to increase this limit, in order to increase the transactions per second for the Bitcoin network, have been controversial. They are also destined to fail. As the block size increases, even on high speed network, propagation and processing time increases proportionately. Some success has been seen in reducing transaction size, BIP 91, Segregated Witness has been activated since 2017, and by making individual transactions smaller improves block performance.

Other attempts to improve the connectivity performance (L), include introducing a separate high speed network and accompanying protocol between miners, the Fast Internet Bitcoin Relay Engine (FIBRE). This replaces TCP between mining nodes, with a custom UDP protocol, which builds in an error correction/detection protocol[7]. FIBRE is mainly used between nodes in mining pools, which can be selective if they need to be. Replacing TCP with UDP is a fairly common approach to poor network performance in networked applications, and can offer some improvement if done correctly, but at the risk of making the application more susceptible to congestion collapse, and less able to handle heterogenous network conditions.

More interestingly the Bitcoin Lightning Network attempts to treat Bitcoin as a clearing mechanism, and only use the blockchain for final settlement[15]. Transactions are made directly between two parties, and can be forwarded, relying on cryptographic signing between partners, and eventual settlement on the blockchain. Scaling is still limited by the reliance on the blockchain, and by the number of channels that users can hold open with other users, but this approach does provide potential mechanisms where a more organised group topology could be adopted, to resolve some of these issues.

A significant amount of localised grouping of nodes has also occurred from the emergence of mining pools, where computers co-operate on the proof of work problem, and share any rewards. Typically the pools use specialised software to co-ordinate their nodes, and this has the side effect of providing some limited sharding within the network. In the limit though, the only way to scale Bitcoin significantly would be a major rearchitecture.

3.2 Fractional Reserve Banking

As a comparison it is interesting to examine the scaling properties of the existing worldwide monetary transfer network, a medieval technology, known as fractional reserve banking. Modern banking systems emerged from a sequence of book keeping and accounting developments in the medieval european period. Initially it proved highly unstable, with bank failure being commonplace. Enough was learned about its behaviour in Britain during the early 19th century to provide regulatory controls that brought its behaviour under some control, the so called British Monetary Orthodoxy, otherwise known as the first gold standard[16]. Fractional reserve banking spread to Asia in the late 19th century, being somewhat forcibily introduced to China in the 1850's[17], and adopted by the Japanese Empire after the Meiji Restoration[18], in the 1870's. Today it provides an extremely reliable world wide monetary transfer network, but whilst its systemic stability has improved somewhat, it still causes periodic, macroeconomically destabilising credit crises, due to its poor fault tolerance characteristics.

[7] UDP is an unreliable protocol, and offers no guarantees that packets will be delivered. On high speed fibre optic networks, packet losses are extremely low, and UDP can provide an optimisation. However should any form of network congestion occur, UDP packets will be dropped.

Contrary to claims made by many cryptocurrency advocates however, banking is not a highly centralised system. Although "central" banks nominally exert control over the system in economic theory, the title is something of a misnomer. In terms of their actual position in the system, the central bank is just another bank. They may attempt to exercise soft authority over the system in some countries, but as events in 2008 demonstrated only too well, their primary role in modern times has been to intervene, using the power of the state if necessary to create and inject asset cash, to ameliorate credit crises. They are unable to prevent them.

Their primary systemic role, is to act as the lender of last resort to banks who have insufficient asset cash liquidity to satisfy requests to transfer deposits to other banks. A bank can quite legitimately be illiquid simply due to short term fluctuations in the movement of asset money around the system. There is no way to guarantee that inflows and outflows of cash will be equal on any given day. A well run bank will monitor its asset cash liquidity closely to prevent this issue, but over the longer term banks can be at the mercy of asymmetric flows of money triggered by lending if their deposit holders have loans issued by other banks. In a competitive system, the presence of the central bank as the lender of last resort is necessary simply to prevent banks withholding overnight loans of asset cash to their own competitive advantage. Most of the power they have comes more through influence, and institutionalised behaviour, than direct control within the system.

To the general confusion of economic theorists, the banking system effectively operates with two types of money, asset or physical cash[8], and ledger held deposits. While ledger deposits can be created from physical cash deposits, the majority of deposits are initially created when loans are issued, which is where the confusion originates. In operation, the bank's holdings of asset cash are effectively statistically multiplexed against their deposits, and primarily used as a semaphore or token to perform transfers of deposits between banks as shown in figure 3.2. Since it is possible to transfer money between customers at the same bank, without any involvement of cash, it is incorrect to not classify deposit money as a form of money as some still do. In fact by 1887[19], deposit tranfers constituted 90% of all monetary transfers in the USA. In the modern era, deposits have more or less taken over from cash as the predominant form of money in the system in many countries.

Fractional reserve banks operate with local ledgers, created using double entry book keeping, which hold customer account information. These are held independently at each bank, and if applicable their branches, with each bank or branch maintaining ledgers recording the exchanges between their own customers, and also between those customers and those at other banks or branches. Banks and their branches historically operated their ledgers as independent shards, but with computerisation some banks, particularly in the UK, have consolidated their activities into a single ledger.

As shown in figure 3.2, banks use peer-to-peer networking to transfer money between banks, and also internationally using the costro/nostro network which the Society for Worldwide Interbank Financial Telecommunication (SWIFT) provides the network facilities for. Banking can be classified as an example of an organised peer-to-peer network, which was especially the case historically, when banks operated within very distinct geographical areas. Technology has markedly increased processing capacity, in the case of the banks, with the consequence that individual banks now scale to considerably larger sizes. Overall though, the architecture follows the organised peer-to-peer or group based topology, allowing it to scale more or less arbitrarily.

As a monetary transfer system, this is an elegant approach. The use of asset cash as a semaphore to transfer money between banks effectively resolves the consensus problem that would otherwise arise. The accompanying double entry book keeping operations deduct cash and the matching deposit, a physical or electronic transfer of cash between banks occurs, and the money is then credited to the receiving deposit account.[9]

The problematic relationship between the banking system and the economies that use it, lies primarily in its fault tolerance and regulation, and in its control structures. Lending has to be carefully controlled, otherwise the associated monetary creation can trigger hyperinflation. Neither liability money nor loans are created evenly across the banking system, and the resultant long term price imbalances can become self-sustaining. It is more accurate to think of a monetary field than a money supply in this context. The relationship of deposit creation with lending is also two sided. Liability deposit money is not just created when loans are made, it is also destroyed when they are repaid, and may also be destroyed if loans have to be written off by the bank. The arithmetic on loss provisions is fairly brutal, most well run banks can afford to write off losses of at most 1% of their total loan book in a year, and any wide spread losses above that can be economically catastrophic, as they trigger a contraction in the money supply.

There is also the danger, that loan repayment will exceed new lending, which also has the potential to trigger monetary deflation. The entire system relies on new lending always being larger than loan repayment, otherwise the money supply

[8] Increasingly held electronically

[9] Strictly speaking, the operation at the originating bank is [credit cash, debit deposit] and [debit cash, credit deposit] at the receiving bank.

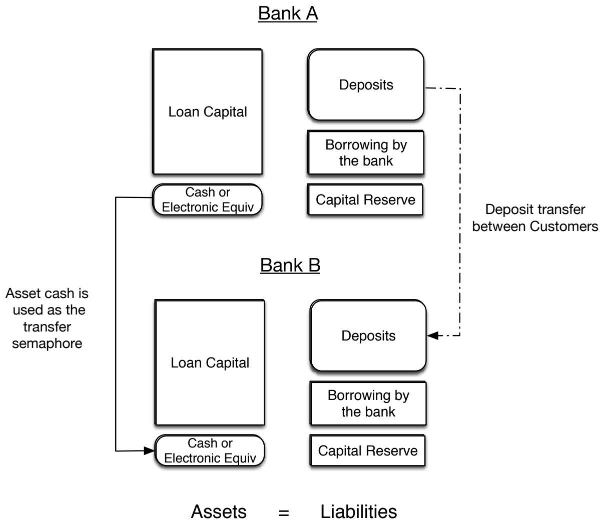

Figure 5: Deposit transfers between Banks using Asset cash as a semaphore

will shrink. Although market pricing can adjust for this, it causes signficant issues with forms of financial instrument such as bank debt, which have no way to alter their repayment structures.

As a peer-to-peer network, control is also an issue. Banks are highly regulated, because there is no true central point of control. Banks make independent lending decisions within their regulatory limits, and regulatory failures, especially profitable regulatory failures such as loan securitzation can be extremely hard to deal with. This is the typical dilemna between hierarchical and peer-to-peer systems, where there is a direct tradeoff between scaling and control.

4 Conclusion

Since they are attempting to solve the same problem, it should not be surprising that as cryptocurrency evolves, it is often rediscovering identical financial mechanisms to those traditionally used by banks. Due to widespread misinformation on the operations of banking this isn't always appreciated by those concerned. As of writing, a number of companies are advertising Bitcoin loans, or the ability to borrow other currencies using Bitcoin as collateral. Cryptocurrency exchanges have emerged, which take deposits of Bitcoin, and provide what are effectively deposit accounts which can be used to transfer Bitcoins between customers and provide trading facilities. That many of them have failed due to illiquidity, suggests that some may have been de facto operating on a fractional reserve basis. This parallels the evolution of the existing banking system from the older goldsmith bankers in the middle ages[20].

Several groups currently appear to be trying to adopt a sharded approach, similar to banking, with Bitcoin being moved into some form of synchronisation point or semaphore. Features of this can be seen in the lightning network evolution,[15] which is attempting to create a separate network to handle recurring transactions between known participants. This is in many ways identical to the netting methods that were traditionally used between banks to settle interbank transfers[21], before the introduction of real-time gross settlement systems.

Whether or not cryptocurrency can adopt some of the methods used by banking to meet its scaling ambitions is an interesting question. Scaling Bitcoin itself is problematic, it is locked into a peer-to-peer architecture, using broadcast as a synchronisation mechanism, so while its scaling can undoubtably be improved, it cannot scale to arbitrary numbers of nodes as is. There is also a non-technical barrier. Moving architecturally to a new topology would bring with it the need to back down a little from the original goal of a global, decentralised, peer-to-peer currency, to an organised mesh network, with some form of sharding. If this could be achieved though, the resulting cryptocurrency system could then potentially challenge the banking system as an alternative to existing monetary transfer mechanisms.

Scalable though it is, fractional reserve banking's close coupling of lending with money creation and destruction, has historically been the root cause of many major financial crises with devastating economic impact. For no other reason, decoupling money creation from bank lending is an attractive idea, one that has also been proposed by a number of economists in the form of full reserve banking. While Bitcoin itself may not be able to scale to support the numbers of transactions required for this, it may well lie on the path to development of something that will.

This is unlikely to be an economically neutral endeavour. Computer scientists should be well aware of the dangers of making changes on the live production system, and that is essentially what is being attempted here. In an ideal world, such changes would be accompanied by careful testing on simulated systems, at scale, as is routinely done for computer network software. At present the frameworks for doing this do not exist either in economics or cryptocurrency development, and it seems more likely that cryptocurrency experiments will continue to evolve alongside the existing system, and probably at some point interact with it.

Banking itself took several centuries to emerge from the development of double entry book keeping in 11th century Egypt. Initially it did not even maintain balanced books, but it evolved through the development of fiat forms of cash, and interbank transfers using checks and letters of credit, to the world spanning network it is today. The pace of cryptocurrency development over the last ten years has been considerably faster.

References

[1] Kyle Croman, Christian Decker, Ittay Eyal, Adem Efe Gencer, Ari Juels, Ahmed Kosba, Andrew Miller, Prateek Saxena, Elaine Shi, Emin Gün Sirer, Dawn Song, and Roger Wattenhofer. On scaling decentralized blockchains. In Jeremy Clark, Sarah Meiklejohn, Peter Y.A. Ryan, Dan Wallach, Michael Brenner, and Kurt Rohloff, editors, *Financial Cryptography and Data Security*, pages 106–125, Berlin, Heidelberg, 2016. Springer Berlin Heidelberg.

[2] Irving Fisher. A program for monetary reform. 1939.

[3] Claude E. Shannon. Communication in the Presence of Noise. *Proceedings of the Institute of Radio Engineers*, 37:10–21, 1949.

[4] W. H. Sykes. Free Trade in Banking. *Journal of the Statistical Society of London*, 30(1):58–67, March 1867.

[5] Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. 2008.

[6] P. Gupta and P. R. Kumar. The capacity of wireless networks. *IEEE Transactions in Information Theory*, 46(2):388–404, 2000.

[7] Anna Scaglione and Sergio Servetto. On the interdependence of routing and data compression in multi-hop sensor networks. *ACM/Kluwer Journal on Mobile Networks and Applications (MONET)*, 2002.

[8] Jacky Mallett. *The Role of Groups in Smart Camera Networks*. PhD thesis, Massachusetts Institute of Technology, 2006.

[9] Michalis Faloutsos, Petros Faloutsos, and Christos Faloutsos. On power-law relationships of the internet topology. *SIGCOMM Comput. Commun. Rev.*, 29(4):251–262, August 1999.

[10] S. Gilbert and N. Lynch. Perspectives on the cap theorem. *Computer*, 45(2):30–36, Feb 2012.

[11] Albert-László Barabási and Eric Bonabeau. Scale-free networks. *Scientific American*, 288(5):60–69, 2003.

[12] Qin Yang Canhui Wang, Xiaowen Chu. Measurement and analysis of the bitcoin networks: A view from mining pools. *Published in arxiv*, 2019.

[13] J. Göbel and A. E. Krzesinski. Increased block size and bitcoin blockchain dynamics. In *2017 27th International Telecommunication Networks and Applications Conference (ITNAC)*, pages 1–6, Nov 2017.

[14] K. Croman, C. Decker, I. Eyal, A. E. Gencer, A. Juels, A. Kosba, A. Miller, P. Saxena, E. Shi, E. G. Sirer, D. Song, and R. Wattenhofer. On scaling decentralized blockchains. *3rd Workshop on Bitcoin Research, Barbados:BITCOIN*, 2016.

[15] Joseph Poon and Thaddeus Dryja. The bitcoin lightning network: Scalable off-chain instant payments. 2016.

[16] Frank Whiston Fetter. *Development of British Monetary Orthodoxy, 1797-1875*. Harvard University Press, 1965.

[17] Chen Zhengping. *A Brief History of Finance in China*. Paths International Ltd., 2014.

[18] Alexander Allan Shand. *Ginko-Boki-Seiho (Book keeping System of Banks)*. Tokyo: Ministry of Finance, 1874.

[19] C. F. Dunbar. Deposits as Currency. *The Quarterly Journal of Economics*, 1(4):401–419, July 1887.

[20] Stephan Francis Quinn. Banking before the Bank: London's unregulated Goldsmith-Bankers, 1660-1694. 1994.

[21] Martin Campbell-Kelly. Victorian data processing. *Communications of the ACM*, 53(10):19–21, 2010.