

BCXP: Blockchain-Centric Network Layer for Efficient Transaction and Block Exchange over Named Data Networking

George Sedky

Faculty of Media Engineering and Technology
German University in Cairo
Egypt
Email: george.sedky@student.guc.edu.eg

Amr El Mougy

Faculty of Media Engineering and Technology
German University in Cairo
Egypt
Email: amr.elmougy@guc.edu.eg

Abstract—Blockchains enable mistrusting entities to agree on a common state, and preserve integrity without any need for a trusted 3rd party. However, blockchains suffer from scalability issues, where transaction throughput is affected by multiple factors such as inter-block time, number of peers, and blockchain forks. Accordingly, this paper proposes a novel decentralized peer-to-peer network layer that addresses these issues, and more. In our proposed Blockchain-Centric Exchange Protocol, an information-centric architecture is used to optimize information exchange. We demonstrate the ability of our design to meet many of today's application requirements such as permissioned blockchains, isolation of transaction, validation, and mining traffic, and eliminate the need for an overlay network. Using computer simulations we implement BCXP, compare its performance with other blockchain-centric protocols and IP overlay networks. The results show that even by pushing such IP overlay protocols to impractical limits, BCXP still provides clear performance advantages.

I. INTRODUCTION

Through distributed consensus, trust-less blockchain systems evolved beyond cryptocurrencies. Due to the emergence of smart contracts, a plethora of applications like decentralized exchanges and crowd-sourcing platforms are now possible.

Nevertheless, blockchain systems might not yet be considered as practical replacement to their centralized trustful counterparts. Fundamentally, blockchains require peers in the network to reach consensus in order to maintain the system's integrity. This requirement is affected by various parameters such as the expected time between successive blocks (inter-block time) and the number of transactions per block. Achieving consensus is also affected by the underlying peer-to-peer network, block propagation delay, and network bandwidth. Widely adopted IP based overlay network layers establish a logical topology of peers while data is still transferred via the physical topology of the network[1]. This introduces higher delays due to traffic redundancy, and becomes increasingly more complex as the number of peers increase, rendering the blockchain's transfer layer a bottleneck hindering scalability.

Blockchain networks' scalability is an active problem in the open source community[2], as well as in research. Some previous work highlighted how latency, bandwidth, and state

forks limit blockchain scalability[3]. Further analysis conducted on Bitcoin's[4] peer-to-peer overlay network, modeled the relationship between block propagation delay and state forks[5]. Solutions were proposed to optimize IP overlay networks, first by increasing block sizes[6][5], which is bound by the network's bandwidth; increasing propagation delay, and giving smaller blocks an advantage of being committed by the majority of peers. Second by increasing the degree of peer connectivity[5], in turn increasing network overhead.

BlockNDN[1] is a proposed information-centric architecture using a synchronization protocol ChronoSync[7] over Named Data Networking (NDN)[8]. But ChronoSync is vulnerable to network partitions[7], leading peers to lose track of the longest chain if they went out of sync. BlockNDN was only concerned with the Bitcoin blockchain, and did not account for transaction propagation. It also required peers to flood the network with synchronization packets. Finally, BlockNDN was not evaluated empirically against IP overlay networks, therefore the advantages of blockchain-centric approaches were not highlighted.

This article introduce an alternative to IP overlay networks: building on top of an information-centric architecture[9] Named Data Networking (NDN)[8], designed primarily for blockchains, i.e. a blockchain-centric network. We implement our Blockchain-Centric Exchange Protocol (BCXP) on top of NDN to exploit many important features such as efficient information dissemination, different blockchain traffic isolation, supporting multiple blockchains concurrently, and enabling on the fly integrity checks. BCXP efficiently broadcasts blocks and transactions with 10x faster propagation, 14x less network traffic overhead, and 3x less fork rate compared to an IP overlay network (network of 90 peers), with booming performance gains as the network grows further.

In section II BCXP's architecture is discussed in detail. Afterwards, Section III presents the performance evaluation of BCXP against a model IP overlay network and BlockNDN. Finally, a summary of the article in section IV.

II. BLOCKCHAIN-CENTRIC EXCHANGE PROTOCOL DESIGN

A. Architecture Overview

BCXP Namespace (blockchain operations)
COPSS publish/subscribe (live mode transfer)
NDN interest/data (fetch mode transfer)

Fig. 1: Abstract Architecture

The architecture of BCXP can be divided into 3 layers (Figure 1). The first layer is Named Data Networking (NDN), where addressing and routing functions are implemented. NDN[8] provides the means for blockchain data to propagate in request/response fashion, where a data packet (response) corresponds to only one interest packet (request). When multiple interest packets arrive at a junction, routers avoid redundancy by collapsing them into a Pending Interest Table (PIT) entry, while forwarding only one interest to the producer. Eventually, when the producer responds, NDN routers consult their PITs and replicate the data packet to satisfy all outgoing interfaces on which an interest arrived, effectively implementing client multicasting natively.

Using NDN's interest/data model is sufficient for pull-based applications, where data transfer is initiated by the receiver. However, this is not the case in blockchain networks, where receivers are not aware of blocks/transactions generation time. This use case fits perfectly with a publish/subscribe model. Here, receivers would have to subscribe to namespaces they are interested in and, as soon as blocks/transactions are produced, they are broadcasted to all subscribers. NDN does not support publish/subscribe communication patterns[10] and hence requires an amendment. The Content Oriented Publish/Subscribe System (COPSS)[10] -the second layer in the BCXP architecture- implements this amendment. In this paper, we use an extended version of COPSS.

NDN leaves information addressing flexible to application designers. The third layer of BCXP is a framework for content addressing. We define a hierarchical namespace to include various blockchain operations, Divided into two complementary transfer modes; In *Live Mode* transactions and blocks are propagated as soon as they are generated, while in *Fetch Mode* old transactions and blocks are obtained from full nodes.

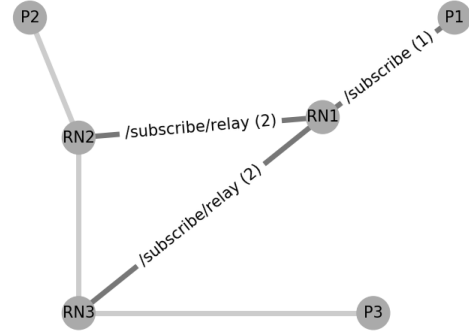
B. COPSS extension

COPSS extends NDN to add a publish/subscribe layer and introduces two new packet types: A subscribe packet and a publish packet. Subscribe packets are forwarded like interest packets using entries in the Forward Information Base (FIB). However, unlike interests, subscribe packets populate Subscriptions Tables (STs) instead of PITs. Internally, STs have similar implementations to PITs and aggregates subscriptions

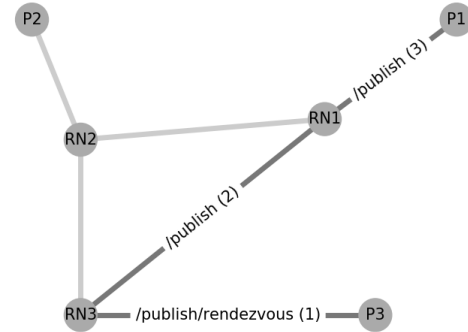
as well. On the other hand, publish packets are similar to data packets. They are sent by producers and are forwarded based on ST entries. Unlike the interests/data communication pattern, in COPSS a single subscribe packet corresponds to an arbitrary number of publish packets.

Also, COPSS requires the introduction of a Rendezvous Node (RN), acting as a root of the subscription tree and responsible for one or multiple content prefixes. Any COPSS-aware router can act as an RN and multiple RNs can be used at the same time. RNs are the only nodes in the system that would require prefix announcements for routing. Hence, the initial setup of COPSS does not require a lot of pre-configuration, but requires COPSS-aware routers, unlike BlockNDN where all content providers need to broadcast routes.

COPSS mandates that all RNs be reachable from all publishers, and authors did not specify how a publisher could exactly leverage multiple RNs in the network. This is very important in a peer-to-peer blockchain network, where almost all peers are considered publishers. Therefore, we extend COPSS RN model, by adding hierarchical subscriptions. Accordingly, RN nodes are now able to subscribe to peer RNs, forming a hierarchy of content distributors.



(a) P1 subscribing, RN1 relay subscription



(b) P3 publishing

Fig. 2: Hierarchical Subscription

In the BCXP implementation, the underlying subscription hierarchy has 2 levels: first when a client subscribes to an RN, and second when an RN receives a new subscription for

an address prefix not in its ST and would initiate a special *relay* subscribe request to all peer RNs that are distributing this content (Figure 2). If NDN routers implement the *best route* forwarding strategy, this additional step would construct a minimum spanning tree at every RN for content distribution.

This extension softens the COPSS requirements. Now, publishers would need only one reachable RN, enabling BCXP to exploit multiple RNs using pure COPSS packets. Clients would also be able to subscribe using relay prefixes directly, and receive traffic from multiple sources in the network for additional reliability.

C. Namespace design

BCXP has 2 generic address schemas, designed to be blockchain agnostic, allowing multiple blockchains to utilize the same network concurrently (possibly also share RNs), and isolate different blockchain traffic types:

- $\{transport\}/bcxp/\{blockchain\}/tx/\{operation\}/\{digest\}?$
- $\{transport\}/bcxp/\{blockchain\}/bk/\{operation\}/\{digest\}?$

The prefix “*transport*” is a COPSS prefix in live mode and simply an empty string in fetch mode, “*blockchain*” field allows the network to distinguish between multiple blockchains sharing the network, “*tx*” & “*bk*” differentiate between transaction and block addresses respectively, while the “*operation*” field allows each blockchain network to define its own operations. As for traffic isolation, each peer is able to narrow down their subscription scope to specific blockchain names and operations only. For example, a peer interested in receiving new blocks would subscribe only to “*/bcxp/mychain/bk/new*”, and receive publish packets related to new blocks only.

The following schema is one possible instance of the generic namespace: “*get*” operation fetch data from full nodes storing the blockchain, “*getheader*” fetch a block’s header and “*sync*” fetch all blocks of the longest chain starting at a specific height.

- $/bcxp/mychain/[bk|tx]/get/\{digest\}$
- $/bcxp/mychain/bk/[getheader|sync]/\{digest\}$

In live mode, COPSS transport-prefixes “*/copss/subscribe*”, “*/copss/relay/subscribe/[peerID]/[ID]*”, “*/copss/publish/rendezvous*”, and “*/copss/publish*” are used to subscribe and broadcast new blocks/transaction as soon they are created using the “*new*” operation.

D. Permissioned blockchains

Permissioned blockchains are special blockchains that can be read/write protected, and sometimes have a multi-step transaction validation process like Hyperledger Fabric. To support permissioned blockchains, we use COPSS two-step mode[10] for live communication, where block meta-data is announced publicly. Here, interested peers would initiate fetch requests to authenticate. Only then, they will be able to fetch data. For blockchains requiring a multi-step transaction validation, new operations are added in the prefix, to allow authorized peers to validate transactions before including them in the system.

E. Lazy Commit Policy (POW only)

An advantage of using BCXP over BlockNDN is the ability for clients to define their own policies to choose which blocks to include in case of a fork, since all new blocks are broadcasted to all peers. Miners following the *lazy commit* policy would listen for incoming blocks before updating the state of their local chain; by broadcast the newly mined block to the network (without disrupting the mining process), and commit instead the first block echoed back from the network. This would give each client a time window, equivalent to the roundtrip time between this client and the nearest RN, to check if another block of the same height is already propagating the network, hence increasing the chance of a miner being on the longest chain.

III. EVALUATION

We investigate the performance of BCXP using computer simulations, built using ns-3[11], the ndnSim module[12][13][14], and the Rocket Fuel 3967[15] network topology. We compare the performance of BCXP to an IP overlay network (UDP with a one-way handshake) and a BlockNDN network (sync interest lifetime larger than simulation run time to optimize performance). In all 3 networks, clients commit the first block they receive as their highest block, while caching stale blocks to be used for fork recovery. The network hash power is divided equally among all peers to calculate the expected time to find a block. This is used as the statistical mean for an exponential random variable indicating the remaining time for a peer to find a block. All simulations were run for 4000 seconds, and are repeated 10 times for statistical confidence. We consider the following evaluation metrics:

- **block propagation delay:** time taken for a block to reach all peers
- **overhead per block:** traffic per link due to propagation of a single block
- **stale block chance:** the ratio of stale blocks committed to the length of the longest chain (fork chance)
- **mean consensus:** the average percentage of peers in consensus during a simulation run
- **min consensus:** the minimum percentage of peers in consensus in a simulation run (worst case 50%)

Unless noted otherwise, the following simulation parameters are used: 10 seconds block time, 70 peers, 5 RNs for BCXP, and a fan out of 6 for each peer in IP overlay networks.

A. Number of rendezvous nodes

We evaluate the effect of increasing COPSS RNs on the performance of BCXP. Propagation delay decreases (200->160 ms) and network overhead increase (50->200 KB) as the number of RNs reach 50. However, for 1 RN, data distribution is centralized, hence a lower stale block chance. Nevertheless, this introduces a single point of failure. We found that having more than 5 RNs improves block propagation delay only by 15 milliseconds. Thus, we conclude that, increasing the number of RNs beyond 5 does not impact performance dramatically.

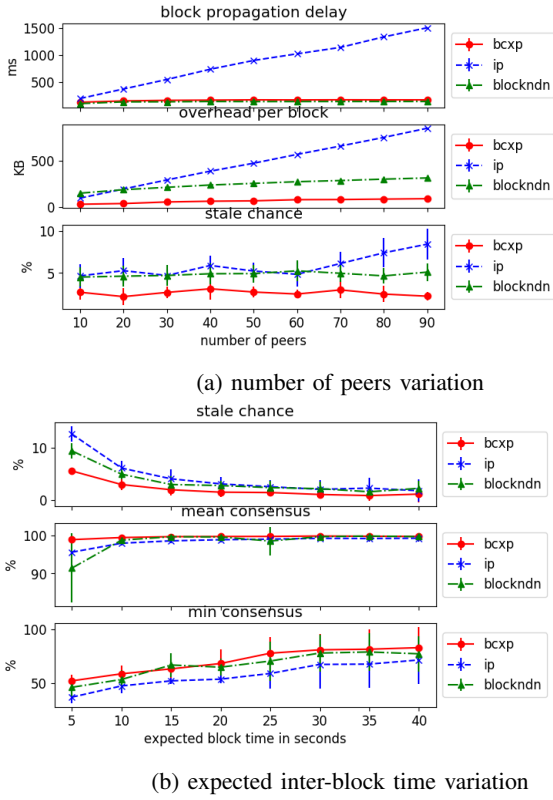


Fig. 3: BCXP, BlockNDN and IP overlay network

B. Scalability

Comparing the performance of the three approaches, Figure 3a shows that blockchain-centric architectures achieve a clear performance advantage. As the number of peers increase, we find that blockchain-centric block propagation delay converges depending on network properties and diameter rather than the number of peers. The overhead per block also decreases dramatically, since blockchain-centric approaches eliminate some traffic redundancy via multicasting. As the number of peers increase, blockchain-centric approaches maintain their stale block chance unlike their IP overlay counterpart due to the relation between propagation delay and stale chance[5].

BCXP achieves the lowest overhead per block 76% that of BlockNDN and 7% that of the IP network, while BlockNDN achieved the lowest propagation delay 77% that of BCXP and 9% that of the IP network.

We also evaluate consensus metrics as inter-block time decreases (Figure 3b). For high block times, all networks performed roughly the same. However, for lower block times blockchain-centric approaches exhibit better performance, with BCXP performing best in metrics of stale block chance, mean consensus over time, and lower consensus bound. Since for low block times, BlockNDN had a very high mean consensus standard deviation, due to network partitions, resulting in an avalanche of forks in some simulation runs. This is because peers follow multiple forks and are not able to recover without a dedicated partition resolution strategy, which is an inherent

problem in ChronoSync[7].

It is worth noting that without the lazy commit policy, BlockNDN has 0.5% lower stale chance than BCXP, due to the lower block propagation delay (40 milliseconds lower)[5].

C. Fully connected IP overlay network

We evaluate as well, a fully connected IP overlay network against blockchain-centric architectures. This unrealistic IP network matches blockchain-centric propagation delay, but as the number of peers increase overhead increases dramatically, reaching 5000% that of BCXP and 1600% that of BlockNDN. Here, the IP network has a comparable stale chance to that of the blockchain-centric protocols.

IV. CONCLUSIONS

This article introduces BCXP, a scalable blockchain network layer. We show using computer simulations how BCXP reduces block propagation delay, minimize network overhead, decrease fork chance, and maintain network stability. Which allows for lower block times and higher transaction throughput than ever possible. However, BCXP's security is yet to be evaluated. For instance NDN namespaces are exposed because they are used in routing, making it vulnerable to censorship.

REFERENCES

- [1] T. Jin, X. Zhang, Y. Liu, et al. Blockndn: A bitcoin blockchain decentralized system over named data networking. *International Conference on Ubiquitous and Future Networks (ICUFN)*, 2017.
- [2] G. Wood. Ethereum: A secure decentralised generalised transaction ledger. *EIP-150 REVISION*.
- [3] J. Yli-Huoma, D. Ko, and S. Choi. Where is current research on blockchain technology? a systematic review. *PLOS*, 2016.
- [4] S. Nakamoto. Bitcoin: A peer-to-peer electronic cash system. *Technical report*, 2009.
- [5] C. Decker, R. Wattenhofer. Information propagation in the bitcoin network. *IEEE International Conference on Peer-to-Peer Computing*, 2013.
- [6] I.E. Adem, E. Gencer, E.G. Sirer. Bitcoin-ng: A scalable blockchain protocol. *Usenix Conference on Networked Systems Design and Implementation*, 2015.
- [7] Z. Zhu, A. Afanasyev. Let's chronosync: Decentralized dataset state synchronization in named data networking. *IEEE International Conference on Network Protocols (ICNP)*, 2013.
- [8] L. Zhang, D. Estrin, J. Burke, et al. Named data networking (ndn) project. *Technical report NDN-0001, PARC*, 2010.
- [9] G. Xylomenos, C.N. Ververidis, V.A. Siris, et al. A survey of information-centric networking research. *IEEE Communications Surveys & Tutorials*, 16, 2013.
- [10] J. Chen, M. Arumathurai, L. Jiao, et al. Copss: An efficient content oriented publish/subscribe system. *ACM/IEEE Seventh Symposium on Architectures for Networking and Communications Systems (ANCS)*, 2011.
- [11] Riley, George F. and Henderson, et al. The ns-3 network simulator. In *Modeling and Tools for Network Simulation*, pages 15–34. Springer, 2010.
- [12] A. Afanasyev, I. Moiseenko, and L. Zhang. ndnsim: Ndn simulator for ns-3. *NDN, Technical Report NDN-0005*, 2012.
- [13] S. Mastorakis, A. Afanasyev, I. Moiseenko, et al. ndnsim 2: An updated ndn simulator for ns-3. *NDN, Technical Report NDN-0028, Revision 2*, 2016.
- [14] S. Mastorakis, A. Afanasyev, and L. Zhang. On the evolution of ndnsim: an open-source simulator for ndn experimentation. *ACM Computer Communication Review*, 2017.
- [15] N. Spring, R. Mahajan, D. Wetherall, et al. Measuring isp topologies with rocketfuel. *IEEE/ACM Transactions on Networking (TON)*, 2004.