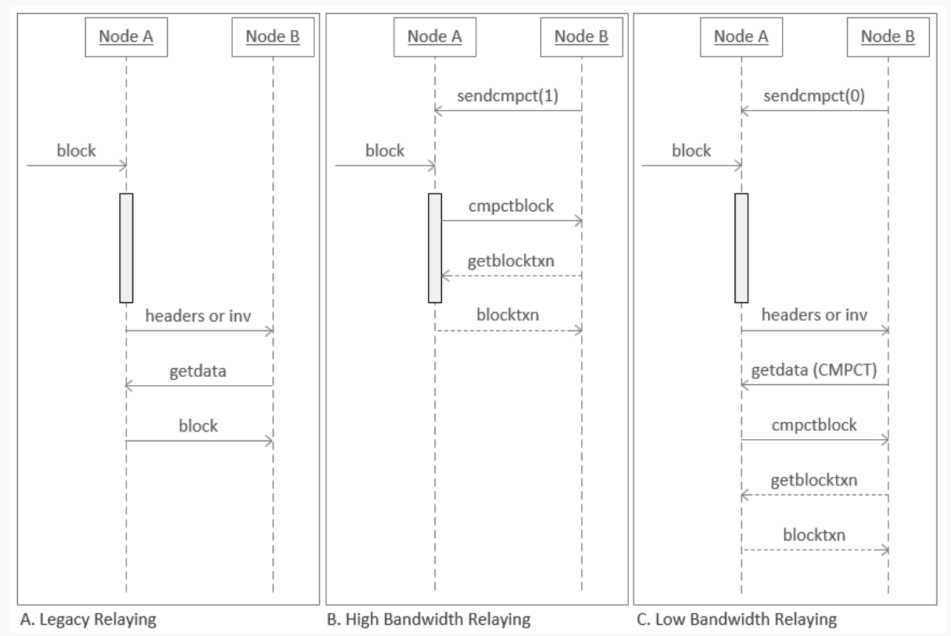
BLOCK PROPAGATION

Overview	Block propagation is important for scaling, as it can be the bottle-neck for faster confirmation of transactions, and maintenance of decentralization as transaction load on public blockchains increases. Where propagation is a large enough issue in the network, head-first mining has emerged - on an incoming block, only the header is validated before work begins on the next block. This limits the chances of working on an orphaned block but can compromise the security of the network as transactions are not validated. Weaknesses in propagation can cause miners to include fewer transactions, as larger blocks would have less chance of being added to the chain. This would artificially reduce the block size and increase transaction fees on the network. Currently the standard for block propagation in Bitcoin is Compact Blocks (also used in BitcoinABC) with very few orphaned blocks, and for Bitcoin Unlimited on Bitcoin Cash is Xthin blocks. After Segwit, all miners had to use Bitcoin Core clients to support the soft fork, so continual improvements that had been made in Core were adopted all at once, rapidly improving the block propagation. In Ethereum, the faster block times increase the need for faster block propagation, but to solve this, Ethereum implemented GHOST, which gave small rewards for mining orphaned blocks. The Ethereum Wire protocol has yet to implement many of the changes on the Bitcoin network, as development is focused on sharding, creating light clients, and Proof-of-Stake, but would likely implement some of the improvements of Bitcoin Core and Bitcoin Unlimited, like Graphene or Compact Blocks.
Timeline of Past Projects	• **2012** - Stratum protocol for mining pools is released, showed the need for better propagation, and advantages of trusted parties in mining.[1] • **2013** - Initial bloom filtered block attempts to improve the original block relay protocol, but were not practical.[2] • **2013** - Initial Bitcoin Relay network created by Matt Corallo to combat miners being hesitant to share IP addresses (fear of DoS). Came right before the Fast block relay protocol; keep track of last 10K transactions, send 2 byte codes of transactions in .5 RTT. Too much overhead for the full network, but worked in hub-spoke model.[3] • **2014** – Block network coding was being tested to get the block as small as possible but was too computationally expensive to fix anything. Required more processing time after blocks were received than was saved on transmission.[4] • **2016** -Falcon: uses cut routing to forward each packet as soon as it is received. Requires a trusted administrator, in this case the Cornell labs that maintain it. Very little information and is not as widely used as FIBRE. Not clear if it is used anymore.[5]

BLOCK PROPAGATION

Current Implementati -ons	• **2013-2016** - Compact Block Relay (BIP152): Two different modes. Ranges from .5 to 1.5 protocol RTT for High Bandwidth, and 1.5 to 2.5 for low bandwidth mode. Reduces block transmission by 99% (originally 12% of overall network bandwidth). In high bandwidth mode, the blocks turn to 20kb in size.[6]



- **2014- 2016** - XThin/Xpedited, Bitcoin unlimited: Developed in parallel with Compact blocks. Uses bloom filters to compress the block, but had bugs. Xpedited is a manual configuration added afterwards for high bandwidth mode. No longer used on BTC because of Segwit but are used on Bitcoin Cash.[7] XVal is an additional transaction validation optimization from mempool to new block.[8] Xthin compression is similar in size to BIP152 working on 1mb blocks but has a super-linear factor in size, so is worse on extremely large blocks. Will be replaced by Graphene.
- **2016-2017** - FIBRE: Uses erasure coding and network coding to implement a fast .5 RTT with little latency over UDP. This is its own topology and connects to the rest of the network, and acts as a single distributed node. Fixes some of the latency in compact blocks. Must be run by a single domain administration, meaning there is some trust involved outside the Bitcoin protocol for these nodes.[9]
- **2017** - Blockstream satellite: Acts as a single node around the world, relaying transactions and blocks through its coverage area. This uses the Bitcoin FIBRE network as well. Only has about 80 kb/second, so would benefit from additional size reduction in block transmission.[10]

BLOCK PROPAGATION

Current Proposals	• **Graphene:** Uses even less bandwidth than compact blocks, with inverted bloom lookup tables for mempool set reconciliation between peers, and bloom filters for the resulting block compression. This reduces bandwidth, but has drawbacks in latency.[11] Also makes some assumptions about transaction ordering that isn't necessarily true in Bitcoin.[12] This is still in development and not yet tested in the real Bitcoin Cash network. There might be further improvements for latency on this with polynomial set reconciliation instead of the IBLT's, and could also be used across the Great firewall of China with some modifications in Bitcoin Core. This technology also could be used with other blockchains.

PROTOCOL 1: Graphene

1: Sender: Sends *inv* for a block.
2: Receiver: Requests unknown block; includes count of txns in her IDpool, m.
3: Sender: Sends Bloom filter $\mathcal{S}$ and IBLT $\mathcal{I}$ (each created from the set of n txn IDs in the block) and essential Bitcoin header fields. The FPR of the filter is $f = \frac{a}{m-n}$, where $a = n/(c\tau)$.
4: Receiver: Creates IBLT $\mathcal{I}'$ from the txn IDs that pass through $\mathcal{S}$. She decodes the *subtraction* [4] of the two blocks, $\mathcal{I} \triangle \mathcal{I}'$.

- **IPFS and Swarm:** Early stage initiative to store the blockchain data, and potentially the mempool on IPFS[13]/ Swarm[14], which would allow for higher consistency in mempool, additional connections/new topologies in the network, and also new protocols for ordering/structuring of transactions in the mempool, which can allow higher compression of blocks for propagation.
- **Template Delta Compression:** By Greg Maxwell; peers keep track of a block template in advance of receiving a block to help with the drawbacks of using set reconciliation. With this so far in testing, the actual transmission for the block itself is around 3000 bytes, and 22% of the blocks can fit within one IP packet.[12] There is necessary computation, but for some cases, this can be better than what is currently used, like the Blockstream satellite or similar hub and spoke type of networks. Helps miners include more transactions, as template/"weak" blocks[15] can be sent frequently and block relay would just include the template delta of the chosen weak block.
- **FIBRE- like swarming:** Would require a soft fork to turn single domain consensus of FIBRE into untrusted peer-to-peer consensus. This would commit the erasure coding into consensus, which would be more CPU intensive. This would break block truncation for

BLOCK PROPAGATION

	miners to have malleable coinbase transactions, so this proposal will need more improvement.[12]
Transaction Relay Overview	Regardless of block size increase, efficiencies in transaction space usage will increase the amount of transactions per block, which is most of the bandwidth of a node. While it is not a large bottleneck now, it will need to be improved to prevent future issues, especially in fractured parts of the internet.
Transaction Relay Proposals	**UTXO commitments in BitTorrent blocks:** Data Structures to do so were created by by Bram Cohen, the founder of BitTorrent, this would involve merkelizing the UTXO's to be stored to fit with the how BitTorrent remembers what parts of which file have been shared/ recieved. This seems to be forgotten/not prioritized as Cohen has started on his own ICO, but others may continue this work.[16] **Individual Transaction compression:** By Pieter Wuille and Greg Maxwell, on average reduces transaction sizes by 28%. May be too CPU intensive, but is in the very early stages of development. This could either be put inside blocks to cause an increase in transactions per block, or done outside just for relaying, in which it would reduce bandwidth usage for the network.[12] There may be applications with this for initial node synching to the network. **Set reconciliation for transaction relay:** Reduces the average relays per transaction, which would decrease its propagation, and confine it to cycles. This can be mitigated with periodic mempool set reconciliation to send transactions between cycles. This would reduce the amount of bandwidth used by transaction relay.[12]
People to Follow	**Greg Maxwell** – Proposed FIBRE-like swarming, Template Delta Compression, has been thinking a lot about coding schemes to reduce transaction sizes, and has been very involved in the all of the work for transaction propagation. **Matt Corallo** – Maintainer of the biggest FIBRE Network, authored BIP152 (compact blocks) **Peter Rizun** – One of the authors of XThin and Xpedited **Gavin Andreesen** – Proposed one of the first set reconciliation schemes and Invertible Bloom Lookup Tables[17]. His work was turned into XThin/Xpedited blocks, and was one of the publishers of the Graphene Paper. **Pieter Wuille** – Worked on BIP152, and has worked on individual transaction compression Not his main development focus area. **Peter Tschipper** – Worked on XThin and Xpedited.

BLOCK PROPAGATION

	A. Pinar Ozisik, George Bissias, Amir Houmansadr, Brian N. Levine – coauthors of Graphene **Viktor Trón** – Worked on IPFS and Swarm for blockchain storage, but has been tabled for now.
Predictions	High throughput blockchains or off-chain micropayment services like Lightning offer new potential to some funding models that previously were too expensive on the regular Bitcoin blockchain. To mitigate the security flaws of head-first mining, there may be micropayment services where block relayers are paid small amounts for correct headers, and penalized for incorrect/non consensus headers. A candidate for this would be Blockstream, who has already stated that they will at some point monetize their satellite API. Currently, the main FIBRE network is being hosted and paid for by Matt Corallo, which is unsustainable in the long term. The lightning network could make the donations feasible again, or instead create the opportunity for businesses that propagate blocks to full nodes/miners. This might be similar to a subscription service for connecting to FIBRE, etc. This would likely not create an immediate change, but could increase competition and create a diversity of options/technologies for block propagation in Bitcoin and other blockchains like Ethereum. They also may help create a financial incentive to providing more internet/blockchain infrastructure in parts of the world, like in Africa and South America, which would help scale networks to more users. I also expect additional subnetworks/topologies that are optimized for specific problems, like FIBRE is for fast worldwide communication. Among these problems would be the Great Firewall of China, or other nations that may restrict crypto currencies as the space matures. Finally, many of the developments and optimizations by the Bitcoin Core and Bitcoin Unlimited teams will likely be adopted for other blockchains, as they pertain to general distributed systems and not just Bitcoin in general.
References:	[1]- https://slushpool.com/help/manual/stratum-protocol [2]- https://github.com/bitcoin/bips/blob/master/bip-0037.mediawiki [3]- http://bitcoinrelaynetwork.org/ [4]- https://en.bitcoin.it/wiki/User:Gmaxwell/block_network_coding [5]- https://www.falcon-net.org/ [6]- https://bitcoincore.org/en/2016/06/07/compact-blocks-faq/ [7]- https://medium.com/@peter_r/towards-massive-on-chain-scaling-presenting-our-block-propagation-results-with-xthin-da54e55dc0e4 and https://bitco.in/forum/threads/buip010-passed-xtreme-thinblocks.774/

[8]- https://bitco.in/forum/threads/xpress-validation-request-for-input.895/

[9]- http://bitcoinfibre.org/public-network.html

[10]- https://blockstream.com/satellite/howitworks/

[11]- https://people.cs.umass.edu/~gbiss/graphene.pdf

[12]- http://diyhpl.us/wiki/transcripts/gmaxwell-2017-11-27-advances-in-block-propagation/ or https://www.youtube.com/watch?v=EHIuuKCm53o

[13]- https://github.com/ethereum/go-ethereum/issues/2050

[14]- https://github.com/ethereum/go-ethereum/issues/2049

[15]- https://people.xiph.org/~greg/weakblocks.txt

[16]- https://www.youtube.com/watch?v=52FVkHlCh7Y

[17]- https://gist.github.com/gavinandresen/e20c3b5a1d4b97f79ac2