Filter / Category	Can Help	Resource	Tooltip Candidate	Jen's suggestion	Modified?
noise	Vlad	https://github.com/libp2p/specs/blob/master/noise/README.md	The Noise Protocol is a framework for building security protocols	The Noise Protocol is a framework for building cryptographic protocols	Unmodified
Handshake	Vlad	https://github.com/libp2p/specs/blob/master/noise/README.md	Every Noise connection begins with a handshake between an initiating peer and a responding peer to efficiently encrypt traffic.	Every Noise connection begins with a handshake between an initiating peer and a responding peer.	Unmodified
Failed to decrypt	Vlad				Unmodified
kad-dht/1.0.0	Zura?	https://github.com/libp2p/specs/blob/master/kad-dht/README.md	Used for peer discovery. The Kademlia Distributed Hash Table (DHT) is a DHT implementation largely based on the Kademlia whitepaper, augmented with notions from S/Kademlia, Coral and the BitTorrent DHT		Unmodified
Find Node	Zura?	https://github.com/libp2p/specs/blob/master/kad-dht/README.md	Finding the closest nodes to a given key.		Unmodified
Put Value	Zura?	https://github.com/libp2p/specs/blob/master/kad-dht/README.md	Value storage. Putting the value to the closest nodes		Unmodified
Get Value	Zura?	https://github.com/libp2p/specs/blob/master/kad-dht/README.md	Value retrieval. Getting a value by its key from the nodes closest to that key		Unmodified
Add Provider	Zura?	https://github.com/libp2p/specs/blob/master/kad-dht/README.md	Adding oneself to the list of providers for a given key at the nodes closest to that key by finding the closest nodes		Unmodified
Get Providers	Zura?	https://github.com/libp2p/specs/blob/master/kad-dht/README.md	Getting providers for a given key from the nodes closest to that key		Unmodified
Ping	Zura?	https://github.com/libp2p/specs/blob/master/kad-dht/README.md	Deprecated. Message type replaced by the dedicated ping protocol		Unmodified
/ipfs/peer-exchange	-				Unmodified
Peer Exchange	-	https://nmw.xyz/noisewtf/.eH7-kYVZ6BBBTXCQPppwbs	Keep the mesh connected when closing	Keeps the mesh connected when closing connections	Unmodified
/mina/bitswap-exchange	-	https://github.com/libp2p/docs/content/bitswap/	Bitswap is a module provided by libp2p that enables distributed data synchronization over a p2p network, somewhat comparable to how BitTorrent works		Unmodified
Bitswap Exchange	-	https://github.com/libp2p/specs/blob/master/bitswap	Using Bitswap, Mina nodes will be able to offset load from Gossipsub, by gossiping only the header of the block, SplitUnconnected body into parts. Download different parts of the block's body	Using Bitswap, Mina nodes can offset load from Gossipsub by gossiping only the header of the block, SplitUnconnected	Unmodified
/mina/node-status	-	https://github.com/libp2p/specs/blob/master/README	NodeStatus is maintained by libp2p_helper process, which is updated by main Mina process using SetNodeStatus call		Unmodified
Node Status	-	https://github.com/libp2p/specs/blob/master/README	As soon as you open substream on this protocol, you will receive a json encoded NodeStatus.		Unmodified
/ipfs/id/1.0.0	Vlad	https://github.com/libp2p/specs/blob/master/identify/README	The identify protocol has the protocol id /ipfs/id/1.0.0, and it is used to query remote peers for their information		Unmodified
Identify	Vlad	https://github.com/libp2p/specs/blob/master/identify/README	The protocol works by opening a stream to the remote peer you want to query. The peer being identified responds by returning an Identify message and closes the stream.	The protocol works by opening a stream to the remote peer you want to query. The peer being identified returns an Identify message.	Unmodified
/ipfs/id/push/1.0.0	Vlad	https://github.com/libp2p/specs/blob/master/identify/README	The identify/push protocol has the protocol id /ipfs/id/push/1.0.0, and it is used to inform known peers about changes that occur at runtime.		Unmodified
IPFS Push	Vlad	https://github.com/libp2p/specs/blob/master/identify/README	Used for pushing the response to identify request with the full identity info to the peer	Responds to identify request by pushing the full identity info to the peer	Unmodified
/p2p/id/delta/1.0.0	Vlad	https://github.com/libp2p/specs/blob/master/identify			Unmodified
IPFS Delta	Vlad	https://github.com/libp2p/specs/blob/master/identify	Used for providing changes (only diff) in the list of protocols that the peer supports.		Unmodified
meshsub/1.1.0	Zura	https://github.com/libp2p/specs/blob/master/pubsub/gossipsub	Mina Gossipsub is a PubSub (Publish/Subscribe) protocol. Using Gossipsub, different messages are published/gossiped through the network. This is used to provide peers with updates regarding the latest state of the blockchain or its network. Each node can, after handshaking and agreeing on this protocol with a peer, publish and subscribe to different topics.		Unmodified
Subscribe	Zura	https://docs.libp2p.io/concepts/publish-subscribe/	Peers keep track of which topics their directly-connected peers are subscribed to. Using this information, each peer is able to build up a picture of the topics around them and which peers are subscribed to each topic.		Unmodified
Unsubscribe	Zura	https://docs.libp2p.io/concepts/publish-subscribe/	When a peer unsubscribes from a topic, it will notify its full-message peers that their connection has been pruned at the same time as sending their unsubscribe messages.		Unmodified
Control	Zura	https://github.com/libp2p/specs/blob/master/pubsub/gossipsub	Control messages are exchanged to maintain topic meshes and emit gossip.		Unmodified
Publish External Transition	Zura	https://github.com/libp2p/specs/blob/master/pubsub/gossipsub	Broadcasts newly produced blocks throughout the network.		Unmodified
Publish Snark Pool Diff	Zura	https://github.com/libp2p/specs/blob/master/pubsub/gossipsub	Broadcasts snark work from mempools throughout the network. Snark coordinators broadcast locally produced snarks in an interval for a period of time after creation, and all nodes rebroadcast externally produced snarks if they were relevant and could be added to the their mempool		Unmodified
Publish Transaction Pool Diff	Zura	https://github.com/libp2p/specs/blob/master/pubsub/gossipsub	Broadcasts transactions from mempools throughout the network. Nodes broadcast locally submitted transactions on an interval for a period of time after creation, as well as rebroadcast externally submitted transactions if they were relevant and could be added to the their mempool		Unmodified
coda/rpcs/0.0.1	Alex	https://github.com/o1-labs/mina/blob/develop/rpcs	RPC - Remote Procedure Call. In some scenarios, a node needs to obtain information from other nodes it is connected to. It makes a particular request to a peer over the P2P network, and then expects a response.		Unmodified
Get some initial peers	Alex	https://github.com/o1-labs/mina/blob/develop/rpcs	Returns known peers to the requester. Currently used to help nodes quickly connect to the network.		Unmodified
Get staged ledger aux and pending coinbases at hash	Alex	https://github.com/o1-labs/mina/blob/develop/rpcs	Returns the staged ledger data associated with a particular block on the blockchain. This is requested by bootstrapping nodes and is used to construct the root staged ledger of the frontier. Subsequent staged ledgers are built from the root using the staged ledger diff information contained in blocks.		Unmodified
Answer sync ledger query	Alex	https://github.com/o1-labs/mina/blob/develop/rpcs	Serves merkle ledger information over a "sync ledger" protocol. The sync ledger protocol works by allowing nodes to request intermediate hashes and leaves of data in a target ledger they want to synchronize with. Using the protocol, requesters are able to determine which parts of their existing ledger need to be updated, and can avoid downloading data in the ledger which is already up to date. Requests are distributed to multiple peers to spread the load of serving sync ledgers throughout the network.		Unmodified
Get transition chain	Alex	https://github.com/o1-labs/mina/blob/develop/rpcs	Returns a bulk bulk set of blocks associated with a provided set of state hashes. This is used by the catchup routine when it is downloading old blocks to re-synchronize with the network over short distances of missing information. At most X blocks (where the maximum number of blocks that can be requested in a single batch is X) can be requested, and no more than 20 will result in no response.		Unmodified
Get ancestry	Alex	https://github.com/o1-labs/mina/blob/develop/rpcs	Returns the target block corresponding to the provided state hash along with the root block of the frontier, and a merkle proof that attests to the connection of blocks linking the target block to the root block.		Unmodified
Get transition chain proof	Alex	https://github.com/o1-labs/mina/blob/develop/rpcs	Returns a transition chain proof for a specified block on the blockchain. A transition chain proof proves that, given some block b1, there exists a block b0 which is k blocks preceding b1. To prove this, a node receives H(b1), and returns H(b0) along with a merkle proof of all the intermediate "state body hashes" along the path b0 -> b1. The requester checks this proof by re-computing H(b1) from the provided merkle proof.		Unmodified
Get transition knowledge	Alex	https://github.com/o1-labs/mina/blob/develop/rpcs	Returns the a list of k state hashes of blocks from the root of the frontier (point of finality) up to the current best tip (most recent block on the canonical chain).		Unmodified
Ban Notify	Alex	https://github.com/o1-labs/mina/blob/develop/rpcs	Notifies a peer that they have been banned. The request contains the time at which the requester banned the recipient.		Unmodified
Get best tip	Alex	https://github.com/o1-labs/mina/blob/develop/rpcs	Returns the best tip block along with the root block of the frontier, and a merkle proof that attests to the connection of blocks linking the target block to the root block. If the root has been finalized, this proof will be of length k, allowing the best tip to be finalized to a block at top of the root block.		Unmodified
Get node status	Alex	https://github.com/o1-labs/mina/blob/develop/rpcs	This acts as a telemetry RPC, which asks a peer to provide information about their node status. Daemons do not have to respond to this request, and node operators may pass a command line flag to opt out of responding to these node status requests.		Unmodified
Get epoch ledger	Alex	https://github.com/o1-labs/mina/blob/develop/rpcs	Returns sparse ledger, or a string description in case of a failure.	Returns sparse ledger, or, in case of a failure, a string description.	Unmodified
Versioned rpc menu	Alex				Unmodified
/multistream/1.0.0	-				Unmodified
Multistream	-				Unmodified
/coda/mplex/1.0.0	-				Unmodified
Coda mplex	-				Unmodified
Unknown	-				Unmodified
Unknown	-				Unmodified