How to read the sequencer feed

Running an Arbitrum relay locally as a feed relay lets you subscribe to an uncompressed sequencer feed for real-time data as the sequencer accepts and orders transactions off-chain.

When connected to websocket port 9642 of the local relay, you'll receive a data feed that looks something like this:

{ "version" :

1 , "messages" :

[{ "sequenceNumber" :

25757171 , "message" :

{ "message" :

{ "header" :

{ "kind" :

3 , "sender" :

"0xa4b000000000000000000073657175656e636572" , "blockNumber" :

16238523 , "timestamp" :

1671691403 , "requestId" :

null , "baseFeeL1" :

null } , "l2Msg" :

"BAL40oKksUiElQL5AISg7rsAgxb6o5SZbYNoIF2DTixsqDpD2xII9GJLG4C4ZAhh6N0AAAAAAAAAAAAAAAAC7EQiq1R1VYgL3/oXgvD921hYRyAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
} , "delayedMessagesRead" :

354560 } , "signature" :

null }] } Breaking this feed down a bit: the top-level data structure is defined by the BroadcastMessage struct :

type BroadcastMessage struct

{ Version int

json:"version" // Note: the "Messages" object naming is slightly ambiguous: since there are different types of messages Messages [ ] * BroadcastFeedMessage json:"messages,omitempty" ConfirmedSequenceNumberMessage * ConfirmedSequenceNumberMessage json:"confirmedSequenceNumberMessage,omitempty" } The messages field is the BroadcastFeedMessage struct :

type BroadcastFeedMessage struct

{ SequenceNumber arbutil . MessageIndex json:"sequenceNumber" Message arbstate . MessageWithMetadata json:"message" Signature [ ] byte

json:"signature" } Each message conforms to arbstate.MessageWithMetadata :

type MessageWithMetadata struct { Message *arbos.L1IncomingMessage json:"message" DelayedMessagesRead uint64 json:"delayedMessagesRead" } Finally, we get the transaction's information in the message subfield as an L1IncomingMessage :

type L1IncomingMessage struct

{ Header * L1IncomingMessageHeader json:"header" L2msg [ ] byte

json:"l2Msg" // Only used for L1MessageType_BatchPostingReport BatchGasCost * uint64

json:"batchGasCost,omitempty" rlp:"optional" } You can use the ParseL2Transactions function to decode the message.

Using the feed relay, you can also retrieve the L2 block number of a message:

- On Arbitrum One
- , this can be done by adding the Arbitrum One genesis block number (22207817) to the sequence number of the feed message.
- Note that in the case of Arbitrum Nova
- , the Nitro genesis number is 0
- , so it doesn't need to be included when adding to the feed message's sequence number Edit this page Last updated on Mar 19, 2024 Previous How to run a feed relay Next How to run a Sequencer Coordinator Manager (SQM)