Transaction

(go-ethereum version 1.8)

[core/types/transaction.go]

Transaction

```
type txdata struct {
    AccountNonce uint64          `json:"nonce"    gencodec:"required"`
    Price        *big.Int        `json:"gasPrice" gencodec:"required"`
    GasLimit     uint64          `json:"gas"      gencodec:"required"`
    Recipient    *common.Address `json:"to"       rlp:"nil"
                                                      // nil means contract
creation
    Amount       *big.Int        `json:"value"    gencodec:"required"`
    Payload      []byte          `json:"input"    gencodec:"required"`

    // Signature values
    V *big.Int `json:"v" gencodec:"required"`
    R *big.Int `json:"r" gencodec:"required"`
    S *big.Int `json:"s" gencodec:"required"`

    // This is only used when marshaling to JSON.
    Hash *common.Hash `json:"hash" rlp:"-"`
}
```

to address

value (Wei)

input data

References : [C1]

Transaction Ordering Examples

Naive approach implementation examples (FIXME)

```
//approach 1
transactions := self.eth.TxPool().GetTransactions()
sort.Sort(types.TxByNonce{transactions})
//approach 2
transactions := self.eth.TxPool().GetTransactions()
sort.Sort(types.TxByPriceAndNonce{transactions})
```

Transaction

order by price & nonce

```
 // approach 3
// commit transactions for this run.
txPerOwner := make(map[common.Address]types.Transactions)
// Sort transactions by owner
for _, tx := range self.eth.TxPool().GetTransactions() {
    from, _ := tx.From() // we can ignore the sender error
    txPerOwner[from] = append(txPerOwner[from], tx)
}
var (
    singleTxOwner types.Transactions
    multiTxOwner  types.Transactions
)
// Categorize transactions by
// 1. 1 owner tx per block
// 2. multi txs owner per block
for _, txs := range txPerOwner {
    if len(txs) == 1 {
        singleTxOwner = append(singleTxOwner, txs[0])
    } else {
        multiTxOwner = append(multiTxOwner, txs...)
    }
}
sort.Sort(types.TxByPrice{singleTxOwner})
sort.Sort(types.TxByNonce{multiTxOwner})
transactions := append(singleTxOwner, multiTxOwner...)
```

sort by owner

input data

order by owner