HRD Metrics

Network Revenue

Network revenues are the sum of `transactions fees` and `stake rewards`. According to mirror node sources, a transaction has the following types (source1, source2):

```
const transactionTypes = {
    0:  "UNKNOWN",
    7:  "CONTRACTCALL",
    8:  "CONTRACTCREATEINSTANCE",
    9:  "CONTRACTUPDATEINSTANCE",
    10: "CRYPTOADDLIVEHASH",
    11: "CRYPTOCREATEACCOUNT",
    12: "CRYPTODELETE",
    13: "CRYPTODELETELIVEHASH",
    14: "CRYPTOTRANSFER",
    15: "CRYPTOUPDATEACCOUNT",
    16: "FILEAPPEND",
    17: "FILECREATE",
    18: "FILEDELETE",
    19: "FILEUPDATE",
    20: "SYSTEMDELETE",
    21: "SYSTEMUNDELETE",
    22: "CONTRACTDELETEINSTANCE",
    23: "FREEZE",
    24: "CONSENSUSCREATETOPIC",
    25: "CONSENSUSUPDATETOPIC",
    26: "CONSENSUSDELETETOPIC",
    27: "CONSENSUSSUBMITMESSAGE",
    28: "UNCHECKEDSUBMIT",
    29: 'TOKENCREATION',
    31: 'TOKENFREEZE',
    32: 'TOKENUNFREEZE',
    33: 'TOKENGRANTKYC',
    34: 'TOKENREVOKEKYC',
    35: 'TOKENDELETION',
    36: 'TOKENUPDATE',
    37: 'TOKENMINT',
    38: 'TOKENBURN',
    39: 'TOKENWIPE',
    40: 'TOKENASSOCIATE',
    41: 'TOKENDISSOCIATE',
    42: "SCHEDULECREATE",
    43: "SCHEDULEDELETE",
    44: "SCHEDULESIGN",
    45: "TOKENFEESCHEDULEUPDATE",
    46: "TOKENPAUSE",
    47: "TOKENUNPAUSE",
    48: "CRYPTOAPPROVEALLOWANCE",
    49: "CRYPTODELETEALLOWANCE",
    50: "ETHEREUMTRANSACTION",
```

```
    51: "NODESTAKEUPDATE",
    52: "UTILPRNG",
    53: "TOKENUPDATENFTS",
    54: "NODECREATE",
    55: "NODEUPDATE",
    56: "NODEDELETE",
    57: "TOKENREJECT",
};
```

We can use these types to get the information we need.

Total Network Revenue

We need to find and sum up this values: `all transactions fees` + `stake reward`

All transactions fees:

```
query TotalTxFees($startTimestamp: bigint!, $endTimestamp: bigint!) {
  transaction_aggregate(
    where: {
      consensus_timestamp: {
        _gte: $startTimestamp,
        _lte: $endTimestamp
      }
    }
  ) {
    aggregate {
      sum {
        charged_tx_fee
      }
    }
  }
}
```

Stake rewards:

```
query StakeRewards($startTimestamp: bigint!, $endTimestamp: bigint!) {
  staking_reward_transfer_aggregate(
    where: {
      consensus_timestamp: {
        _gte: $startTimestamp,
        _lte: $endTimestamp
      }
    }
  ) {
    aggregate {
      sum {
        amount
      }
    }
  }
}
```

Income

For income sections we can use this query with different transaction types:

```
query TxFeesByTxTypes($startTimestamp: bigint!, $endTimestamp: bigint!, $txTypes:
[smallint!]!) {
  transaction_aggregate(
    where: {
      consensus_timestamp: {
        _gte: $startTimestamp,
        _lte: $endTimestamp
      },
      type: {
        _in: $txTypes
      }
    }
  ) {
    aggregate {
      sum {
        charged_tx_fee
      }
    }
  }
}
```

Hedera Consensus Service (HCS) Income

HCS Transaction Types:

```
const transactionTypesHSC = {
    24: "CONSENSUSCREATETOPIC",
    25: "CONSENSUSUPDATETOPIC",
    26: "CONSENSUSDELETETOPIC",
    27: "CONSENSUSSUBMITMESSAGE",
};
```

We will use:

```
const txTypes = [24, 25, 26, 27]
```

Hedera Token Service (HTS) Income

HTS Transaction Types:

```
const transactionTypesHTS = {
    29: 'TOKENCREATION',
    31: 'TOKENFREEZE',
    32: 'TOKENUNFREEZE',
    33: 'TOKENGRANTKYC',
    34: 'TOKENREVOKEKYC',
    35: 'TOKENDELETION',
    36: 'TOKENUPDATE',
    37: 'TOKENMINT',
```

```
    38: 'TOKENBURN',
    39: 'TOKENWIPE',
    40: 'TOKENASSOCIATE',
    41: 'TOKENDISSOCIATE',
    45: "TOKENFEESCHEDULEUPDATE",
    46: "TOKENPAUSE",
    47: "TOKENUNPAUSE",
    53: "TOKENUPDATENFTS",
    57: "TOKENREJECT",
};
```

We will use:

```
const txTypes = [29, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 45, 46, 47, 53, 57]
```

Hedera Smart Contract Service (HSCS) Income

HSCS Transaction Types:

```
const transactionTypesHSCS = {
    7:  "CONTRACTCALL",
    8:  "CONTRACTCREATEINSTANCE",
    9:  "CONTRACTUPDATEINSTANCE",
    22: "CONTRACTDELETEINSTANCE",
};
```

We will use:

```
const txTypes = [7, 8, 9, 22]
```

Other Income

For **Other** Income we can use simillar query but with `_nin` operator:

```
query TxFeesWithoutTxTypes($startTimestamp: bigint!, $endTimestamp: bigint!, $txTypes:
[smallint!]!) {
  transaction_aggregate(
    where: {
      consensus_timestamp: {
        _gte: $startTimestamp,
        _lte: $endTimestamp
      },
      type: {
        _nin: $txTypes
      }
    }
  ) {
    aggregate {
      sum {
        charged_tx_fee
      }
    }
  }
}
```

Pass to `txTypes` all txTypes from previous income sections:

```
const txTypes = [...txTypesHCS, ...txTypesHTS, ...txTypesHSCS]
```

Deposits

Deposits refer to the HBAR tokens that participants lock or stake for various purposes, such as operating nodes, participating in staking for rewards, ensuring governance integrity by council members, and funding the treasury for network development and growth.

Staking Deposits

```
query TotalHbarStaked {
  network_stake(
    order_by: {consensus_timestamp: desc}
    limit: 1
  ) {
    stake_total
  }
}
```