Summary – Ethereum network statistic

Huajing Yu

I. Experiments

We introduce the experimental settings and results.

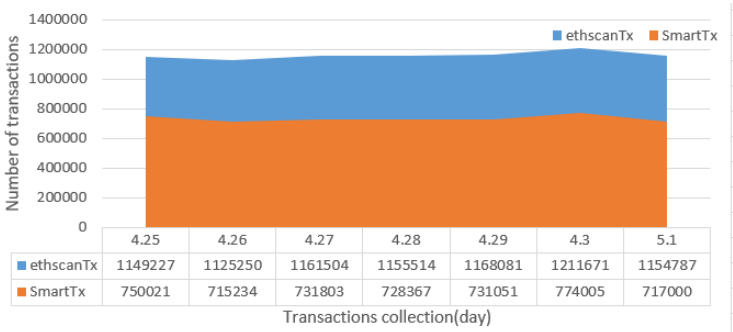
A. settings

We deployed a full Ethereum node in China collecting transactions information in a week and divided them into transfer transactions, contract transactions. We define a transaction as a transfer transaction if it is only used for exchanging currency. We define a transaction with some input data as a contract transaction, either its receiver has deployed a smart contract and the data are methods that are used to trigger the code, or its receiver is empty and the data are bytecodes that are used to deploy a smart contract.

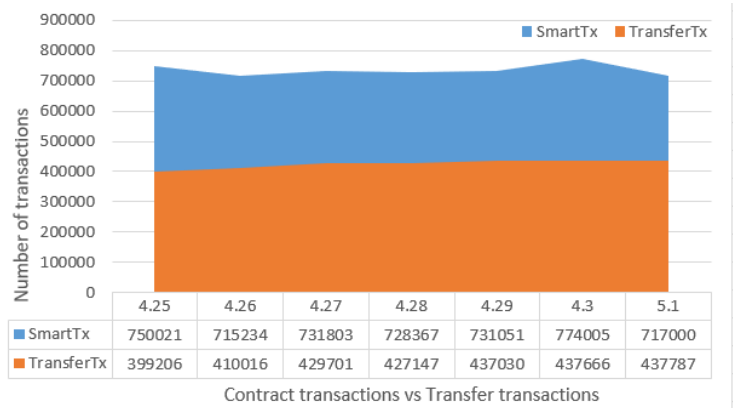
We collected 8,126,034 transactions in a week from Apr. 25 to Fri. 1 2022, there are 5,147,481 contract transactions and 2,987,553 transfer transactions(as shown in table I). We present the contract transactions in Figure 1, where we show the numbers of contract transactions in total transaction and the numbers of both the contract and transfer transactions. Figure I-A shows the contract transactions number in the total transactions. Figure I-A shows the comparison between the contract and transfer transactions. As shown in Figure 1, the contract transactions takes 63.35% of the total transactions and 99.91% gas used in a week, while the remaining 36.65% transfer transactions only cost 0.01% gas used in a week.

TABLE I
Block transactions.

Time	Blocks	Conact transactions	Transfer transactions	Total
Apr.25	6341	750,021	399,206	1,149,227
Apr.26	6319	715,234	410,006	1,125,250
Apr.27	6380	731,803	429,701	1,161,504
Apr.28	6347	728,367	427,147	1,155,514
Apr.29	6375	731,051	437.030	1,168,081
Apr.30	6347	774,005	437,666	1,211,671
Fri.1	6380	717,000	437,787	1,154,787
Total	44,489	5,147,481	2,978,553	8,126,034



(a) contract transaction numbers



(b) contract vs transfer transactions

Fig. 1. Contract transactions.

We sorted the collected 5,147,481 contract transactions in the descending order and extracted 90,920 different smart contract address. We discover that the ranked first contract address was triggered 491,969 times and maintained 9.56% of all contract transactions and the top 25 contract of 0.02% of the triggered contracts maintain 54.67% of the contracts transaction (as shown in table II).

We divided the top 25th contracts into 10 types, nft, stablecoin, dex, defi, mateUniversise, wallet, layer2, dao (as shown in Table III) We list the smart contract program in each type and the percentage of the all contracts transactions in this type comparing to all contract transactions.

nft is time-sensitive

We present in Figure I-A the number of contract transactions among different smart contracts, where the top 25th contracts account for the 54.67% of the contract transactions ,the remaining 90,895 contracts take the 45.33% contract transactions. According to the contracts transactions, we discover that there are 52,698 smart contracts were only triggered once and 88,186 smart contracts were used less than 100 times and only 50 smart contracts were triggered more than 10,000 times(as shown in Figure I-A).

We collected the pending transactions from ethscan and compared their gas price to the block transactions that are linked to the chain at the same time. The new transaction fee mechanism after EIP1559 calculates gas price by two bidding parameters instead of the firs -price auctions,thus we use the prior value to compare pending and block transactions belong to the eip1559 transaction fee mechanism.

$$GasPrice = \min[MaxFeePerGas, BaseFeePerGas + MaxPriorFeePerGas] \quad (1)$$

Deduct the $GasPrice * Gas$ fee from the sender account's

TABLE II
SMART TRANSACTIONS.

Tag(contract address)	Count	Percentage
USDT	491,969	9.56%
OpenSea: Wyvern Exchange v2	475,834	9.24%
Uniswap V3: Router 2	316,903	6.16%
ENS: ETH Registrar Controller	240,506	4.67%
Apecoin: APE Token NFT	212,445	4.13%
USDC	185,392	3.60%
Wrapped Ether	164,300	3.19%
The Otherside: OTHR Token	106,445	2.07%
Metamask: Swap Router	80,496	1.56%
Uniswap V2: Router 2	74,117	1.43%
StrongBlock: Service	53,841	1.05%
KOK PLAY: KOK Token	47,292	0.92%
VeeFriends Series 2: VF2 Token NFT	46,710	0.91%
Shiba Inu: SHIB Token	35,615	0.69%
1inch v4: Router	35,050	0.69%
MEV Bot	33,082	0.64%
OpenSea: Registry	31,648	0.61%
LooksRare: Exchange	24,826	0.48%
Polygon (Matic): Bridge	24,135	0.47%
Polygon (Matic): Matic Token	23,798	0.46 %
MEV Bot	23,078	0.45%
SushiSwap: Router	22,849	0.44%
Imaginary Ones: IO Token	21,780	0.42 %
MEV Bot	21,591	0,42%
Dai Stablecoin	20,447	0.40%

TABLE III
SMART TRANSACTIONS.

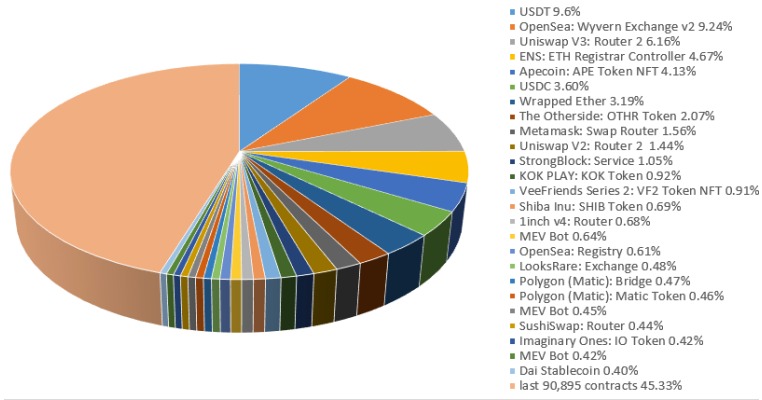
Type	Smart contract	Percentage
nft	OpenSea: Wyvern Exchange v2, ENS: ETH Registrar Controller, Apecoin: APE Token, VeeFriends Series 2: VF2 Token, OpenSea: Registry, LooksRare: Exchange, Imaginary Ones: IO Token	20.47%
stabecoin	USDT, USDC, Dai Stablecoin	13.56%
dex	Uniswap V3: Router 2, Wrapped Ether, Uniswap V2: Router 2 , 1inch v4: Router	11.47%
defi	StrongBlock: Service, Shiba Inu: SHIB Token, SushiSwap: Router	2.18%
mateuniverse	The Otherside: OTHR Token	2.07%
wallet	Metamask: Swap Router	1.56%
mev	MEV Bot	1.51%
layer2	Polygon (Matic): Bridge, Polygon (Matic): Matic Token	0.93%
dao	KOK PLAY: KOK Token	0.92%

balance before transactions execution in case of the denial of service and execution termination.

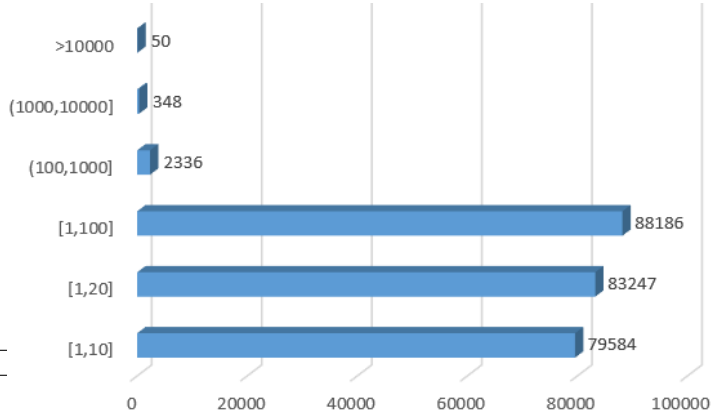
TABLE IV
NOTATIONS RELATED TO EIP1559.

Notation	Description
BaseFeePerGas	The minimum GasFeePerGas needed for a inclusion in a block, it is adjusted based on the total used gas in the previous block.
MaxFeePerGas	The maximum gas that a user is willing to pay for a transaction.
MaxProriFeePerGas	The maximum gas that a user is willing to pay the miner for execution.
GasTarget	EIP1559 changes the block size, the total gas limit is doubled from 15,000 to 30,000, and the gas target is set at 15,000 that are expected to use in a block.

Here, the base fee is adjusted dynamically through the previous block gas as follows n refers to the block height,



(a) Top 25 contract transactions percentage.



(b) Contract number

Fig. 2. Smart contracts used in tranactions.

$BaseFee_n$ and $GasUsed_n$ refers to the base fee and gas used in the block n.

$$BaseFee_{n+1} = BaseFee_n(1 + \frac{1}{8} \times \frac{GasUsed_n - GasTarget}{GasTarget}) \quad (2)$$

We collected 8,902 pending transactions and 502 block transactions from the ethscan mempool. We discover that even there is a higher prior gas fee in the pending pool, the miner might choose the lower one, such as there is a lot transactions with 0.1 GWei prior fee from users to $0x4f6742badb049791cd9a37ea913f2bac38d01279$ in the block 14,786,429, thus we make the assumption that one reason of the network congestion is that the miner might choose the cooperating user's transactions as a first priority even there has more higher gas in the pending pool.

We collected 26,844 blocks and discovered that there are 64 different miners who blocked successfully, and the top miner owns 27.76% blocks with sorting the miner blocks in a descending order(as shown in Figure 3).

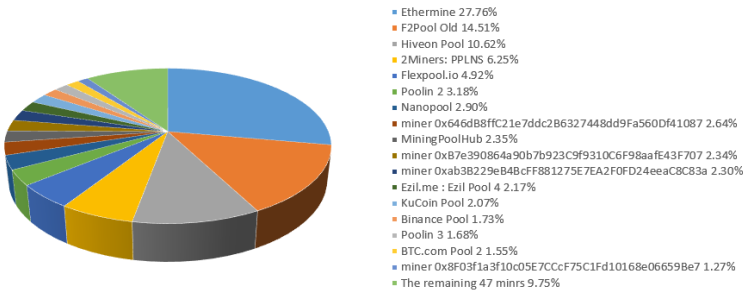


Fig. 3. Miner distribution.