Decentralized Reputation System

Senior Project Poster Day 2015
Department of Computer and Information Science
University of Pennsylvania

Abstract

We are building an open-source, decentralized reputation system with an API for nascent P2P transaction networks. No reputation system of this kind currently exists while others are proprietary and platform dependent. We believe that this will make P2P markets more trustworthy and efficient.

Contribution

We are creating a new reputation algorithm and implementing it as a smart contract.

Net Flow Convergence Algorithm

Observation on closed network: net flow is 0

\[ N_G(I, O) = \left( \sum_{i=1}^{n} v_i(i) \right) - \left( \sum_{o=0}^{m} v_o(o) \right) = 0 \]

where \( v(e) \) is weight of edge \( e \)

Convergence rate is defined as:

\[ \lim_{i \to n} N_G(I_i, O_i) \]

\[ \lim_{i \to n} \sum_{i=0}^{n} \frac{N_G(I_{i+1}, O_{i+1}) - N_G(I_i, O_i)}{|I_{i+1} - I_i| + |O_{i+1} - O_i|} \]

Net flow = net inflow – net outflow

Net flow convergence = the rate net flow converges to 0

Accuracy of Net Flow Convergence

<table>
<thead>
<tr>
<th>Shared Similarity (&lt;10)</th>
<th>Net Flow Convergence</th>
</tr>
</thead>
<tbody>
<tr>
<td>Eigenvector Centrality</td>
<td>&gt;95%</td>
</tr>
<tr>
<td>Degree Centrality</td>
<td>&gt;82%</td>
</tr>
</tbody>
</table>

Historical vs. Real-time Data

Historical data for backtesting

Real-time data for implementation

Smart Contract Architecture

Reputation Application Layer
(graph_contract, netflow_contract, transaction_contract)

Smart Contract Layer in Serpent
(./counterpartyd OR ./pyeth)

Transaction Layer
(./bitcoind)

User Interface

<table>
<thead>
<tr>
<th>Method</th>
<th>Return Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>transaction_contract</td>
<td>String</td>
<td>Records the outcome of a transaction. It is a distributed ledger on the blockchains. It contains transaction data as well as transaction metadata.</td>
</tr>
<tr>
<td>netflow_contract</td>
<td>Array of numbers</td>
<td>Calculates the reputation of each node in the P2P network.</td>
</tr>
<tr>
<td>graph_contract</td>
<td>Array of numbers</td>
<td>Calculates the reputation of each node in the P2P network.</td>
</tr>
</tbody>
</table>

$ serpent compile transaction_contract.se
6100ad8061000...

$ python
>>> from pyethereun import tester as t
>>> s = t.state()
>>> c = s.abi_contract('transaction_contract.se')
>>> c.register('12afadfa', 'Matt', 'Defu', 11, 1) [1]