A Petri Nets Model for Blockchain Analysis

Seminar for the course of CMCS

Luca Lombardo

Based on the work of: Andrea Pinna, Roberto Tonelli, Matteo Orrú, Michele Marchesi

Structure of the Presentation

A Petri Net Model for Blockchain Analysis

Key Contributions

- **Addresses Petri Net (APN):** Maps Bitcoin addresses to *places* and transactions to *transitions* in a P/T Net
- **Entities Petri Net (EPN):** Groups addresses into owner entities using a clustering algorithm

Advantages of PN Formalism

- Algebraic structure enables formal analysis of transaction patterns
- Native representation of blockchain architecture through PN semantics
- Enables dynamic simulations for network behavior forecasting

Model Features

- Preserves transaction topology
- Supports address clustering
- Enables multi-scale analysis

Paper Structure

- System overview
- PN model definition
- APN/EPN construction
- Blockchain analysis results

Bitcoin Blockchain: Architecture & Transactions

Overview

- **Distributed Public Ledger:** A global database that stores every validated Bitcoin transaction.
- **Chain of Blocks:** Transactions are grouped into blocks that form an ordered sequence (the Blockchain).

Transaction Structure

- **Inputs:** Reference previous unspent outputs (UTXOs) from earlier transactions.
- **Outputs:** List one or more addresses (e.g., strings starting with "1" or "4") along with their associated values.
- **UTXO Model:** An address's balance equals the sum of its unspent outputs.

User Interaction

- **Digital Wallets:** Bitcoin clients store public/private key pairs that manage one or more addresses.
- **Anonymity:** Only addresses are recorded; no personal identity information is required.

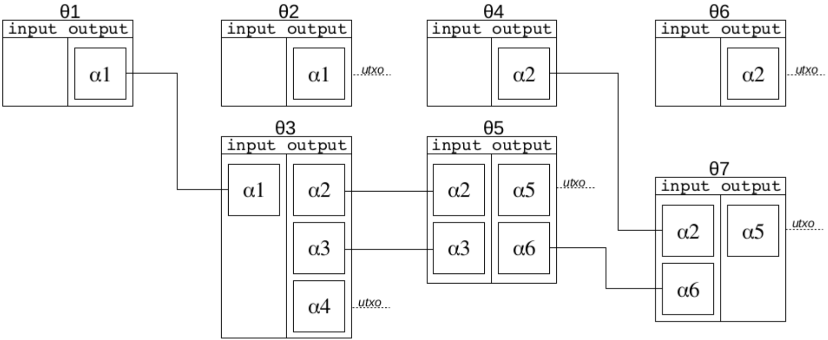


Figure: Simplified transaction schema

Bitcoin Blockchain: Transaction Verification & Mining

Transaction Validation

- **Broadcast & Pending:** Transactions are shared in the peer-to-peer network and wait to be validated.
- **Validation Rules:**
  - Each input must reference a valid UTXO.
  - The total value of outputs must not exceed that of inputs.

Mining Process

- **Proof-of-Work:** Miners solve a computational puzzle by finding a hash.
- **Block Contents:** A block includes the selected transactions, the previous block's hash, its block height, and miner information.
- **Incentives:** The first miner to solve the puzzle receives a block reward

Petri Nets: Basic Concepts

Overview

A **Petri Net** is a formal model for distributed systems built on a bipartite graph consisting of:

- **Places:** Represent conditions, resources, or system states.
- **Transitions:** Represent events that change these states.

Graph Structure

- **Bipartite Graph:** Only two types of nodes are allowed, and connections occur only between nodes of different types.
- **Arcs:** Directed arcs link places and transitions:
  - *Pre-arcs:* Arcs from places to transitions (inputs).
  - *Post-arcs:* Arcs from transitions to places (outputs).

Petri Nets: Algebraic Formalism & Markings

Algebraic Description

A Petri Net is formally defined as a quadruple:

$$N = (P, T, Pre, Post)$$

where:

- $P = \{p_1, p_2, \ldots, p_m\}$ is the set of places.
- $T = \{t_1, t_2, \ldots, t_n\}$ is the set of transitions.
- $Pre : P \times T \to \mathbb{N}$ is the *pre-incidence* function.
- $Post : P \times T \to \mathbb{N}$ is the *post-incidence* function.

These incidence functions are typically represented as $m \times n$ matrices.

Petri Nets: Algebraic Formalism & Markings

Markings & Firing Rule

- A **marking** M is a vector assigning tokens to places, thus representing the system state.
- **Firing:** When a transition fires, it:
  1. Consumes tokens from its pre-connected places.
  2. Produces tokens in its post-connected places.
- The complete system is denoted as $\langle N, \mathbf{M}_0 \rangle$, where $\mathbf{M}_0$ is the initial marking.
- In our work on Blockchain analysis, we focus on the net structure, without defining a specific marking.

Addresses Petri Net: Overview and Definitions

Motivation

- Blockchain transactions move bitcoins between addresses.
- The inherent bipartite structure (addresses and transactions) suggests a natural mapping to a Petri Net.

Definitions

- $\mathcal{A} = \{\alpha_1, \alpha_2, \ldots, \alpha_m\}$: Finite set of addresses (inputs/outputs).
- $\Theta = \{\theta_1, \theta_2, \ldots, \theta_n\}$: Set of validated transactions.

Addresses Petri Net Structure

- Define $N_\alpha = (P_\alpha, T, \mathbf{PreA}, \mathbf{PostA})$ where:
  - $P_\alpha = \{p\alpha_1, p\alpha_2, \ldots, p\alpha_m\}$ associates one place per address.
  - $T = \{t_1, t_2, \ldots, t_n\}$ associates one transition per transaction.
  - **PreA** and **PostA** are the pre- and post-incidence matrices.
- These sets are built by scanning the Blockchain for new addresses and transactions.

Addresses Petri Net: Constructing the Incidence Matrices

Transaction Representation

- Each transaction θ is split into:
  - $\textbf{In}(\theta) \subseteq \mathcal{A}$: Set of input addresses.
  - $\textbf{Out}(\theta) \subseteq \mathcal{A}$: Set of output addresses.
- The corresponding transition t represents θ in the Petri Net.

Matrix Construction

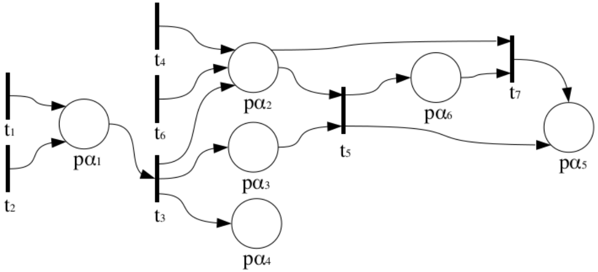
- For every $\alpha \in \textbf{In}(\theta)$:
  - Add a **pre-arc** from place $p\alpha$ to transition t.
  - Set $\textbf{PreA}(p\alpha, t) = 1$.
- For every $\alpha \in \textbf{Out}(\theta)$:
  - Add a **post-arc** from transition t to place $p\alpha$.
  - Set $\textbf{PostA}(p\alpha, t) = 1$.
- The matrices **PreA** and **PostA** are of dimension $m \times n$.

Addresses Petri Net: Example and Analysis

Example Overview

- Consider a simplified set of transactions (e.g. as in the previous slide)
- The Addresses Petri Net has:
  - 6 places ($P_\alpha = \{p\alpha_1, \dots, p\alpha_6\}$).
  - 7 transitions ($T = \{t_1, \dots, t_7\}$).

Below a graphical representation of an addresses Petri Net equivalent to the simplified transaction chains.



Pre-incidence matrix of the simplified transaction

$$\mathbf{PreA}= \begin{array}{cc} \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} & \begin{array}{c} p\alpha_1 \\ p\alpha_2 \\ p\alpha_3 \\ p\alpha_4 \\ p\alpha_5 \\ p\alpha_6 \end{array} \\ \begin{array}{ccccccc} t_1 & t_2 & t_3 & t_4 & t_5 & t_6 & t_7 \end{array} & \end{array}$$

Figure: Pre-incidence matrix of the Petri net for the example of the simplified transaction schema.

Post-incidence matrix of the simplified transaction

$$\mathbf{PostA} = \begin{array}{c} \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \begin{array}{c} p\alpha_1 \\ p\alpha_2 \\ p\alpha_3 \\ p\alpha_4 \\ p\alpha_5 \\ p\alpha_6 \end{array} \\ \begin{array}{ccccccc} t_1 & t_2 & t_3 & t_4 & t_5 & t_6 & t_7 \end{array} \end{array}$$

Figure: Post-incidence matrix of the Petri net for the example of the simplified transaction schema.

Address Petri Nets: Incidence Matrices and Analysis

Incidence Matrices and Analysis

- **Pre-incidence matrix PreA:** Captures the number of times an address appears as an input.
- **Post-incidence matrix PostA:** Captures the outputs associated with each transaction.
- By computing the difference ($\mathbf{PostA} - \mathbf{PreA}$) for each row, one can:
  - Determine the number of UTXOs per address.
  - Infer whether an address balance is null (zero tokens).
- Transactions sharing identical input and output sets may be merged into one transition with a firing clock, aiding dynamic analyses.

Entities Petri Net: Introduction

Motivation

- Bitcoin users often control multiple addresses to manage exchanges and preserve anonymity.
- We define an **entity** as the person, organization, or group that controls a set of addresses.
- **Key Property:** All addresses appearing in the input of a single transaction must belong to the same entity (since transferring funds requires control over all associated private keys).

Why Group Addresses?

- Enables high-level analysis of the Blockchain.
- Reduces complexity by clustering addresses that are likely controlled by the same user.
- Provides a more natural mapping to real-world economic actors.

From Addresses to Entities: Definitions & Mapping

Mapping Addresses to Entities

- Let $\mathcal{A} = \{\alpha_1, \alpha_2, \ldots, \alpha_m\}$ be the set of addresses.
- Let $\Theta = \{\theta_1, \theta_2, \ldots, \theta_n\}$ be the set of transactions.
- In the Addresses Petri Net $N_\alpha = (P_\alpha, T, \mathbf{PreA}, \mathbf{PostA})$, each place $p\alpha$ corresponds to an address α.

Defining an Entity

- An **entity** ϵ is a set of addresses: $\epsilon \subseteq \mathcal{A}$.
- Denote the set of entities as $E = \{\epsilon_1, \epsilon_2, \ldots, \epsilon_k\}$.
- In the **Entities Petri Net** $N_\epsilon = (P_\epsilon, T, \mathbf{PreE}, \mathbf{PostE})$, each place $p\epsilon$ represents one entity.

Algorithm 1 Compute Entities from the Addresses Petri Net

$T^* \leftarrow T$ ▷ Set of unexplored transactions
$E \leftarrow \emptyset$ ▷ Set of entities
while $T^* \neq \emptyset$ **do**
  Select and remove a transaction t from T^*
  $e \leftarrow \emptyset$ ▷ Initialize new entity
  for all p_i such that $\mathbf{PreA}(p_i, t) = 1$ **do**
    $e \leftarrow e \cup \{p_i\}$
  end for
  $e^* \leftarrow e$ ▷ Set of unexplored places within e
  while $e^* \neq \emptyset$ **do**
    Select a place p from e^*
    for all transactions t' with $\mathbf{PreA}(p, t') = 1$ **do**
      for all p_h such that $\mathbf{PreA}(p_h, t') = 1$ **do**
        $e \leftarrow e \cup \{p_h\}$, $e^* \leftarrow e^* \cup \{p_h\}$
      end for
      Remove t' from T^*
    end for
    Remove p from e^*
  end while
  $E \leftarrow E \cup \{e\}$
end while

Entities Petri Net: Incidence Matrices and Aggregation

Constructing the Entities Petri Net

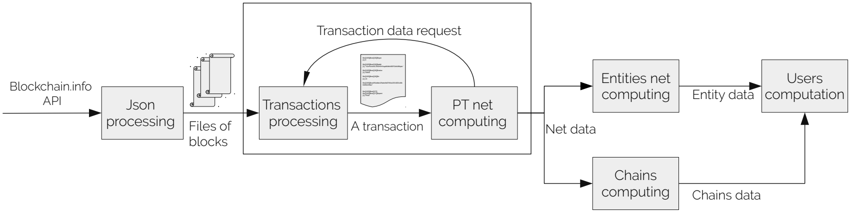
- Each computed entity $e \in E$ is represented by a unique place $p\epsilon$ in the Entities Petri Net $N_\epsilon = (P_\epsilon, T, \mathbf{PreE}, \mathbf{PostE})$.
- The set of transitions T remains the same as in the Addresses Petri Net.

Aggregating Incidence Matrices

- For each entity e, identify all corresponding addresses $p\alpha \in e$ from the Addresses Petri Net.
- **PreE** and **PostE** are obtained by summing the rows in **PreA** and **PostA** for all places in e.
- This aggregation captures the cumulative input and output interactions of all addresses within the entity.

Data Acquisition and Processing Pipeline

- **Approach:** Downloaded formatted JSON blocks from *blockchain.info*.
- **Dataset:** Parsed the first 180,000 blocks (Jan 2009 – Mar 2012)
- **Implementation:** Data processing executed in R using RStudio IDE.
- **Performance:** Total processing time $\approx$ 250 hours (avg. 5 sec per block).



Addresses Petri Net: Global Statistics and CCDF Analysis

- **Addresses:** 3,730,480 distinct addresses.
- **Transactions:** 3,142,019 transactions (columns in **PreA**/**PostA**).
- **Arcs:** 4,575,888 pre-arcs and 7,352,494 post-arcs.
- The number of nonzero elements in a row of **PreA** (or **PostA**) indicates the number of input (or output) transactions for that address.
- CCDF plots reveal a power-law distribution: many addresses with few transactions and a few addresses with very high activity.

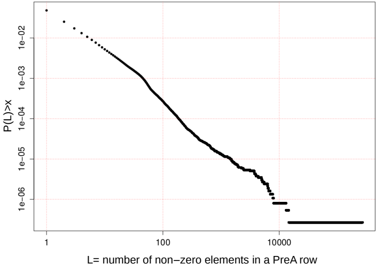


Figure: CCDF of L for **PreA**.

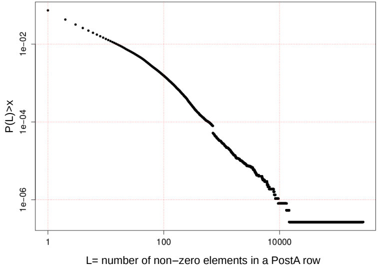


Figure: CCDF of L for **PostA**.

Addresses Petri Net: Most Used and Imbalanced Addresses

Usage Ranking

The most used addresses are identified by summing the nonzero elements in both **PreA** and **PostA** rows.

Imbalanced Addresses

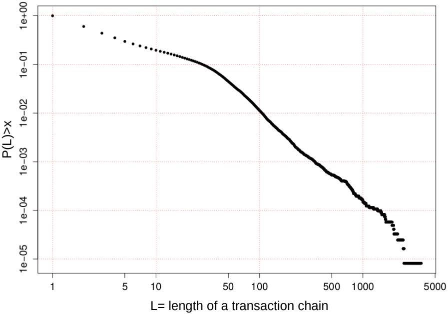
609,295 addresses show zero entries in **PreA** but at least one nonzero in **PostA**. These addresses have never been used to spend bitcoins but only to receive them. Table 1 lists the top 5 addresses by incoming transactions along with their current balances.

Address	L post	current balance BTC
15S1TFTosxrgZxkqJR2n1AFJ22ZJE2rTCk	3,853	120.85215349
1PtnGiNvhAKbuUQ6nZ7nF3CDKCKGfeMsCX	1,199	0
129FTwWoi5H5ujasMZ6M6VjJzBJfsXVQGw	1,138	0.78425567
1FN9kKsZA9XttrAwuDDgsXjs6CXUR2fzmt	1,111	0
1DYvtKtZ2Ay9vTjzjb9BiRauMgXdjRDaD	973	14.5601

Table: Summary of first 5 most imbalanced addresses

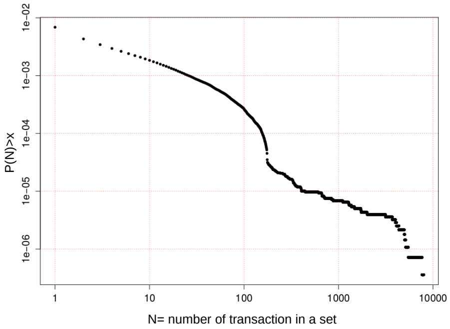
Addresses Petri Net: Disposable Address Chains

- **Disposable Addresses:** Defined as addresses used exactly twice (once to receive and once to send all bitcoins).
- Transactions involving disposable addresses typically have:
  - One pre-arc (single input) and two post-arcs (output to two distinct addresses).
- The model easily identifies these by analyzing **PreA** and **PostA**.
- 122,155 disposable address chains have been detected, involving 1,350,010 different addresses/transactions.



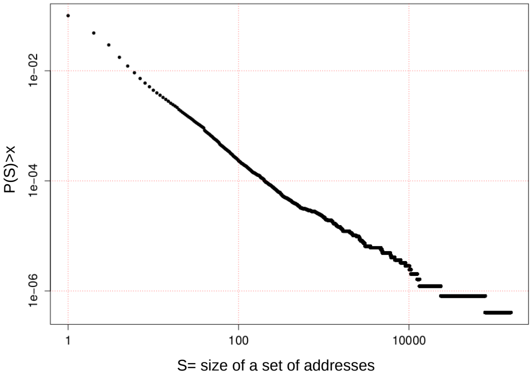
Addresses Petri Net: Repeated Transaction Patterns

- **Repeated Transfers:** Users sometimes execute transactions with identical sets of input and output addresses.
- In the Petri Net model, such repetitions appear as transitions with identical pre- and post-arc configurations.
- Approximately 11% of transactions are repetitions, reflecting steady bitcoin flows between fixed groups of addresses.
- The CCDF of grouped transaction sizes (i.e., the number of repetitions) illustrates this phenomenon.



Entities Petri Net: Address Distribution and Entity Control

- **Entities and Addresses:** 2,461,010 entities control 3,730,480 addresses.
- Distribution is highly non-uniform and follows a power-law pattern:
    - Many entities hold a single address.
    - Few entities control a large number of addresses and influence a significant portion of bitcoin transactions.



Entities Petri Net: Transaction Distributions for Entities

- **Transaction Involvement:** The number of non-zero elements in the rows of **PreE** and **PostE** indicates how many transactions an entity is involved in.
- **Power-law Distribution:** Transactions among entities show a power-law behavior for both input and output transactions.

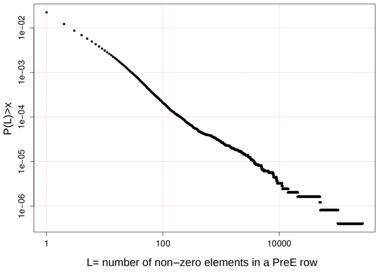


Figure: CCDF of the length L for **PreE**.

Figure: CCDF of the length L for **PostE**.

Entities Petri Net: Most Active Entities

- **Top Entities:** The 5 most active entities are identified by the sum of non-zero elements in both **PreE** and **PostE**.
- Their balances are calculated by summing the balances of all addresses belonging to each entity.
- **Tags:** Some entities are associated with known tags (e.g., deepbit.net, ilovethebtc).

Entity number	L pre	L post	size	tags
95237	270,204	275,398	2	deepbit.net
2	102,186	283,973	156,725	ilovethebtc
37	51,228	147,712	78,251	jmm5699
11	49,959	97,732	10,37	- unknow -
130	20,857	58,350	23,649	Instawallet

Table: Summary of first 5 most active entities

Entities Petri Net: Repeated Transaction Patterns

- **Repeated Transactions:** About 22.6% of transactions are repetitions of previous transactions between the same input and output entities.
- These repetitions indicate steady fluxes of bitcoins at the entity level.
- **Visualization:** The CCDF for grouped transaction sizes demonstrates this repetitive behavior.

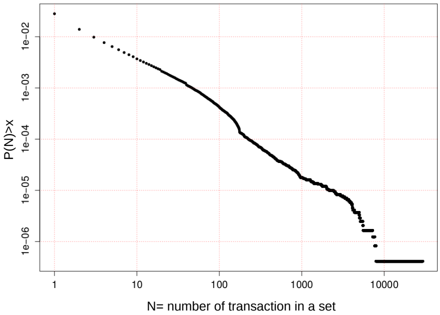


Figure: CCDF of the size L of grouped transaction sets for the Entities Petri Net.

Analyzing Bitcoin Users with Petri Nets

Petri Net Model

Clusters Bitcoin addresses into entities (246,660 owners control 1.5M addresses) and traces transaction chains (122,155 owners linked to 1.35M addresses).

User Classification

368,815 engaged owners use disposable addresses or multiple addresses (72.6% of transactions). 609,295 addresses are likely "deposit addresses" for engaged users. 255,045 addresses are owned by occasional users.

Case Studies

Most active input address (270k transactions): *DeepBit* mining pool (now defunct). Largest output entity (156k addresses): Tags include *ilovethebtc*, *mikeo*, and links to unreachable domains.

Insight

Strongly uneven transaction distribution (CCDFs) suggests miner pools use high-activity addresses for reward redistribution.

Advantages of Petri Net Formalism

Non-Deterministic Dynamics

Models UTXO-enabled transactions as independent events. Natively handles probabilistic block inclusion (miner fees, validation).

Simulation Power

Captures concurrent non-conflicting transactions (e.g., same-block validations). Enables statistical modeling of future states (e.g., miner pool growth predictions).

Sequential Analysis

Automatically identifies chains of disposable addresses (privacy-preserving flows).

Matrix-Driven Insights

Pre/Post matrices reveal never-spent addresses (609k output-only), single-use addresses, and transaction volumes.

State Equations

Enable probabilistic forecasting of Bitcoin fluxes between entities (exchanges, pools) using historical data.

Conclusions

Novel Petri Net Architecture

Processed first 180,000 blocks ($\approx$ 3.5 years of Bitcoin history).

Addresses $\rightarrow$ Places (2.7M+), Transactions $\rightarrow$ Transitions.

Pre/Post matrices capture all input-output relationships.

Key Discoveries

Universal power-law distributions were observed in transaction arcs (pre/post connections), disposable address chain lengths, and repeated input-output transaction clusters.

Entity Network Reconstruction: Built a secondary Petri Net of address owners through matrix analysis.

Computational Considerations

The current blockchain (480k+ blocks, 240M+ transactions) challenges full-scale analysis.

Flexible partial analysis allows investigation of specific address subsets or time windows.