Analyzing Celestia Blockchain

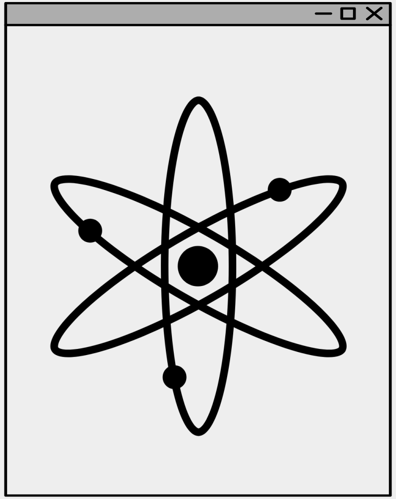
902388 - Gabriele Pilotto
885448 - Pietro Visconti

Problem Description

The system we will to model takes in consideration **Celestia**, a blockchain within the Cosmos ecosystem.

Since Cosmos and Celestia **operates with the same consensus algorithm, but with different times**, our work aims to:

- evaluate system's **throughput**
- determine **optimal timeouts** for:
    - **homogeneous** proposer scenario.
    - **non-homogeneous** proposer scenario.



01

Intro

Introducing Celestia and comparing it with Cosmos.

Introduction Celestia

What is Celestia?

A **modular blockchain** platform that separates core blockchain functions (consensus, data availability, execution) to improve scalability and flexibility.

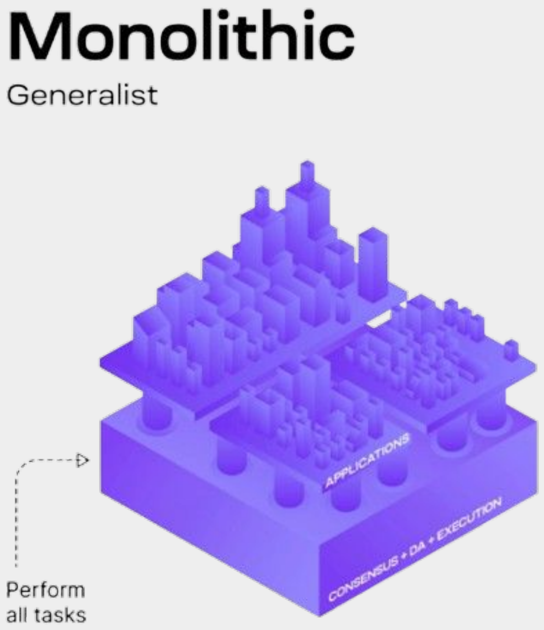
Its main feature, along with modularity, is **Data Availability** (DA).

What is a Monolithic Blockchain

One **single layer** combining:
- Consensus
- Data availability
- Execution

Challenges:
- **Limited scalability**: all nodes process every task
- High **resource requirements** for validation

Examples:
- Bitcoin
- Ethereum
- Cosmos chains.



What is a Modular Blockchain

Presents **decoupled layers**:

- Consensus & Data Availability layer
- Settlement layer
- Execution layer

Benefits:

- Scales by **separating responsibilities**
- **Flexible** for different use cases

Celestia is based on **modularity.**

Modular Blockchains: Why?

Scalability

By offloading resource-intensive tasks to separate layers.

Innovation

Specialized blockchains (rollups) are easier to build.

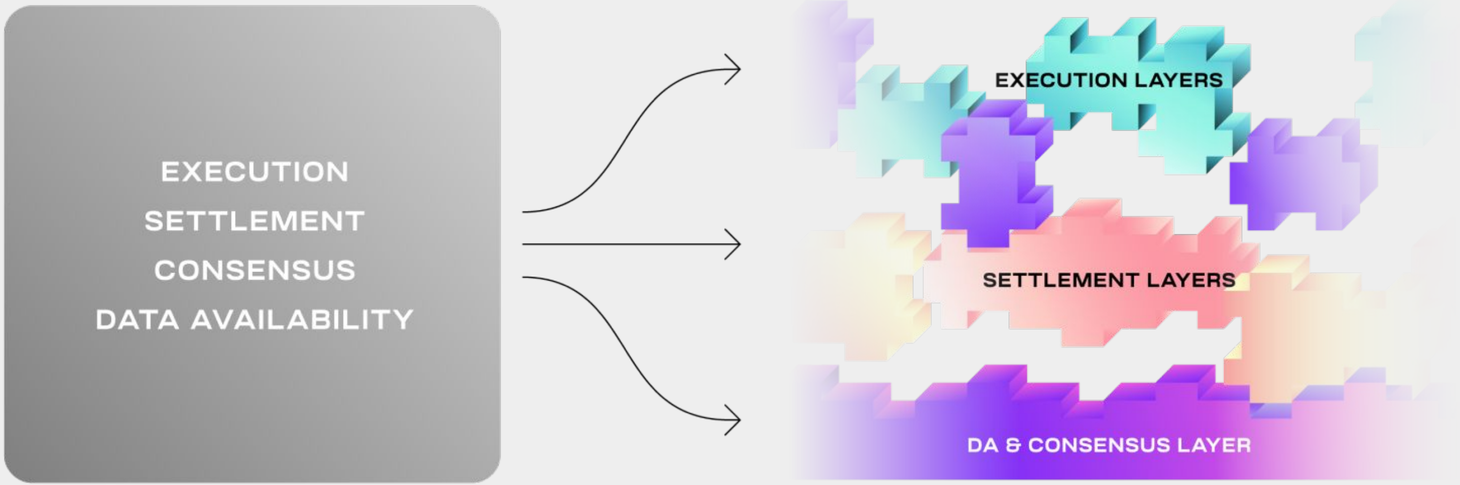
Accessibility

No need of high-hardware requirements: lightweight nodes are welcome.

Interoperability

Different blockchain can communicate without additional protocols.

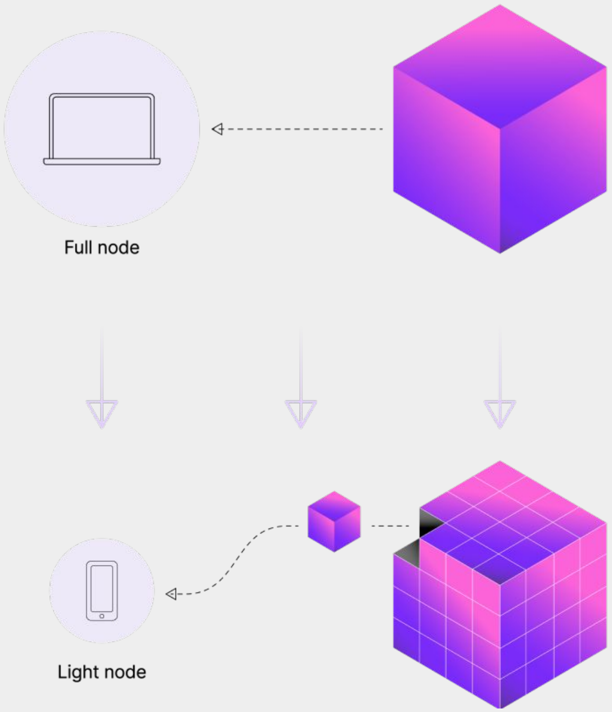
Monolithic vs Modular Blockchains



Data Availability (DA)

Definition

DA ensures all transaction data is published and accessible, and any node in the network is able to verify its validity.



Problems in monolithic chains:

- Nodes **must download and store** all transaction data.
- It **decreases scalability** as high hardware power is required.

Solution:

Celestia's **Data Availability Sampling**.

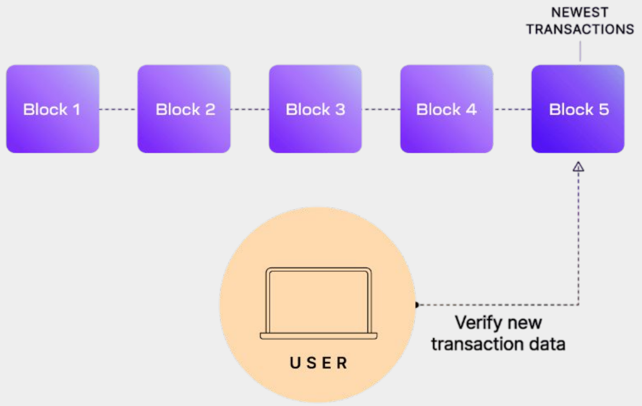
Data Availability Sampling (DAS)

How it works:

1. Breaks **data into chunks** for nodes to sample randomly.
2. Confirms availability **without storing** full data.

Main benefit:

it **supports lightweight nodes**, enhancing scalability and accessibility.



Celestia's Drawback

Compared to Cosmos, Celestia is way slower in terms of **block creation speed**. Today (03/12/2024), the block creation times are:

11.91

seconds for **Celestia**

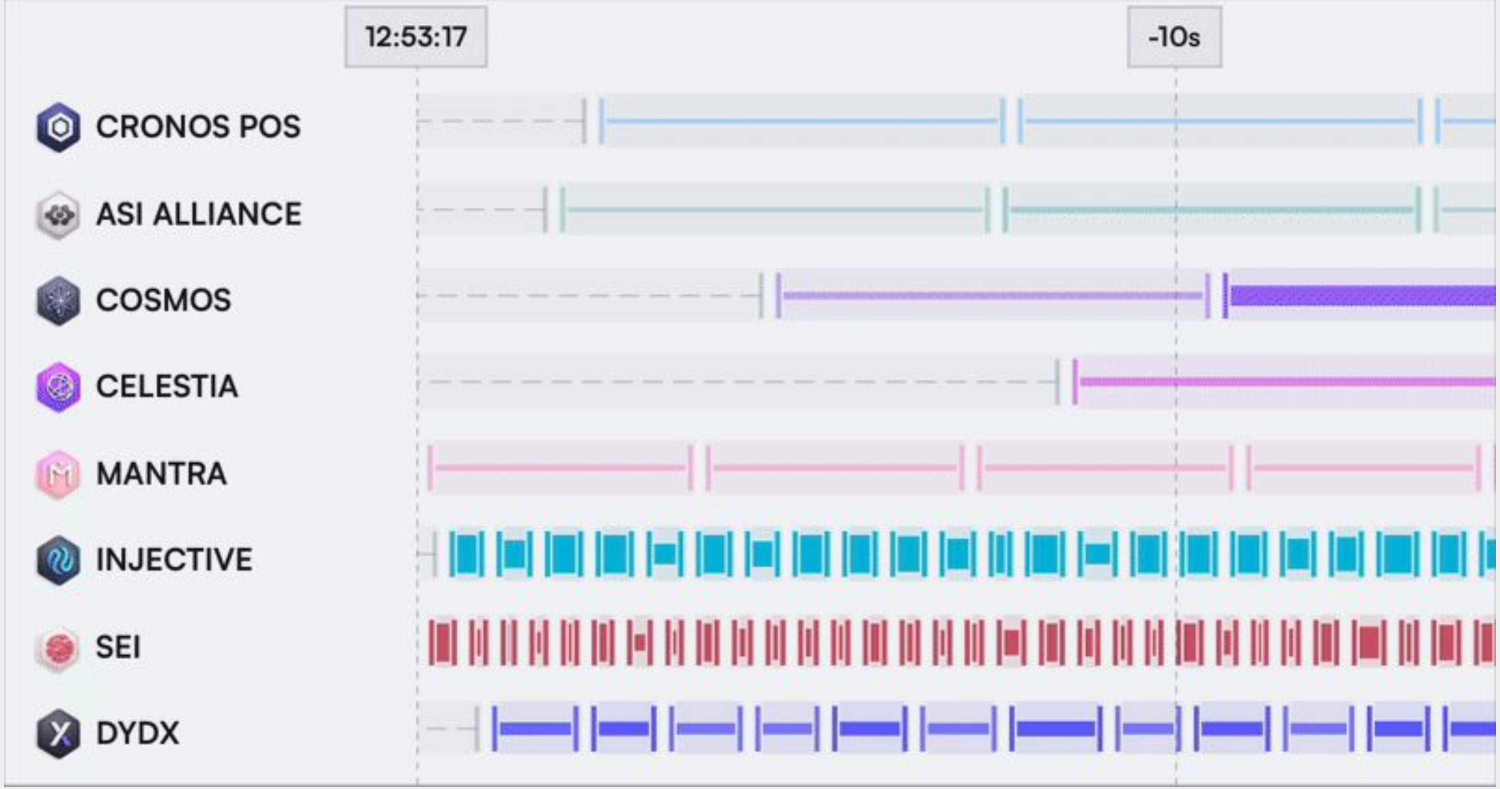
5.89

seconds for **Cosmos**

For this reason, **our PEPA model for Celestia** will be designed with **all parameters doubled compared to the Cosmos** model presented in the initial paper.

Block creation times are provided by mintscan.io

Block Speeds Comparison



02

System

Model, Components, CTMC, Infinitesimal Generator Matrix

2.1

System Analysis

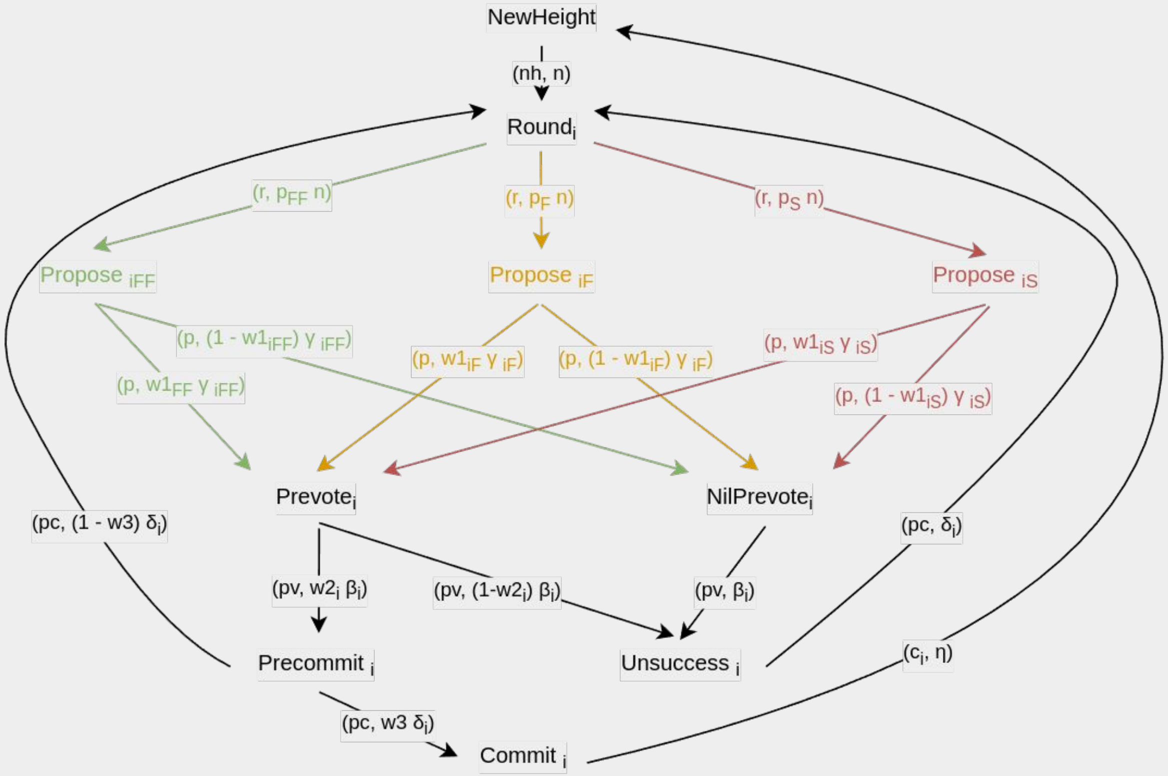
PEPA Model & Derivation Graph

System: PEPA Model

$NewHeight$	$\overset{def}{=}$	$(nh, n).Round_1$
$Round_i$	$\overset{def}{=}$	$(r, p_{FF}\,n).Propose_{i_{FF}} + (r, p_F\,n).Propose_{i_F} + (r, p_S\,n).Propose_{i_S}$
$Propose_{i_{FF}}$	$\overset{def}{=}$	$(p, w_{1_{i_{FF}}}\gamma_{i_{FF}}).Prevote_i + (p, (1 - w_{1_{i_{FF}}})\gamma_{i_{FF}}).NilPrevote_i$
$Propose_{i_F}$	$\overset{def}{=}$	$(p, w_{1_{i_F}}\gamma_{i_F}).Prevote_i + (p, (1 - w_{1_{i_F}})\gamma_{i_F}).NilPrevote_i$
$Propose_{i_S}$	$\overset{def}{=}$	$(p, w_{1_{i_S}}\gamma_{i_S}).Prevote_i + (p, (1 - w_{1_{i_S}})\gamma_{i_S}).NilPrevote_i$
$Prevote_i$	$\overset{def}{=}$	$(pv, w_{2_i}\beta_i).Precommit_i + (pv, (1 - w_{2_i})\beta_i).Unsuccess_i$
$NilPrevote_i$	$\overset{def}{=}$	$(pv, \beta_i).Unsuccess_i$
$Unsuccess_i$	$\overset{def}{=}$	$(pc, \delta_i).Round_j$
$Precommit_i$	$\overset{def}{=}$	$(pc, w_3\delta_i).Commit_i + (pc, (1 - w_3)\delta_i).Round_j$
$Commit_i$	$\overset{def}{=}$	$(c_i, \eta).NewHeight$

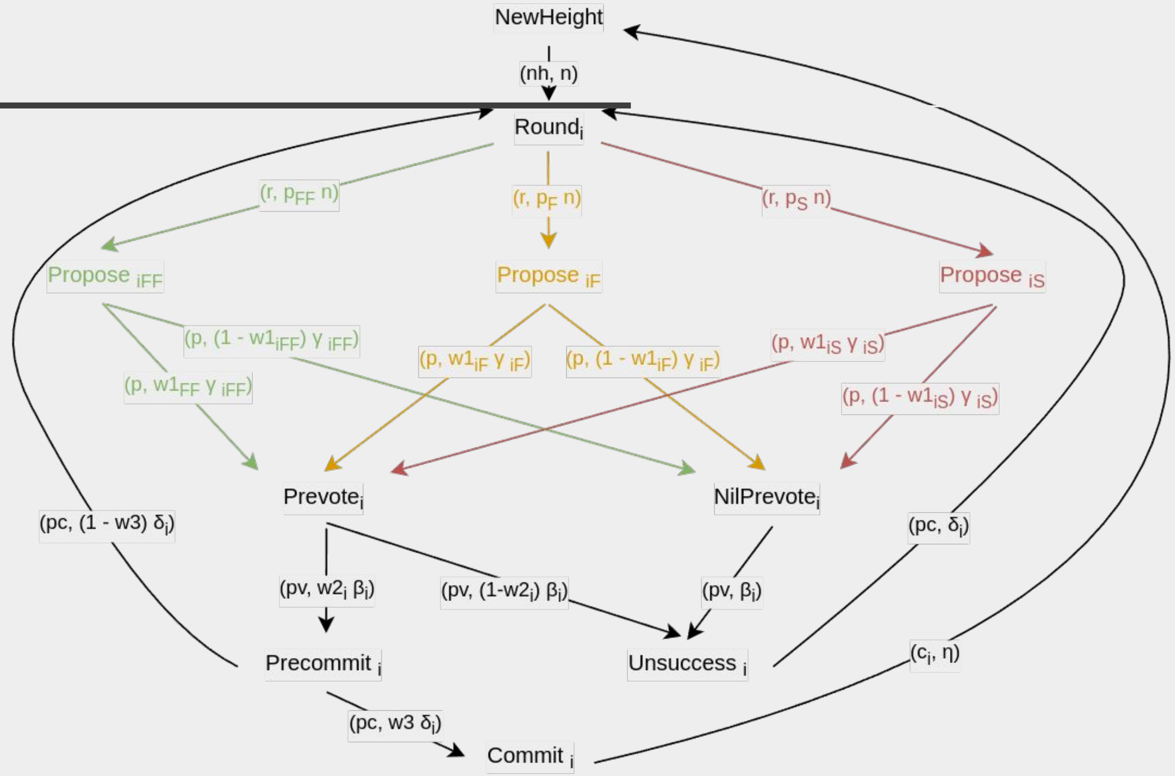
From now, we will **consider** only the **non-homogeneous** model

System: Derivation Graph

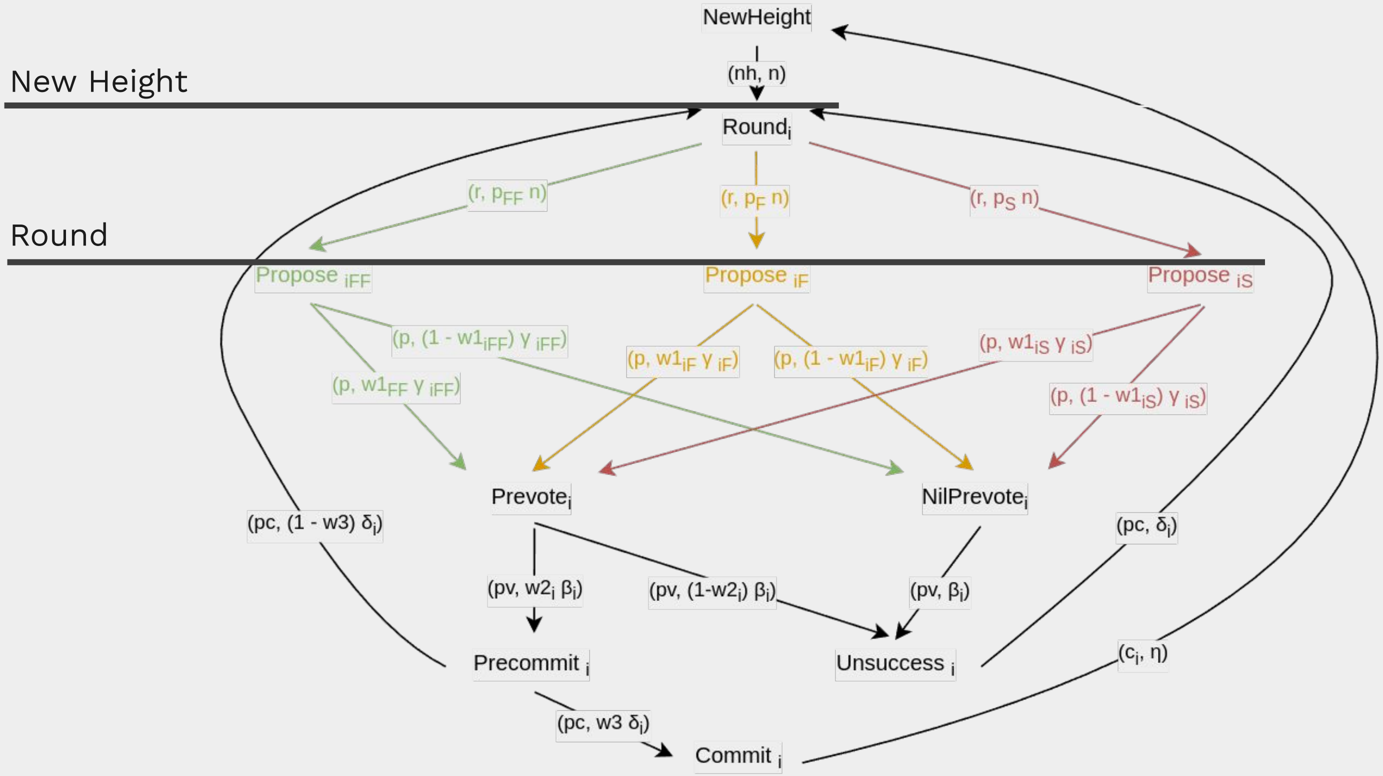


System: Derivation Graph

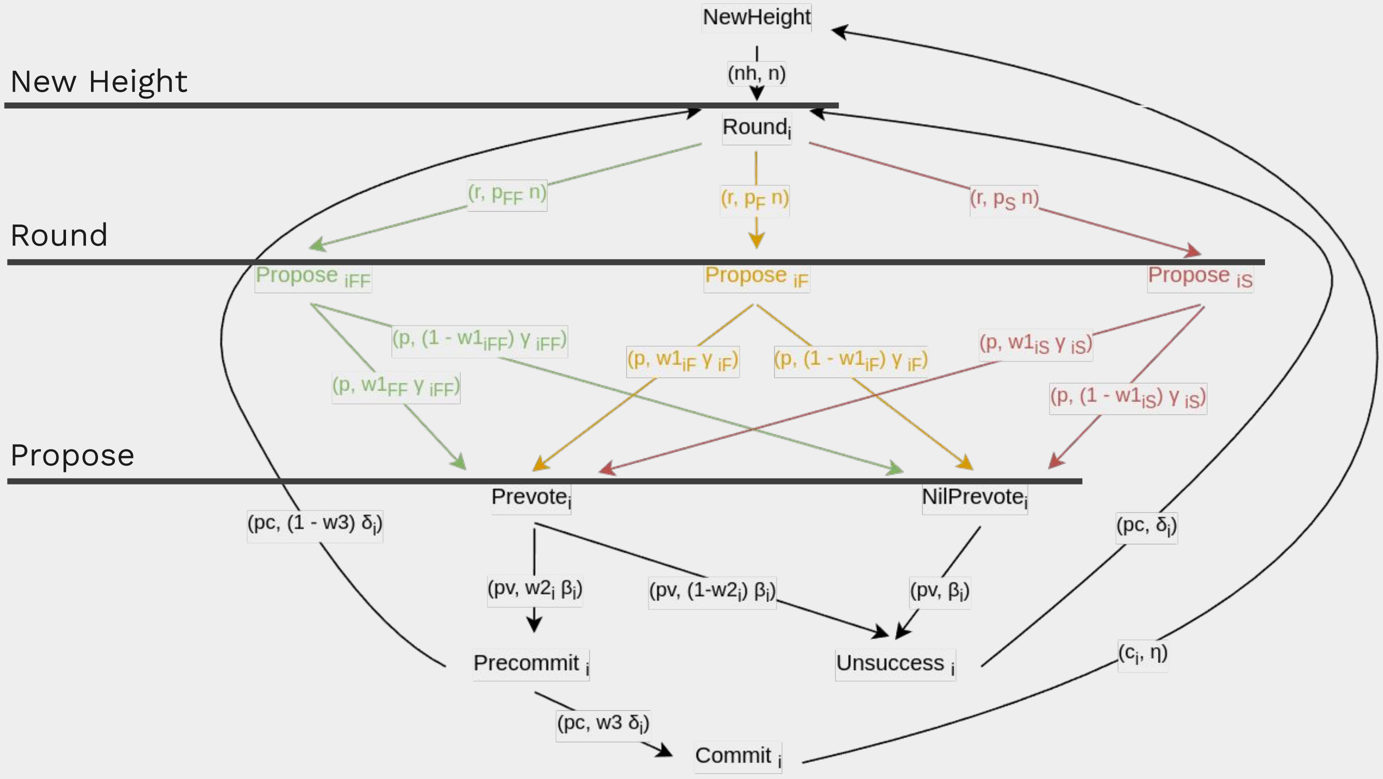
New Height



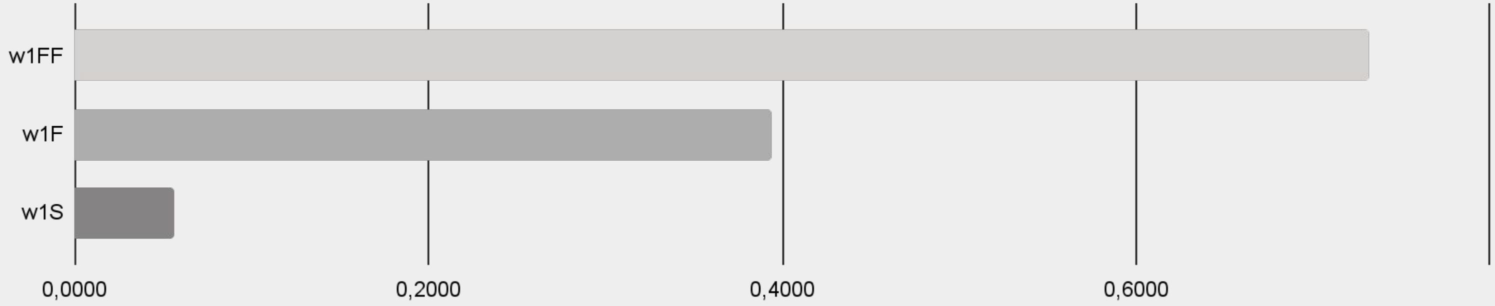
System: Derivation Graph



System: Derivation Graph



Propose $_{FF, F, S}$: Difference in Probabilities



$w1_{FF}$

0.7317 for the Super Fast category

$w1_{F}$

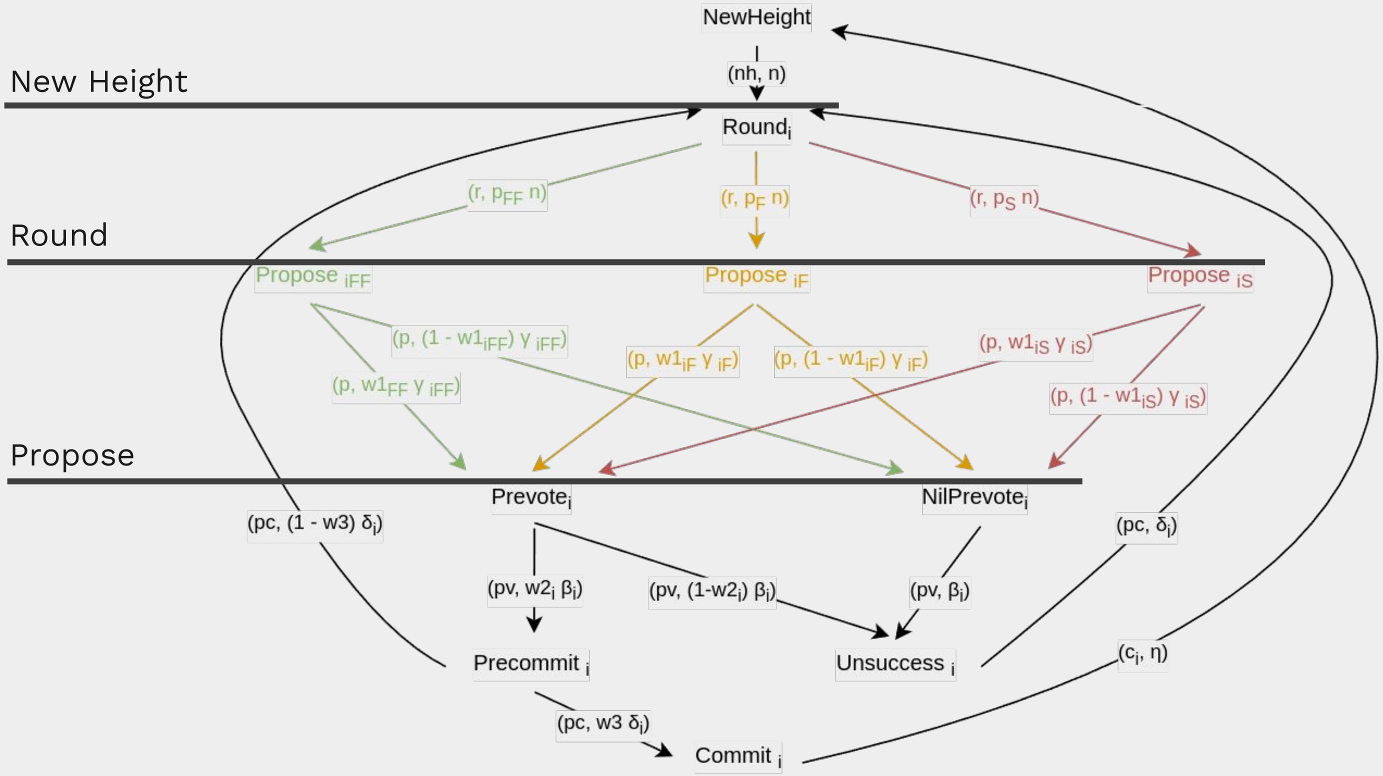
0.3935 for the Fast category

$w1_{S}$

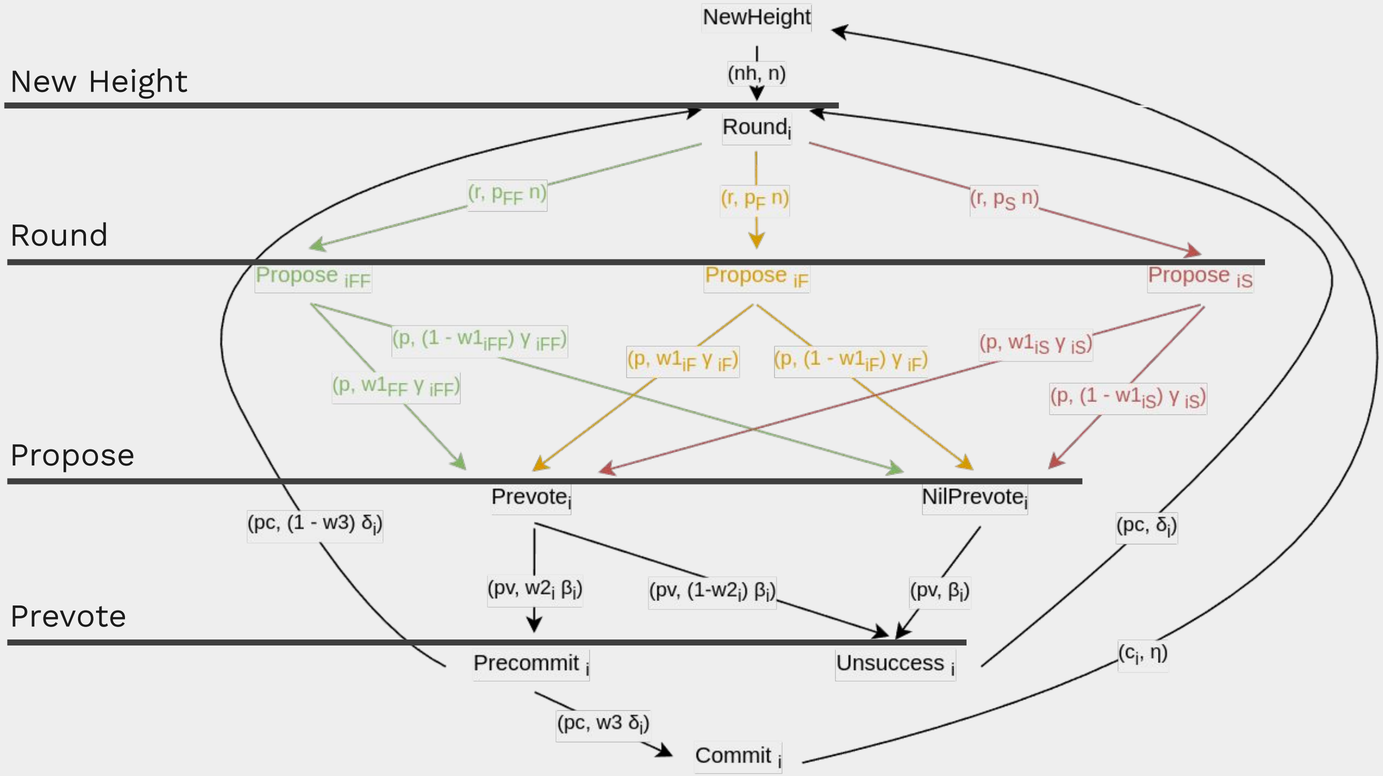
0.0566 for the Slow category

This is valid only for the *non-homogeneous* scenario.

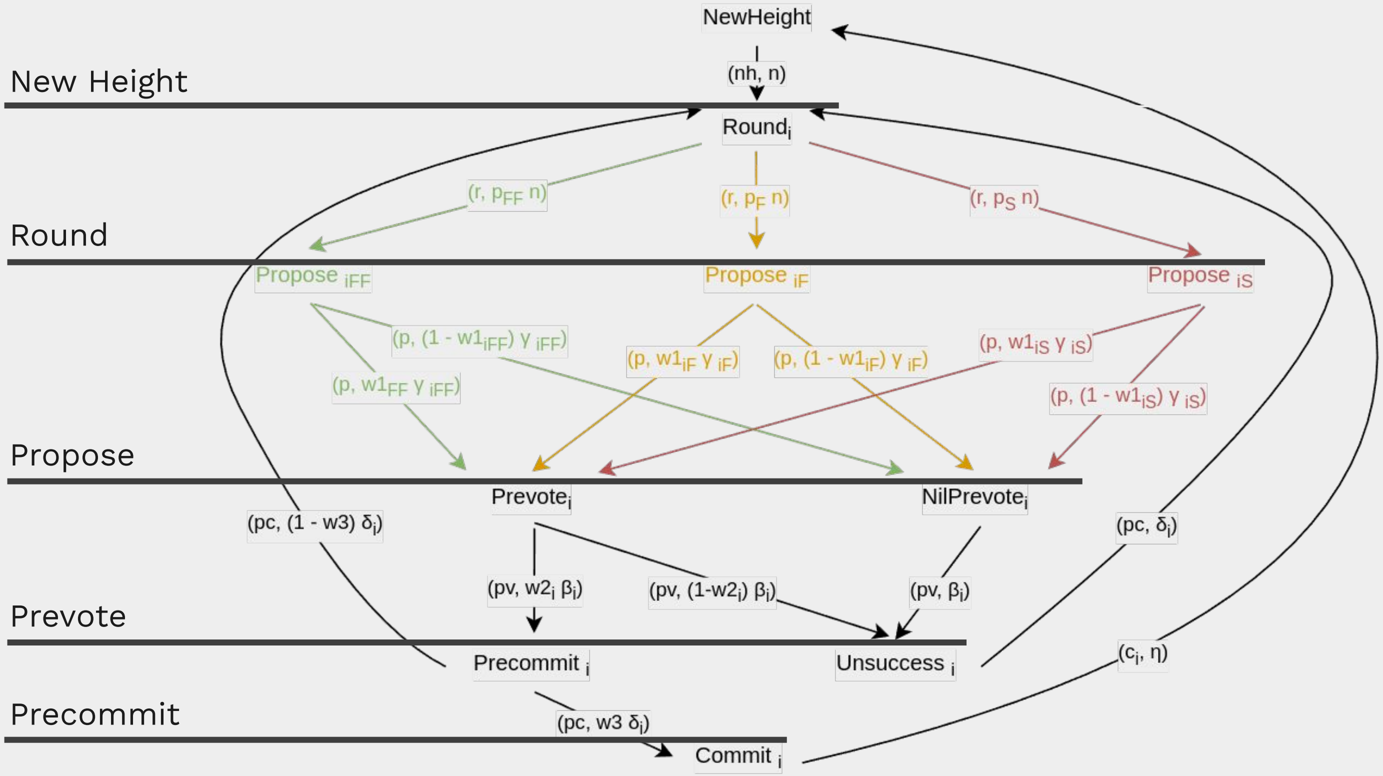
System: Derivation Graph



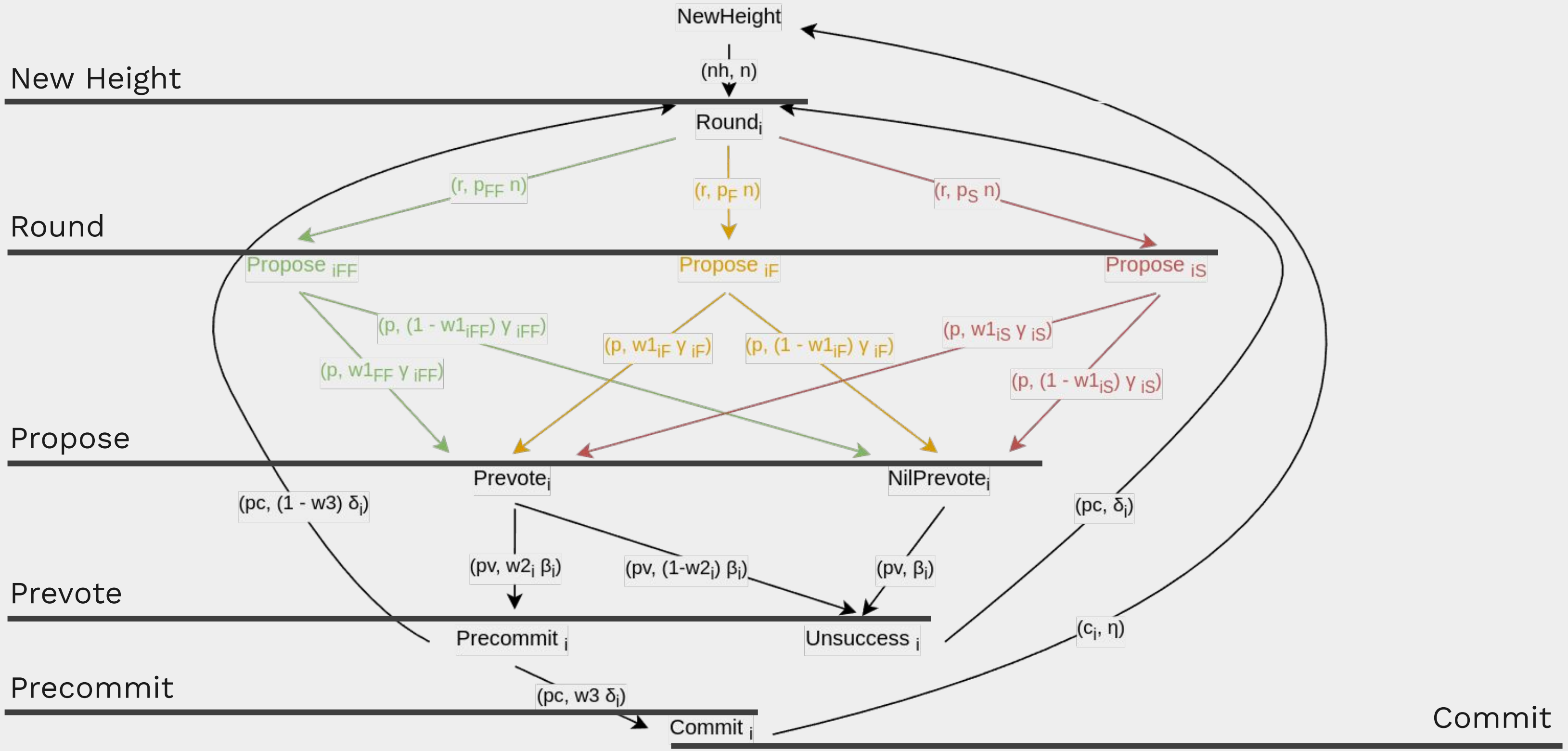
System: Derivation Graph



System: Derivation Graph



System: Derivation Graph

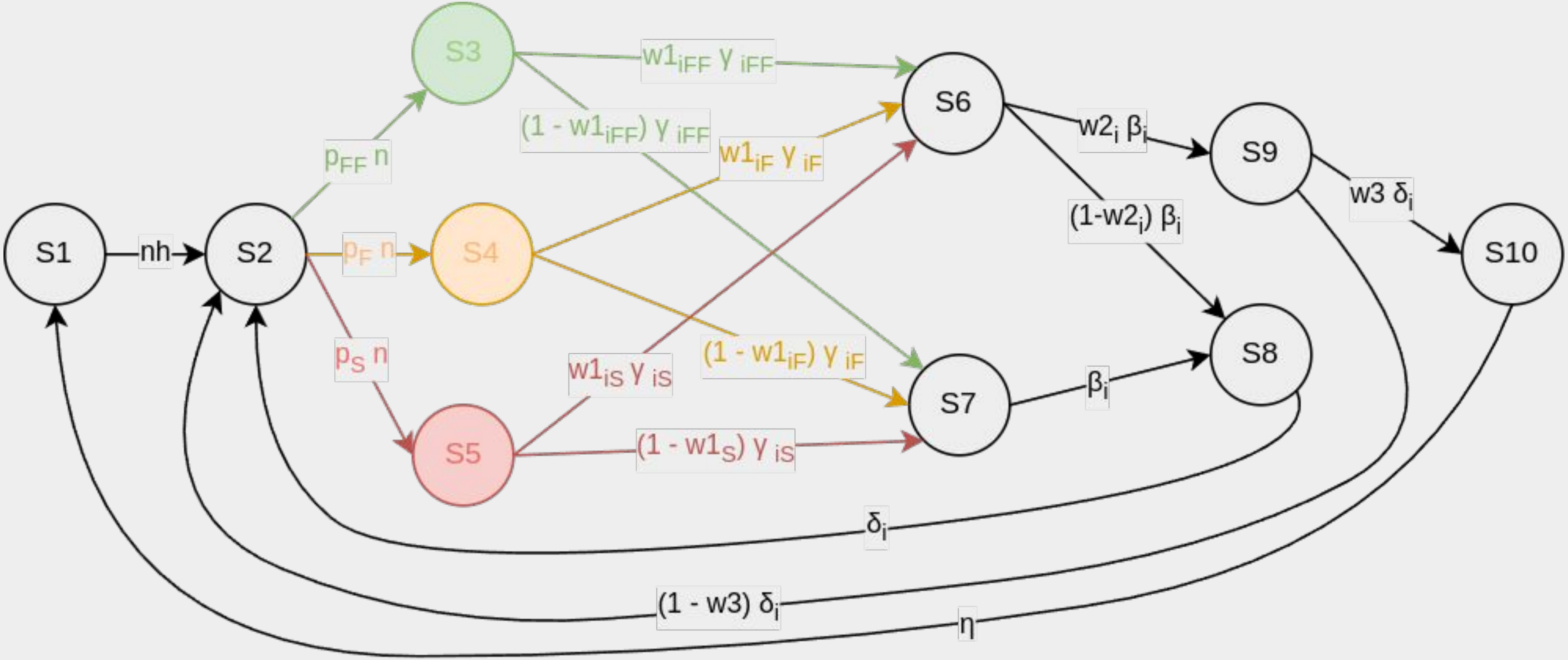


2.2

System Analysis

CTMC and Infinitesimal Generator Matrix

System: Continuous Time Markov Chain



System: Infinitesimal Generator Matrix

$$
\begin{bmatrix}
-nh & nh & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & -(p_{FF}n + p_F n + p_S n) & p_{FF}n & p_F n & p_S n & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & -(w1_{i_{FF}}\gamma_{i_{FF}} + (1-w1_{i_{FF}})\gamma_{i_{FF}}) & 0 & 0 & w1_{i_{FF}}\gamma_{i_{FF}} & (1-w1_{i_{FF}})\gamma_{i_{FF}} & 0 & 0 & 0 \\
0 & 0 & 0 & -(w1_{i_F}\gamma_{i_F} + (1-w1_{i_F})\gamma_{i_F}) & 0 & w1_{i_F}\gamma_{i_F} & (1-w1_{i_F})\gamma_{i_F} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -(w1_{i_S}\gamma_{i_S} + (1-w1_{i_S})\gamma_{i_S}) & w1_{i_S}\gamma_{i_S} & (1-w1_{i_S})\gamma_{i_S} & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & -((1-w2_i)\beta_i + w2_i\beta_i) & 0 & (1-w2_i)\beta_i & w2_i\beta_i & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & -(\beta_i) & \beta_i & 0 & 0 \\
0 & \delta_i & 0 & 0 & 0 & 0 & 0 & -(\delta_i) & 0 & 0 \\
0 & (1-w3)\delta_i & 0 & 0 & 0 & 0 & 0 & 0 & -((1-w3)\delta_i) & 0 \\
\eta & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -\eta
\end{bmatrix}
$$

2.3

System Analysis

Steady-State Distribution

Definition: Global Balance Equations

The **Global Balance Equations** represents the **equilibrium** obtained when the system **reaches the steady state**.

Total rate of the system **leaving the state *i*** in equilibrium

Total rate of the system **entering the state *i*** in equilibrium

$$\pi_i \sum_{x_j \in S, j \neq i} q_{ij} = \sum_{x_j \in S, j \neq i} \pi_j q_{ji}$$

Stationary probability of state *i*

All possible transition from and to *i*

Transition rate from *i* to *j*

In **steady-state** condition, the **leaving** rate **must be equal** to the **entering** rate.

Definition: Global Balance Equations

In a **matrix perspective**, if we define **π=[π1,π2,...,πn]** the **vector of all stationary probability** and **$Q=[q_{ij}]$** the **transition rate matrix**, we have that the **Global Balance Equation** can be expressed as:

$$\pi Q = 0$$

Since in **steady-state** condition, the **leaving** rate **must be equal** to the **entering** rate.

Definition: Steady-State Distribution

The **Steady-State Distribution** represents the **probability distribution** of the states within the system once a **long-term state of equilibrium** is reached.
The distribution can be computed by **solving the global balance equations** system:

$$\pi Q = 0 \rightarrow Q^t \pi = 0$$

and applying the **normalization constraint** to eliminate redundancy

$$\sum_{x_i \in S} \pi_i = 1$$

Requirement: Transposed and Normalized IGM

$$
\begin{bmatrix}
-hn & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \eta \\
nh & -(p_{FF}n + p_F n + p_S n) & 0 & 0 & 0 & 0 & 0 & \delta_i & (1-w_3)\delta_i & 0 \\
0 & p_{FF}n & -(w1_{i_{FF}}\gamma_{i_{FF}} + (1-w1_{i_{FF}})\gamma_{i_{FF}} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & p_F n & 0 & -(w1_{i_F}\gamma_{i_F} + (1-w1_{i_F})\gamma_{i_F} & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & p_S n & 0 & 0 & -(w1_{i_S}\gamma_{i_S} + (1-w1_{i_S})\gamma_{i_S} & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & w1_{i_{FF}}\gamma_{i_{FF}} & w1_{i_F}\gamma_{i_F} & w1_{i_S}\gamma_{i_S} & -((1-w2_i)\beta_i + w2_i\beta_i & 0 & 0 & 0 & 0 \\
0 & 0 & (1-w1_{i_{FF}})\gamma_{i_{FF}} & (1-w1_{i_F})\gamma_{i_F} & (1-w1_{i_S})\gamma_{i_S} & 0 & -(\beta_i) & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & (1-w2_i)\beta_i & \beta_i & -\delta_i & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & w2\beta_i & 0 & 0 & -((1-w3)\delta_i) & 0 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1
\end{bmatrix}
$$

Steady-State: Distribution System

$$
\begin{cases}
\pi_{S1} \cdot (-hn) + \pi_{S10} \cdot \eta = 0 \\
\pi_{S1} \cdot nh + \pi_{S2} \cdot (-(p_{FF}n + p_F n + p_S n)) + \pi_{S8} \cdot \delta_1 + \pi_{S9} \cdot (1 - w3)\delta_1 = 0 \\
\pi_{S2} \cdot p_{FF}n + \pi_{S3} \cdot (-(w1_{1_{FF}}\gamma_{1FF} + (1 - w1_{1_{FF}})\gamma_{1_{FF}} = 0 \\
\pi_{S2} \cdot p_F n + \pi_{S4} \cdot (-(w1_{1_F}\gamma_{1F} + (1 - w1_{1_F})\gamma_{1_F} = 0 \\
\pi_{S2} \cdot p_S n + \pi_{S5} \cdot (-(w1_{1_S}\gamma_{1S} + (1 - w1_{1_S})\gamma_{1_S} = 0 \\
\pi_{S3} \cdot w1_{1_{FF}}\gamma_{1_{FF}} + \pi_{S4} \cdot w1_{1_F}\gamma_{1_F} + \pi_{S5} \cdot w1_{1_S}\gamma_{1_S} + \pi_{S6} \cdot (-((1 - w2_1)\beta_1 + w2_1\beta_1)) = 0 \\
\pi_{S3} \cdot (1 - w1_{1_{FF}})\gamma_{1_{FF}} + \pi_{S4} \cdot (1 - w1_{1_F})\gamma_{1_F} + \pi_{S5} \cdot (1 - w1_{1_S})\gamma_{1_S} + \pi_{S7} \cdot (-(\beta_1)) = 0 \\
\pi_{S6} \cdot (1 - w2_1)\beta_1 + \pi_{S7} \cdot \beta_1 + \pi_{S8} \cdot (-\delta_1) = 0 \\
\pi_{S6} \cdot w2\beta_1 + \pi_{S9} \cdot (-((1 - w3)\delta_1)) = 0 \\
\pi_{S1} + \pi_{S2} + \pi_{S3} + \pi_{S4} + \pi_{S5} + \pi_{S6} + \pi_{S7} + \pi_{S8} + \pi_{S9} + \pi_{S10} = 1
\end{cases}
$$

This system of linear equations considers **only the first round**.

Steady-State Distribution

After resolving the steady state distribution system, we get the following results for the **first round**.

State	Component	Steady-State Distribution
1	NewHeight	0.018364642912109864
2	Round1	0.018364642912109864
3	$\text{Propose}_{FF}1$	0.013957128613203626
4	$\text{Propose}_{F}1$	0.03672928582421985
5	$\text{Propose}_{S}1$	0.03672928582421973
6	Prevote1	0.014468889995687616
7	NilPrevote1	0.0222603958255321
8	PreCommit1	0.009145785366274129
9	Unsuccess1	0.027583500457945612
10	Commit1	0.009136639581907874

03

Metrics

Starting from the Steady-State Distribution

Performance metric: Throughput

The Throughput measure of an activity in respect to the system corresponds to the **expected number of completed activities per unit of time**.
In our case, it corresponds to **commit/s -> blocks/s**.

In order to compute it, we associate a reward equal to the rate of the activity to all the states of the system from which said activity is performed.

$$\rho_i = \begin{cases} \sum_k r_{activity} & \text{If the activity is performed} \\ 0 & \text{Otherwise} \end{cases}$$

$$T_{activity} = \sum_i \underbrace{\overbrace{\rho i}^{\text{Reward}} \pi_{Si}}_{\text{Steady-state probability}}$$

Metrics: Two Scenarios

≠

Non-homogeneous

Proposers operate **with distinct rates**.

=

Homogeneous

Proposers operate **with the same rate**.

3.1

Metrics

Non-Homogeneous Scenario

Non-Homogeneous: Parameters

Parameter	Value
$t_{1_{FF}}$	2.28
t_{1_F}	6.0
t_{1_S}	51.46
t_2	2.0

Table 1: Processing Times

Parameter	Value
p_{FF}	1/3
p_F	1/3
p_S	1/3
$w_{1_{FF}}$	0.7317
w_{1_F}	0.3935
w_{1_S}	0.0566
w_2	0.6321
w_3	0.99

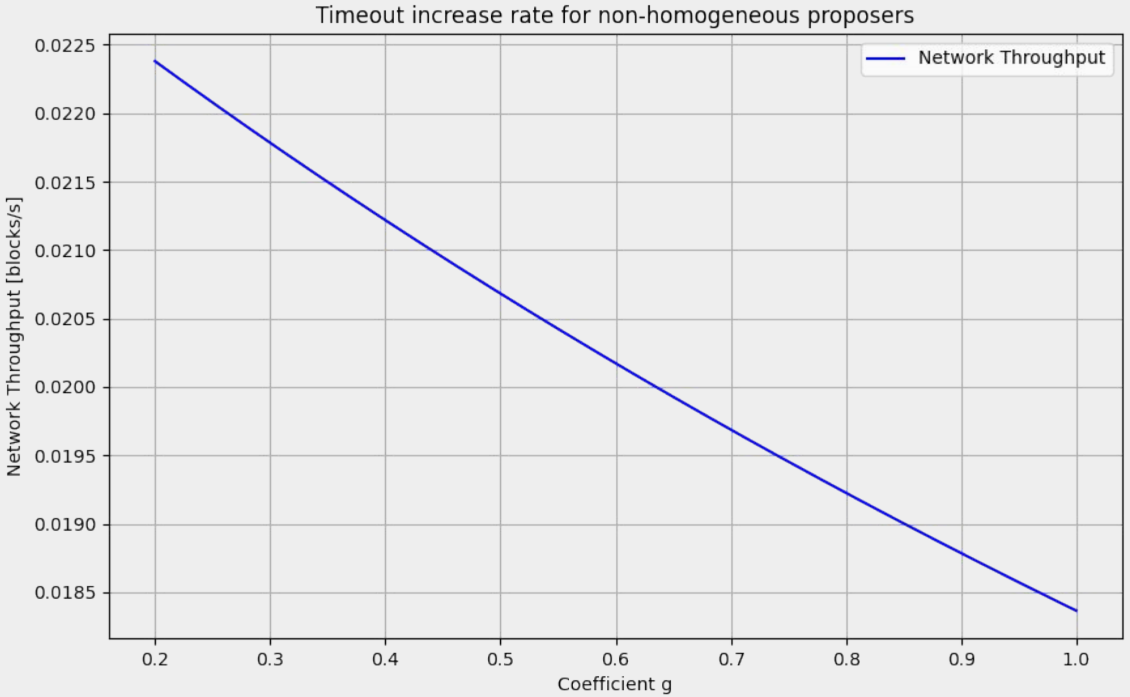
Table 2: Probabilities

Parameter	Value
T_1	6.0
T_2	2.0
T_3	2.0
T_4	2.0
g	1.0

Table 3: Timeouts

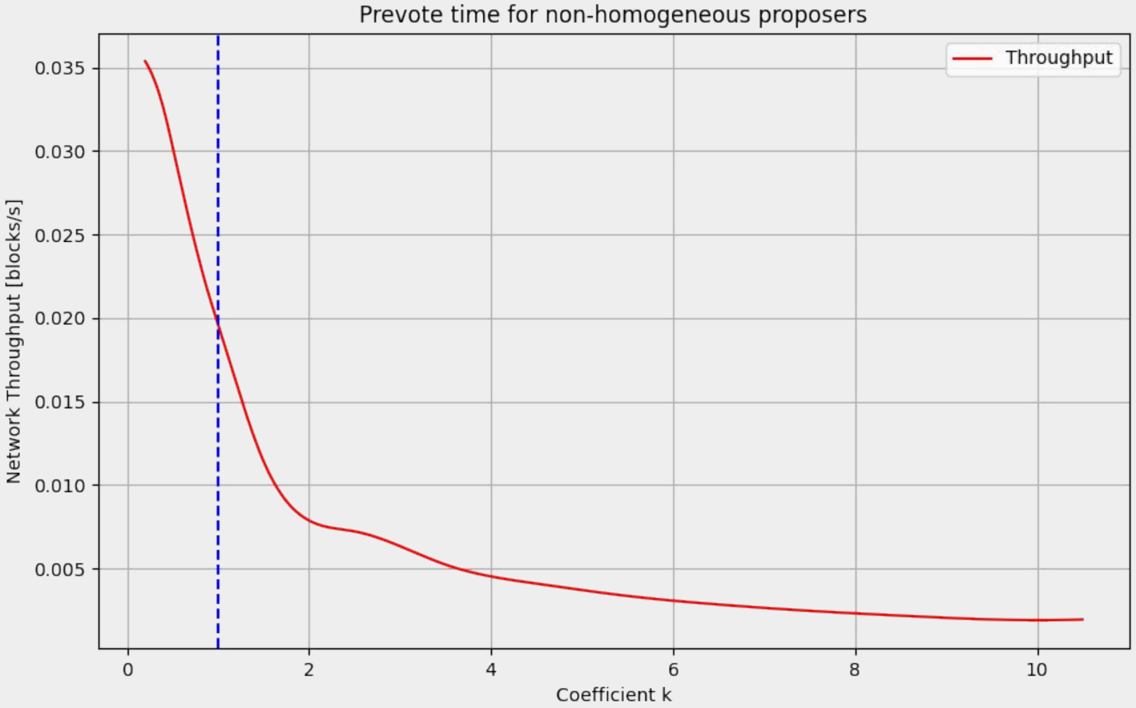
Processing times and timeouts are expressed in seconds.

Non-Homogeneous: Results



Non-Homogeneous: Results

$$t_2 = kT_2$$



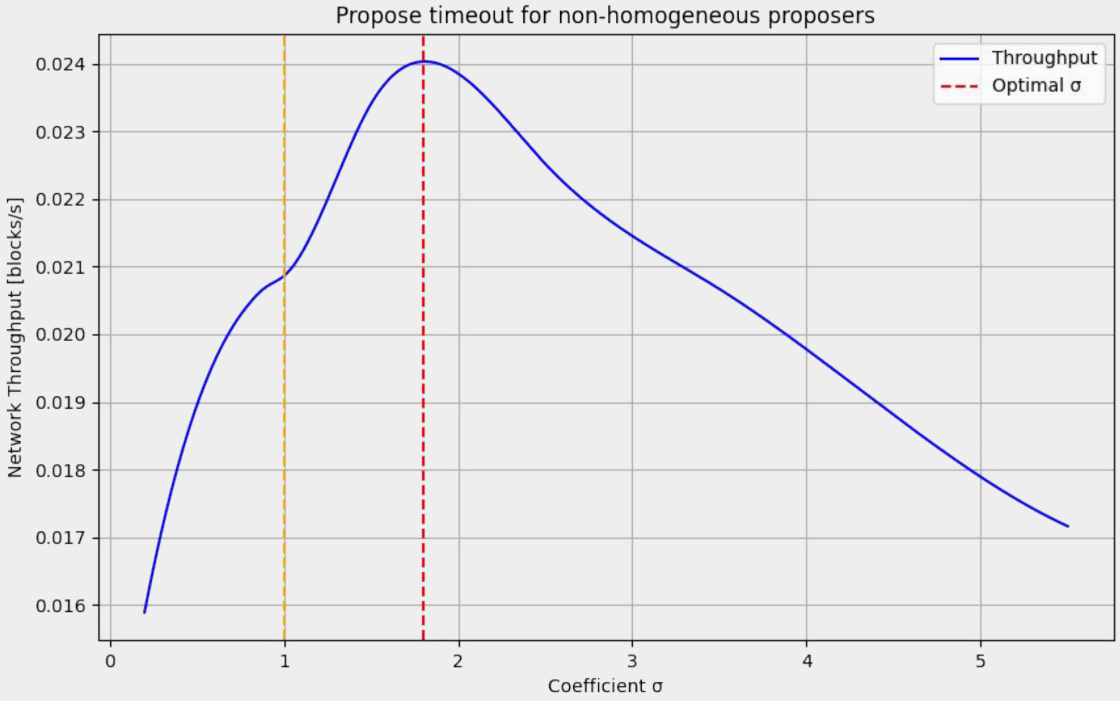
Non - Homogeneous: Results

$$T_1 = t_{1F}\sigma_1$$

Optimal Propose Timeout:

$$T_1 = t_{1F}\sigma_{\max}$$

$$T_1 = 6.0 \times 1.8 = 10.8\,[\mathrm{s}]$$



3.2

Metrics

Homogeneous Scenario

Homogeneous: Parameters

Parameter	Value
t_{1_h}	6.0
t_{2_h}	2.0

Table 4: Processing Times

Parameter	Value
w_{1_h}	0.63
w_2	0.6321
w_3	0.99

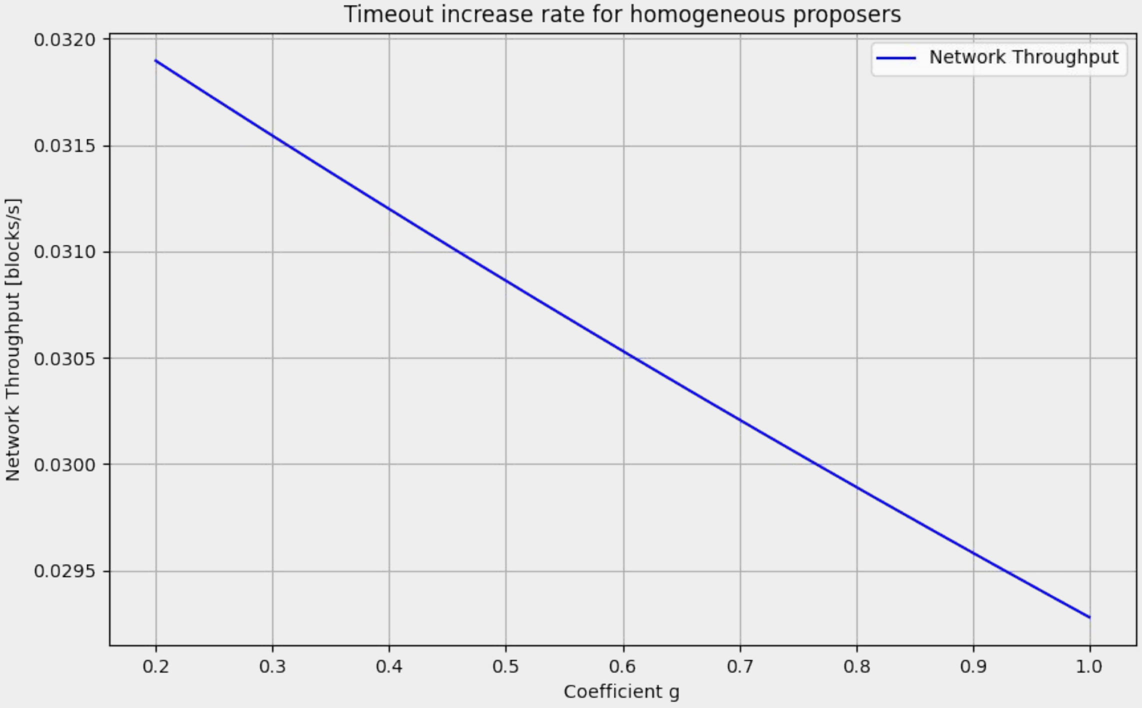
Table 5: Probabilities

Parameter	Value
T_1	6.0
T_2	2.0
T_3	2.0
T_4	2.0
g	1.0

Table 6: Timeouts

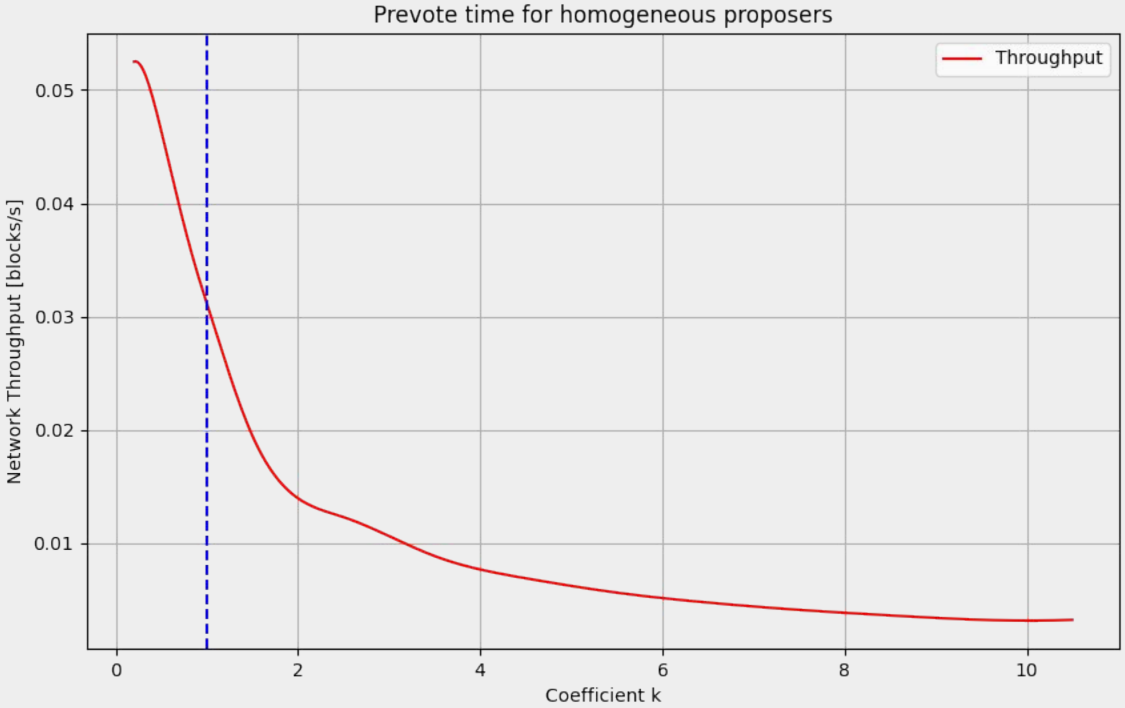
Processing times and timeouts are expressed in seconds.

Homogeneous: Results



Homogeneous: Results

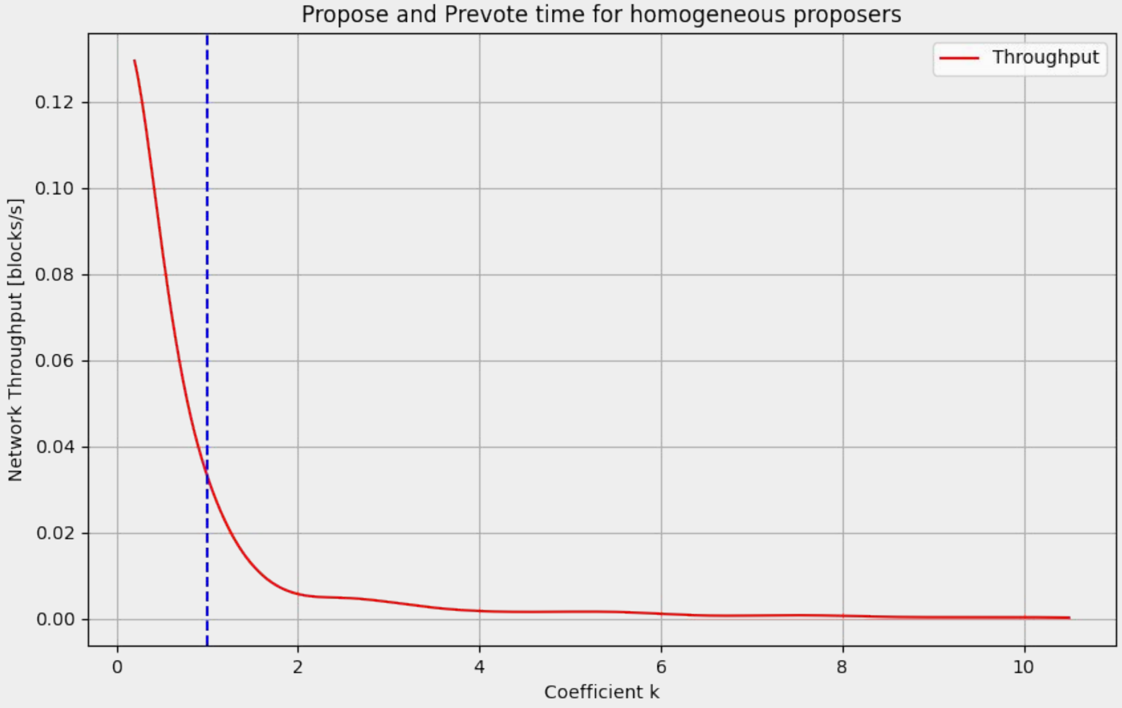
$$t_2 = kT_2$$



Homogeneous: Results

$$t_{1_h} = kT_1$$

$$t_2 = kT_2$$



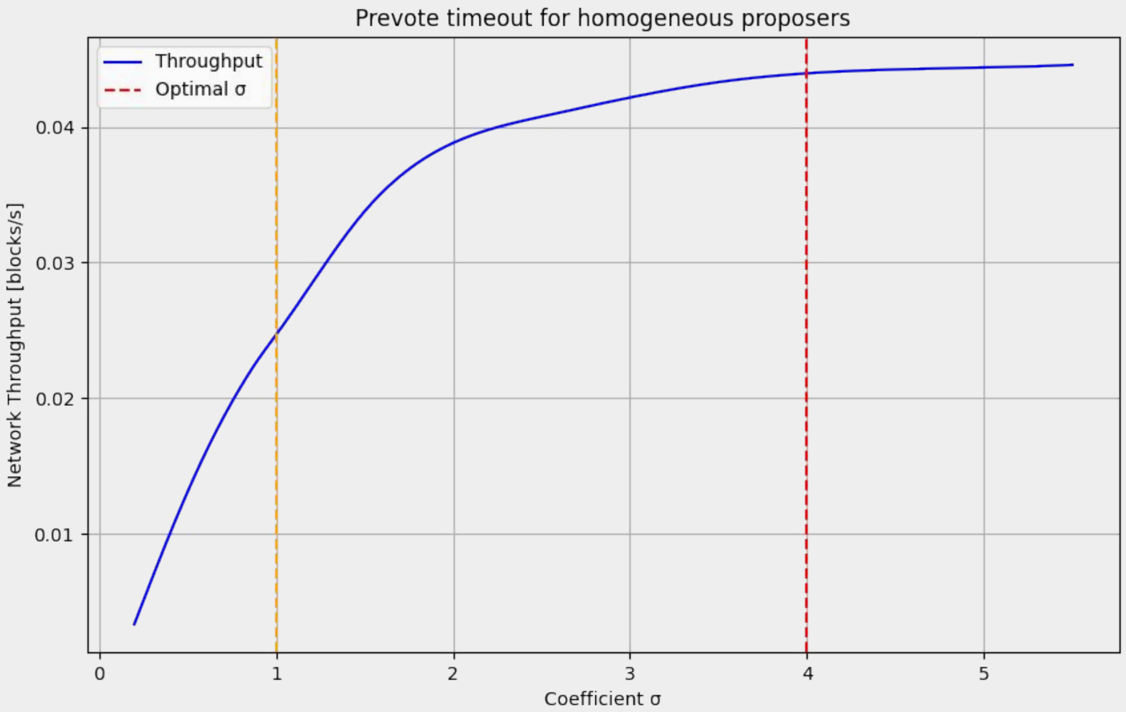
Homogeneous: Results

$$T_2 = t_{2_h}\sigma_1$$

Optimal Prevote Timeout:

$$T_2 = t_{2_h}\sigma_{max}$$

$$T_2 = 2.0 \cdot 4.0 = 8.0[s]$$



04

Extra

Notions of Lumpability

Model property: Lumpability

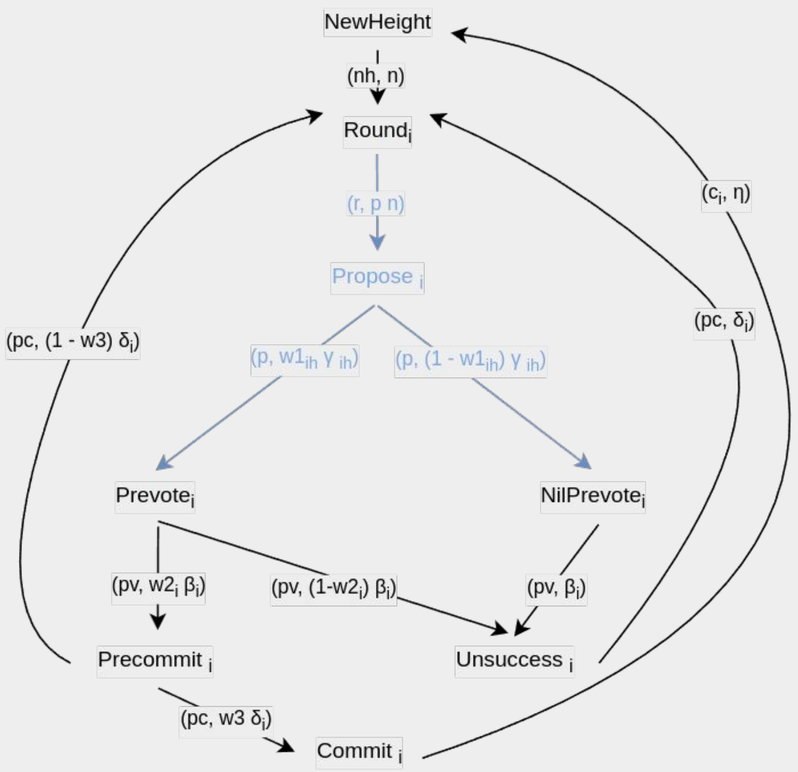
Lumpability:

a property that makes it possible to **simplify large and complex systems**. This simplification involves reducing the system to a smaller, more manageable version while **maintaining key behavioral or performance characteristics**.

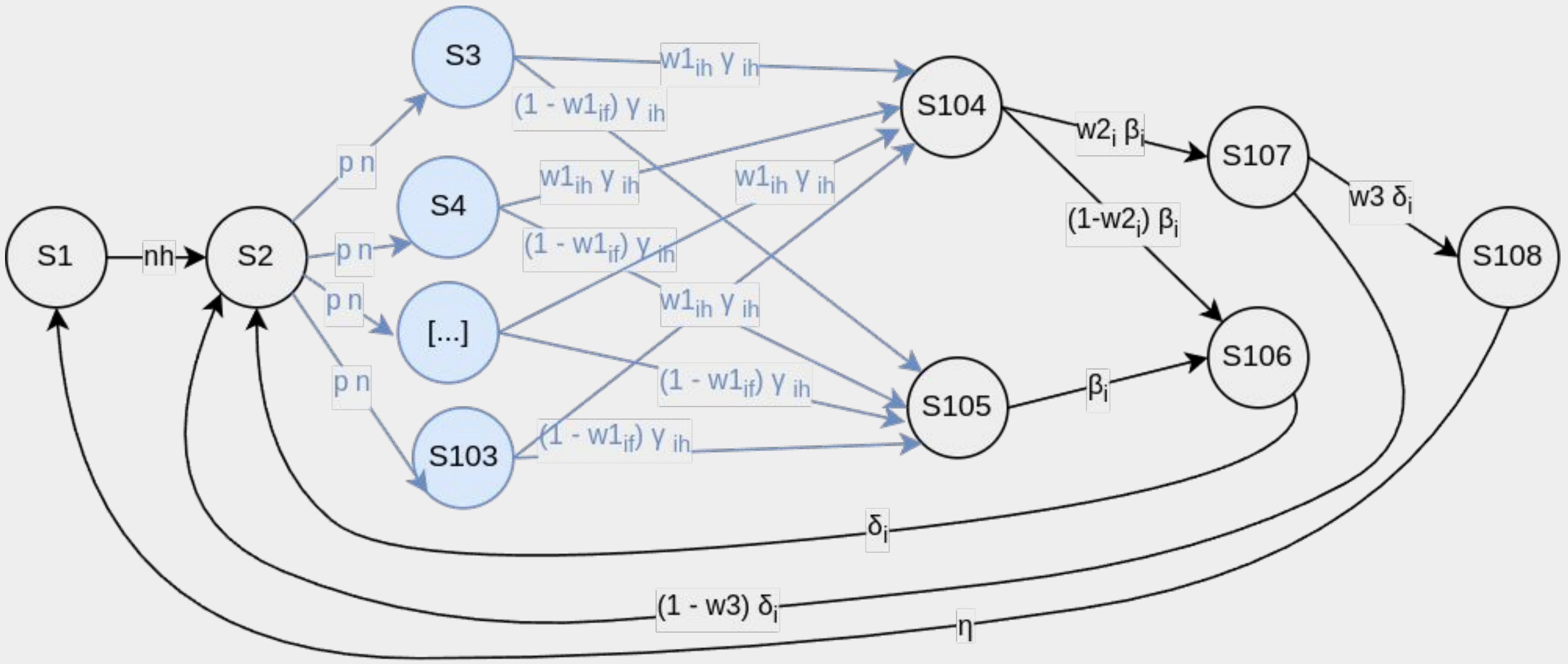
Key elements:

- **State Space reduction**: simplifies the system by condensing its state space, making it easier to analyze.
- **Equivalence relation**: Groups states into equivalence classes, enabling a meaningful simplification without losing critical details.

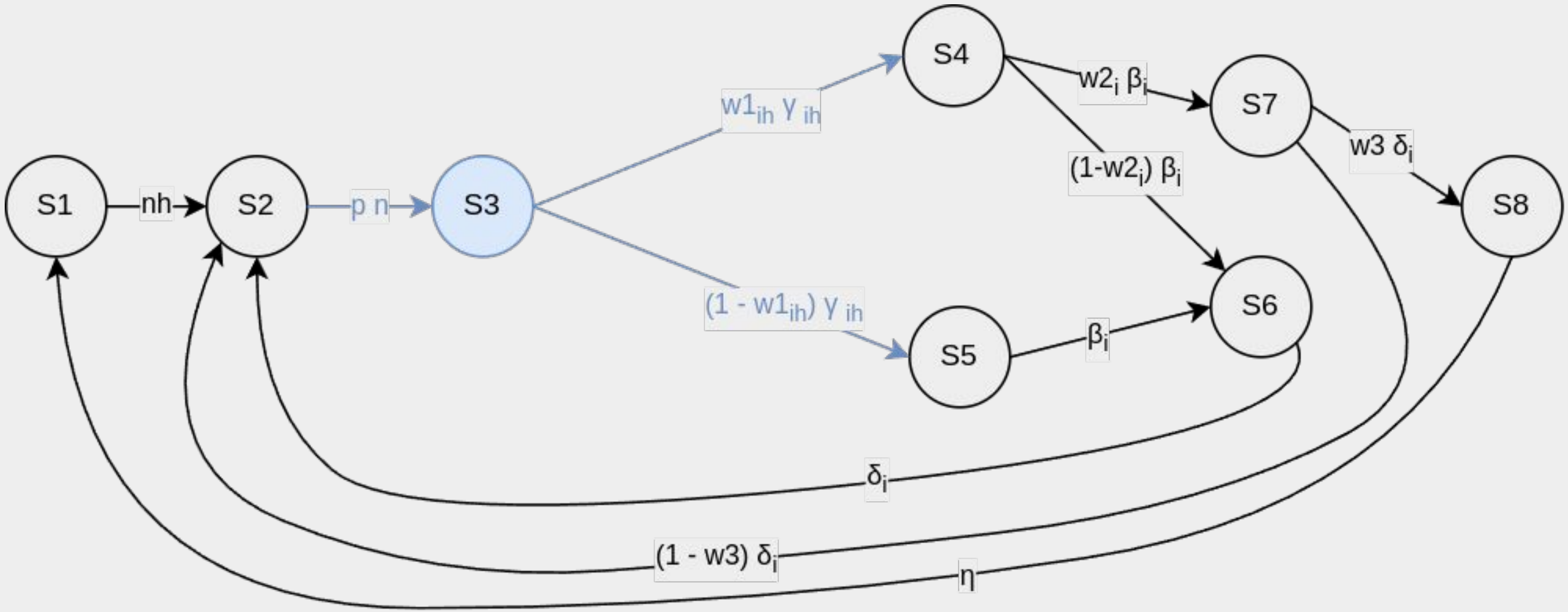
Homogeneous: Derivation Graph



Homogeneous: Non Lumped



Homogeneous: Lumped



The reduction involves both **state aggregation** and **rates simplification**.