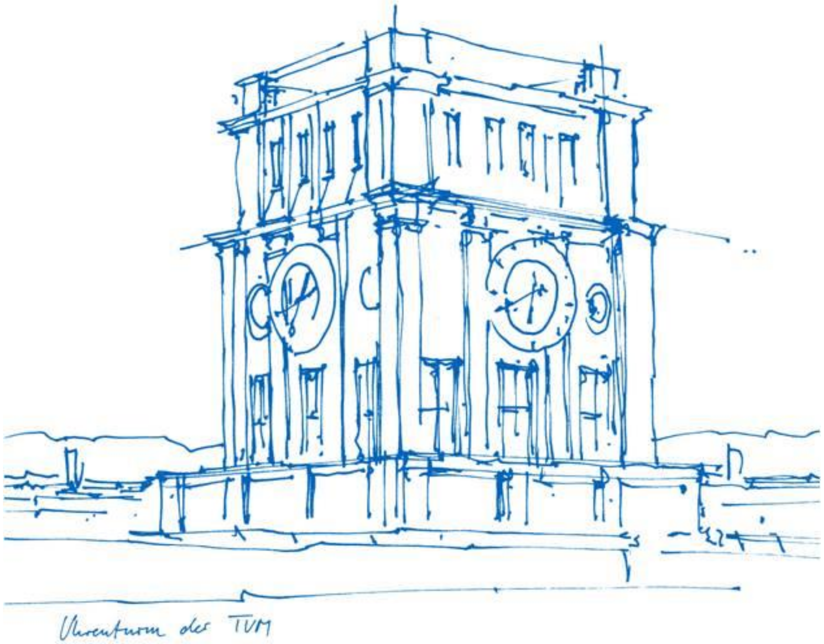
Simulation-Based Analysis of Blockchain Architectures: Double-Spend Attacks

Technische Universität München

Bachelor‘s Thesis

Leo Eichhorn

Garching, 17 July 2018



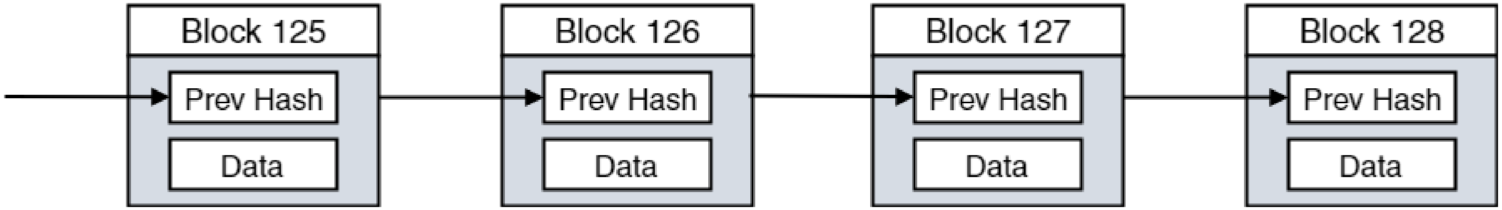
Outline

1. Context
   - Blockchain
   - Double-Spend Attacks
   - Problem Statement
   - Approach
2. Blockchain Simulation
3. Analysis
4. Empirical Model
5. Conclusion

Blockchain

- Distributed database
- Entries (blocks) are linked by their hashes
- Peer-to-peer network of nodes maintaining local copies of the blockchain
- Next block is chosen by “random” node and broadcasted to all peers

➢ No intermediate, trusted authority



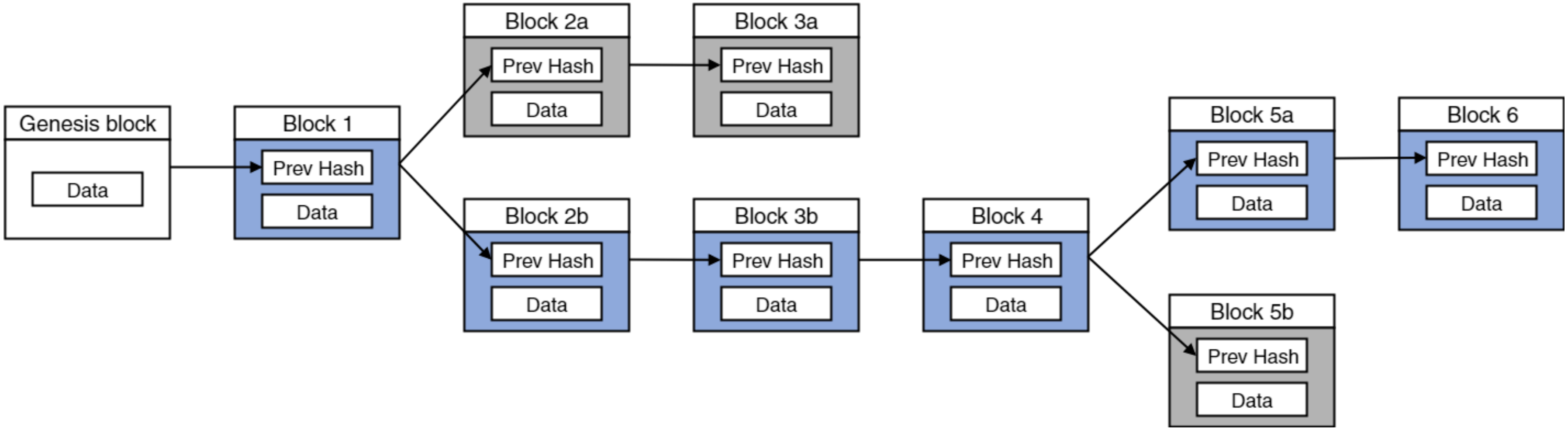
Bitcoin

- “Random” node is represented by first node solving cryptographic puzzle (proof of work)
  - ➢ Changing nonce values in block until hash smaller than a target mining difficulty
- Requires high amount of computational power
- Node is compensated with block reward and transaction fees

Block Propagation / Stale Blocks

- Peer-to-peer networks are influenced by latency times
- Two blocks mined at roughly the same time: branch in blockchain
- Consensus is eventually retained due to *longest chain rule*
- Blocks of shorter branch turn stale

➢ Stale blocks indicate a *waste* of computational power



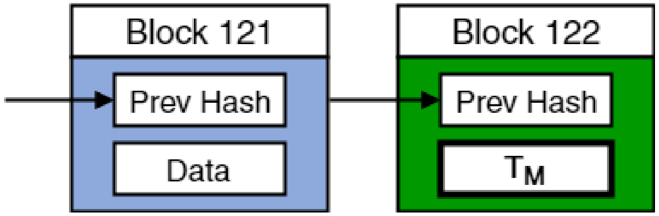
Double-Spend Attacks

- Name related to Bitcoin:
  - Group of dishonest nodes reverts transaction to a merchant after receiving the purchased product
  - Attacker needs to mine new blocks faster than the remaining network

Example DSA with 3 Confirmations

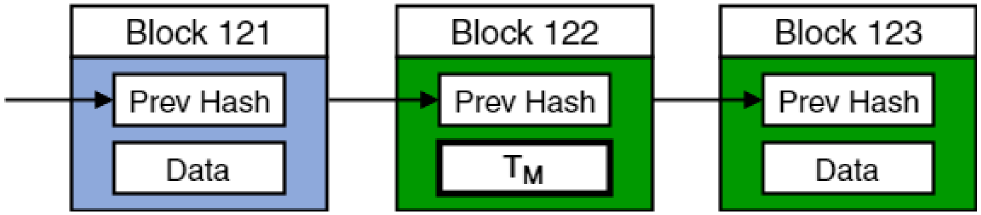
- Attacker *A* generates two transactions:
  - $\mathbf{T_M}$, to pay the merchant ($A \rightarrow M$: 500)
  - $\mathbf{T_A}$, to revert the payment ($A \rightarrow A$: 500)

- $\mathbf{T_M}$ is published an mined into the next block



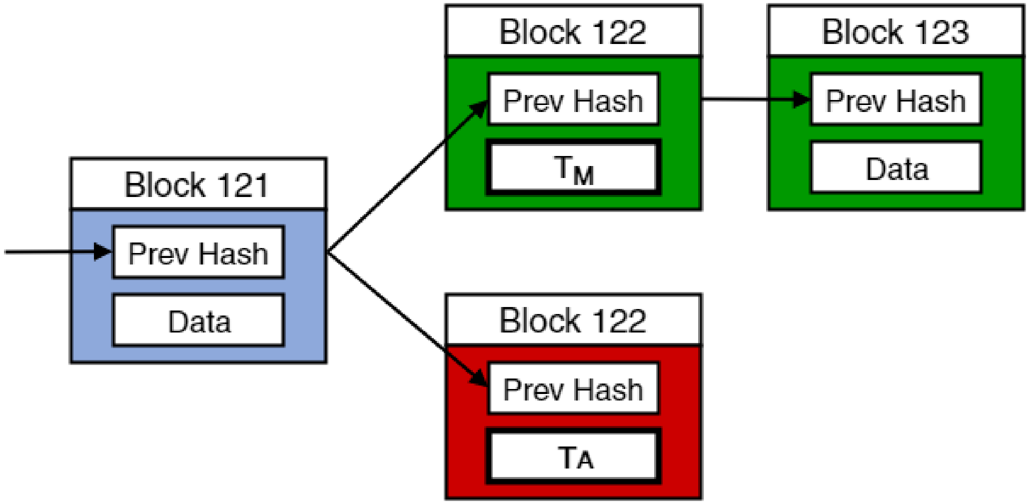
Example DSA with 3 Confirmations

- Honest network keeps mining on the longest chain
- Merchant *M* waits until payment $\mathbf{T_M}$ is confirmed



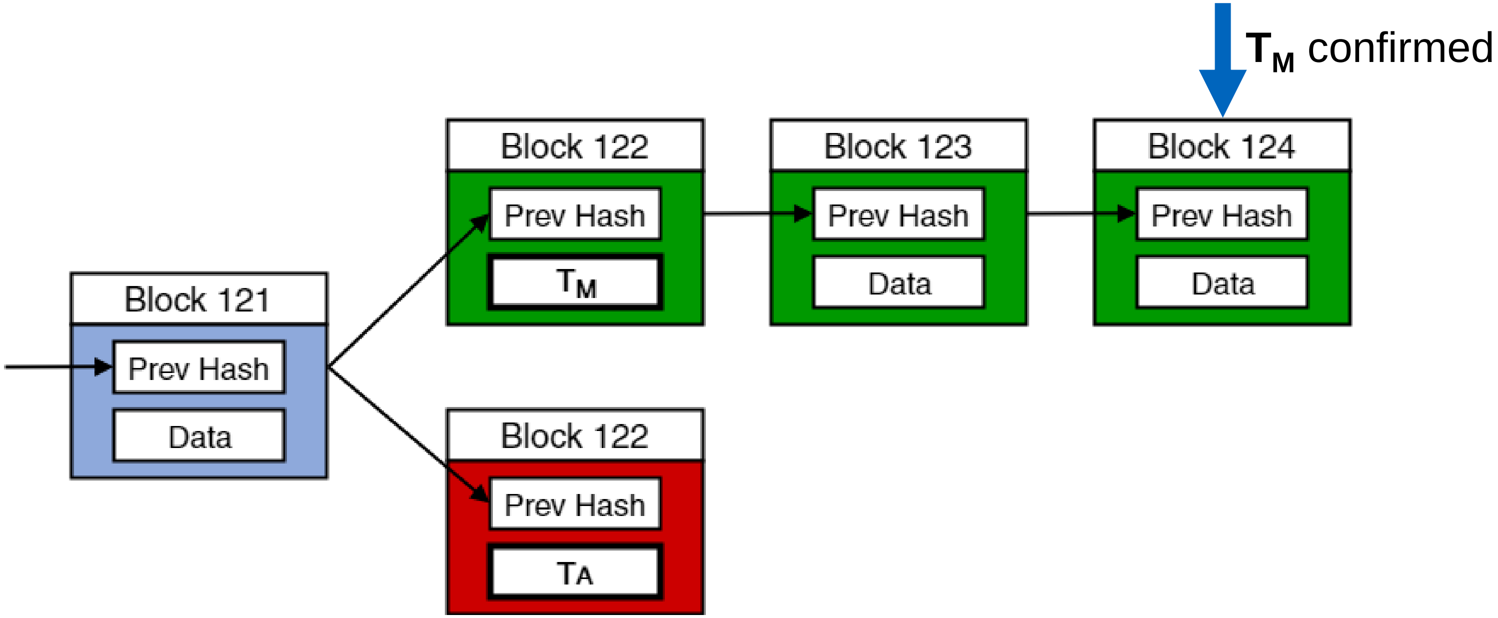
Example DSA with 3 Confirmations

- Attacking party secretly starts mining a branch containing $\mathbf{T_A}$
- On top of latest block before $\mathbf{T_M}$ is mined into the blockchain



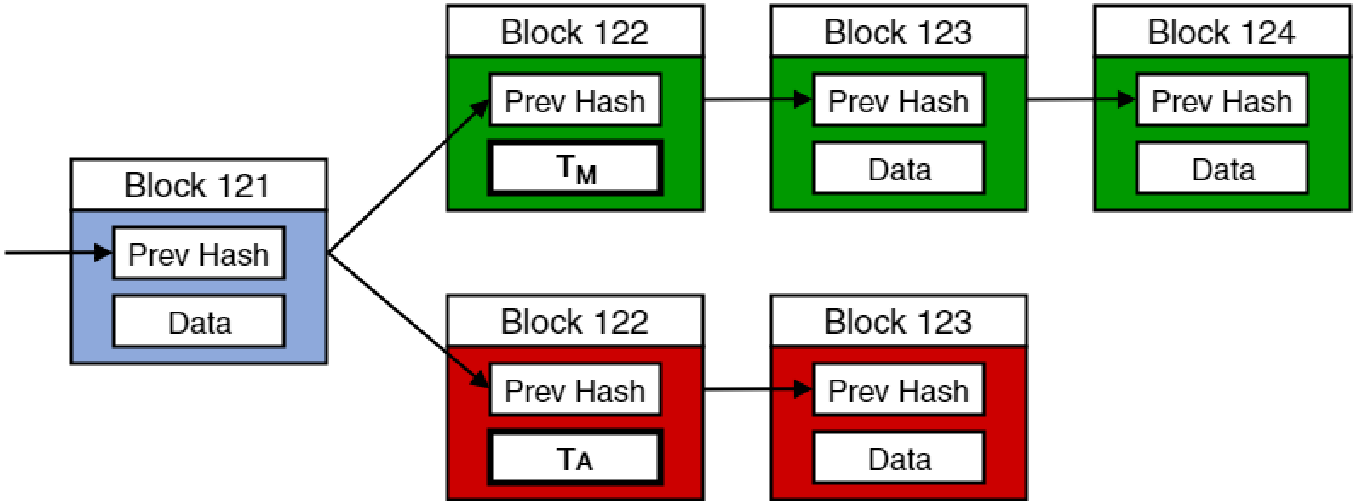
Example DSA with 3 Confirmations

- Merchant‘s payment $\mathbf{T_M}$ is confirmed
- M delivers the purchased product (irreversible)



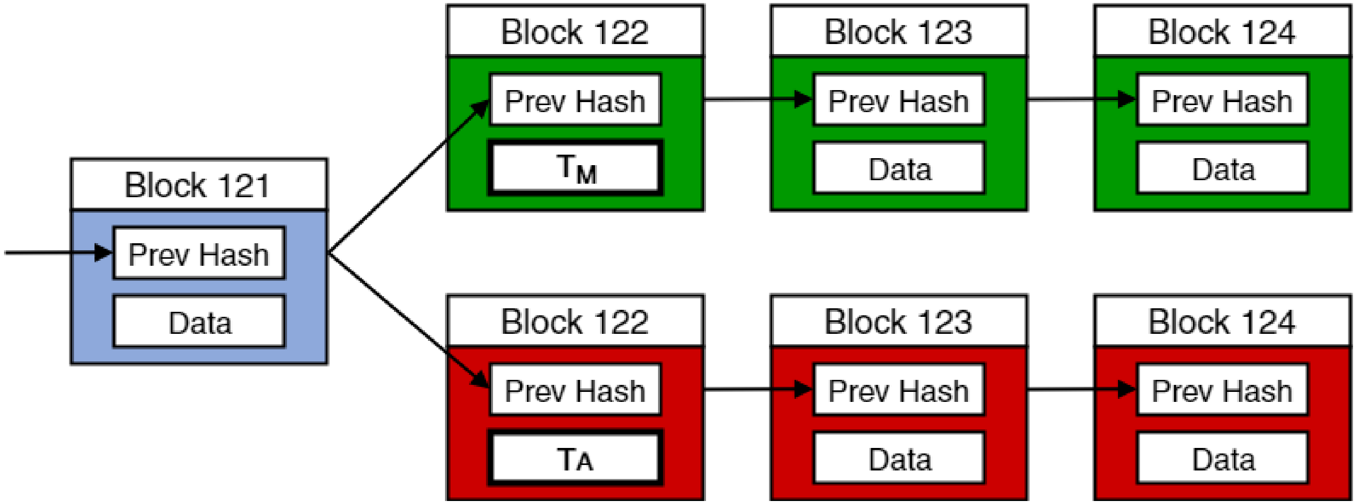
Example DSA with 3 Confirmations

- A tries to mine more blocks than the remaining network in order to replace $\mathbf{T_M}$ with $\mathbf{T_A}$



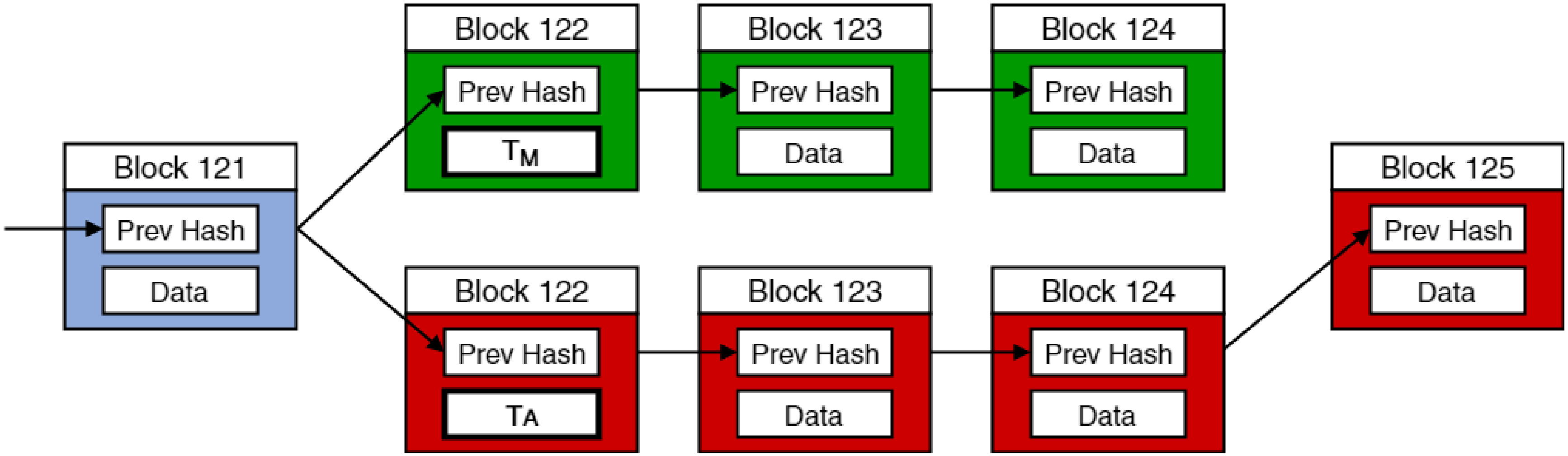
Example DSA with 3 Confirmations

- A tries to mine more blocks than the remaining network in order to replace $\mathbf{T_M}$ with $\mathbf{T_A}$



Example DSA with 3 Confirmations

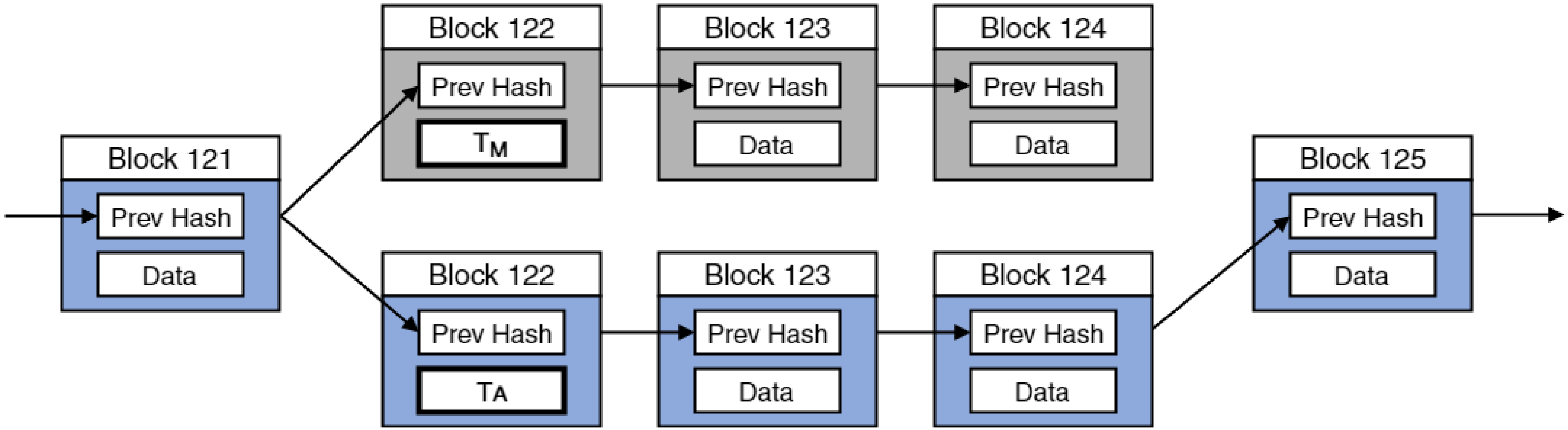
- A tries to mine more blocks than the remaining network in order to replace $\mathbf{T_M}$ with $\mathbf{T_A}$



Example DSA with 3 Confirmations

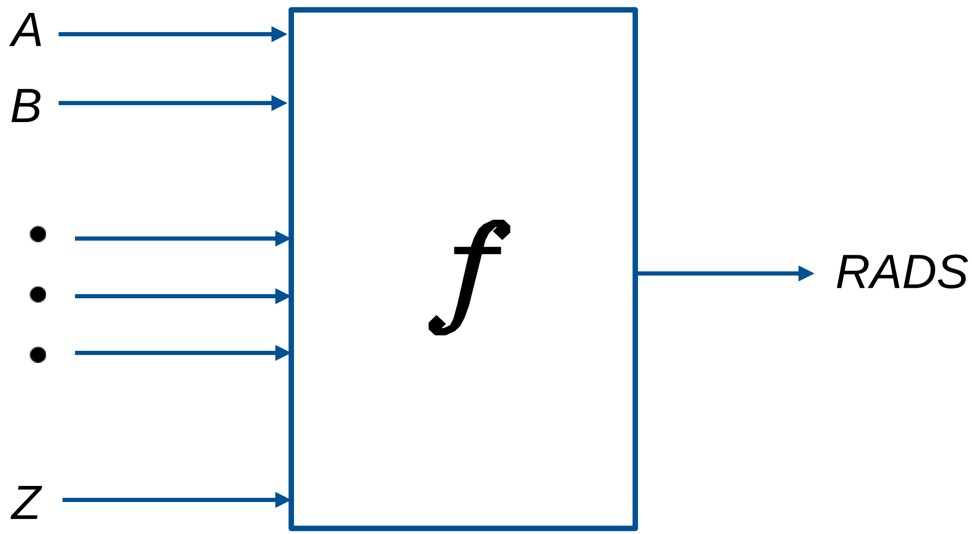
- *A* publishes the longer chain
- Blockchain containing more proof of work is new valid chain
- Branch containing $\mathbf{T_M}$ turns stale

➢ *A* keeps the delivered product and the payment



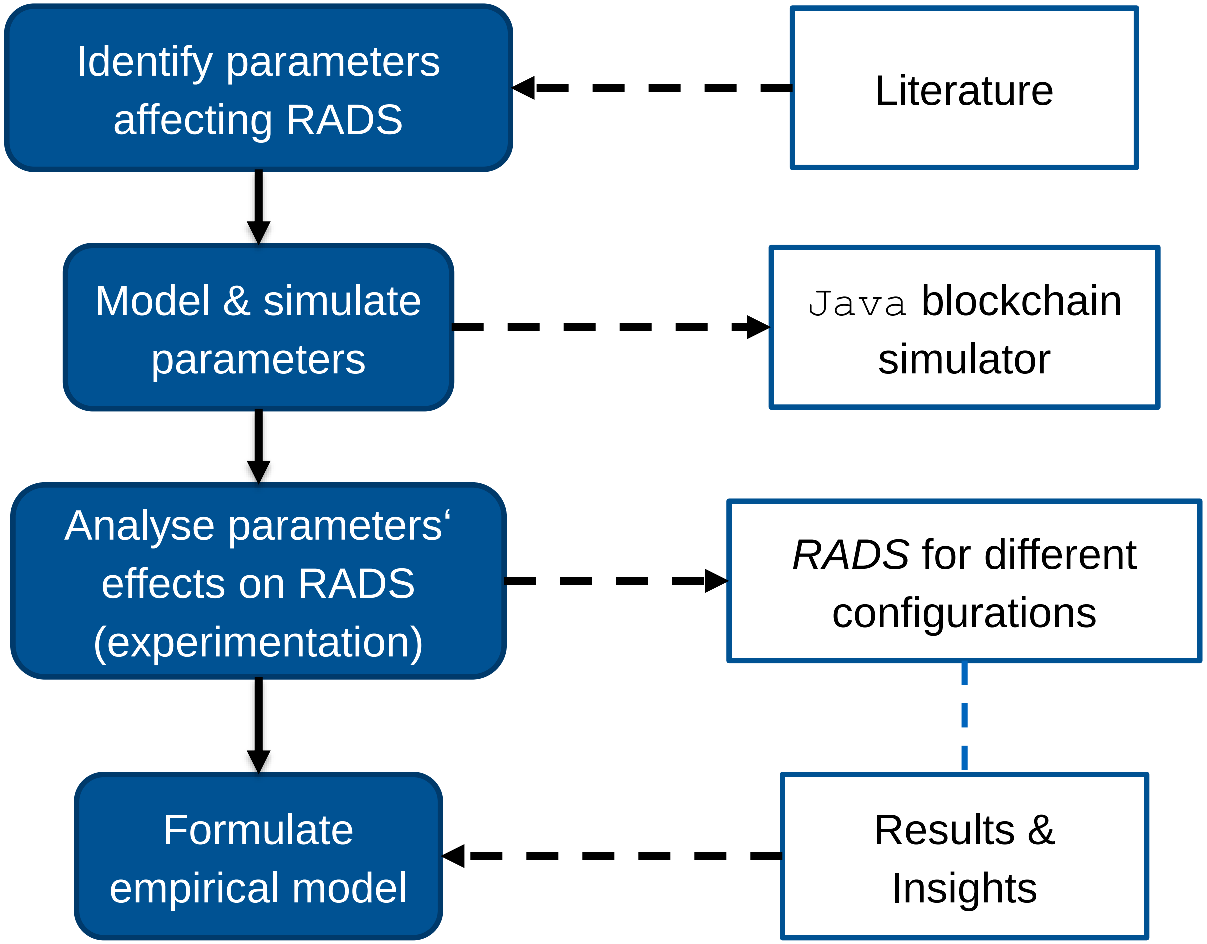
Problem Statement

- A blockchain architecture‘s resistance against double-spend attacks (RADS) may depend on many factors



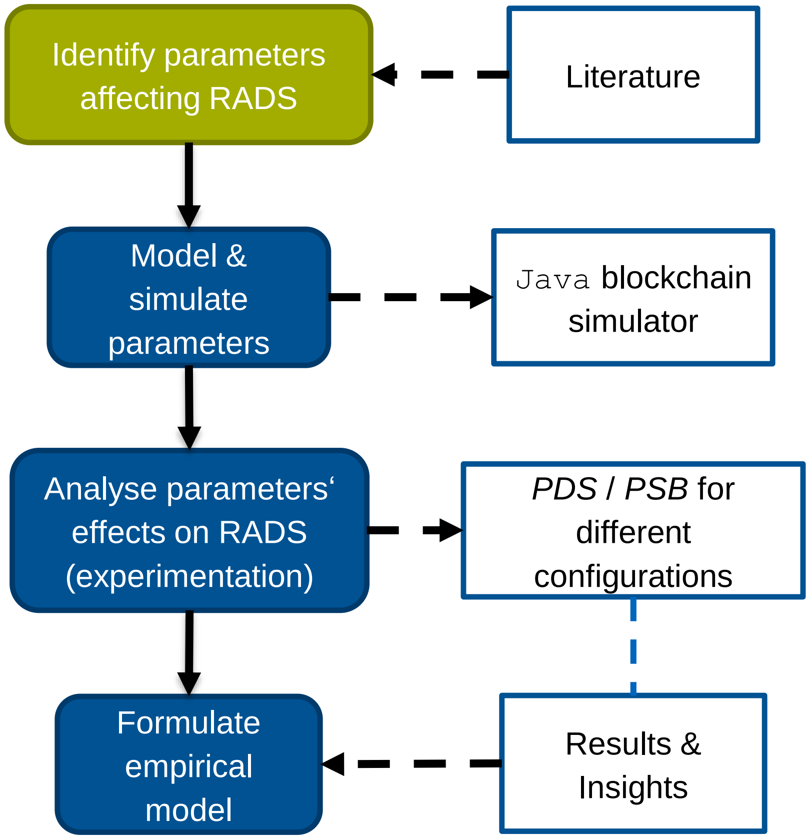
➢ Knowing more about factors affecting RADS and function f would allow architect to improve predictions

Approach

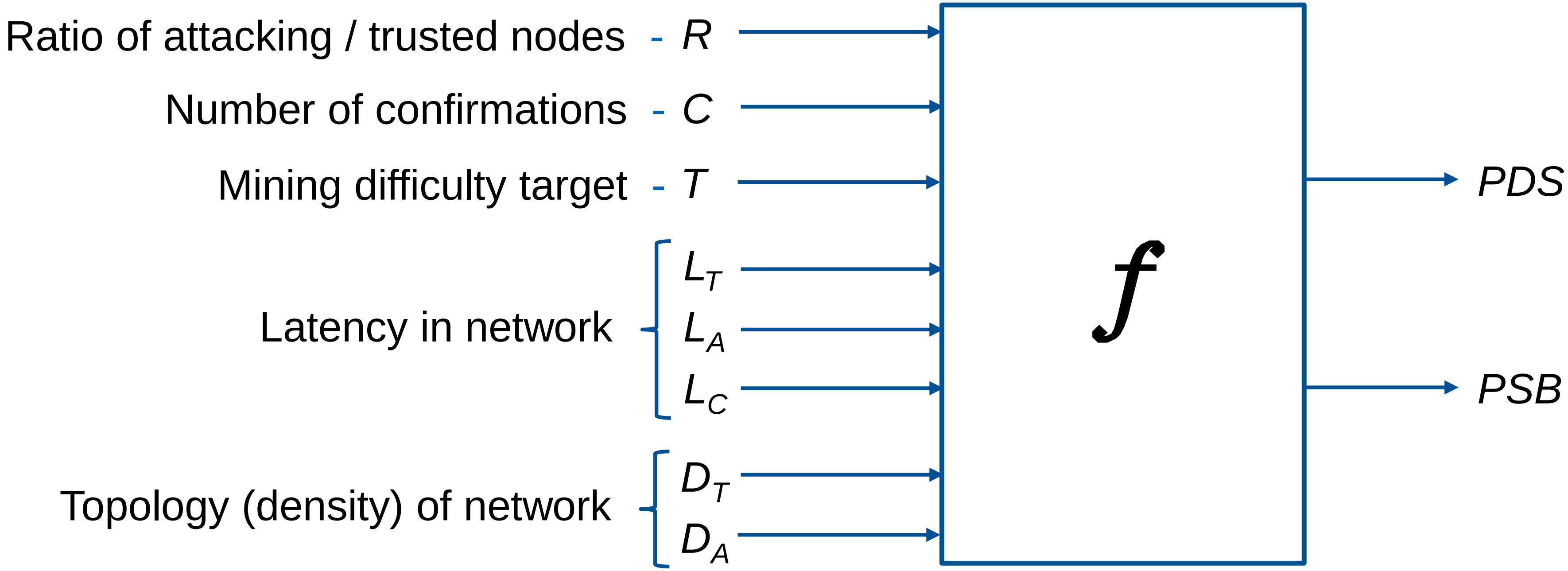

Identify parameters affecting RADS
Literature
Model & simulate parameters
Java blockchain simulator
Analyse parameters‘ effects on RADS (experimentation)
RADS for different configurations
Formulate empirical model
Results & Insights

Outline

1. Context
2. Blockchain Simulation
   – Simulation Parameters
   – Simulation Model
3. Analysis
4. Empirical Model
5. Conclusion



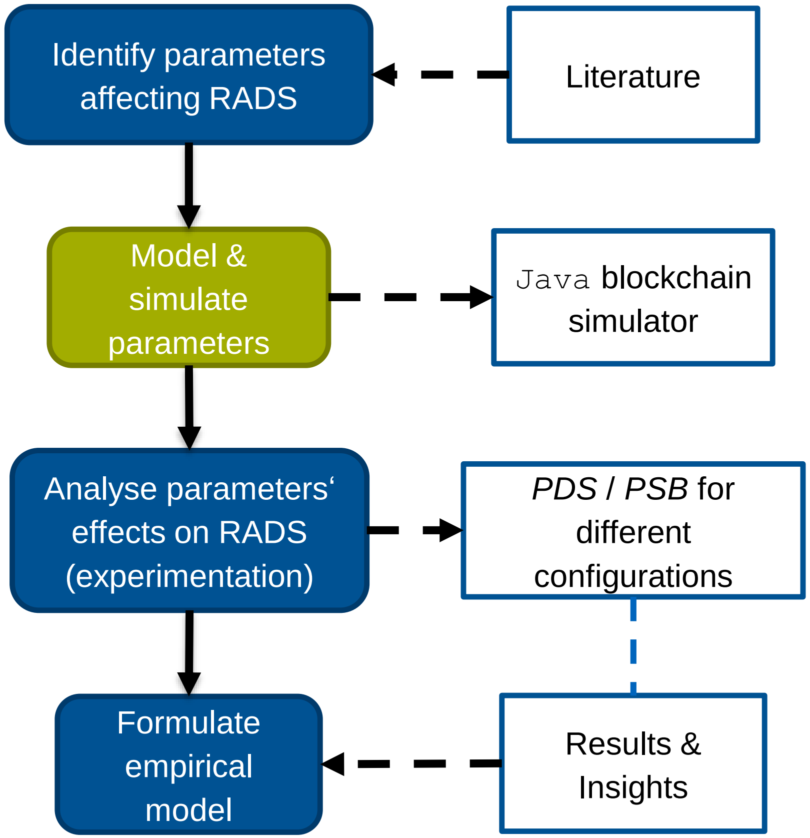
Simulation Parameters



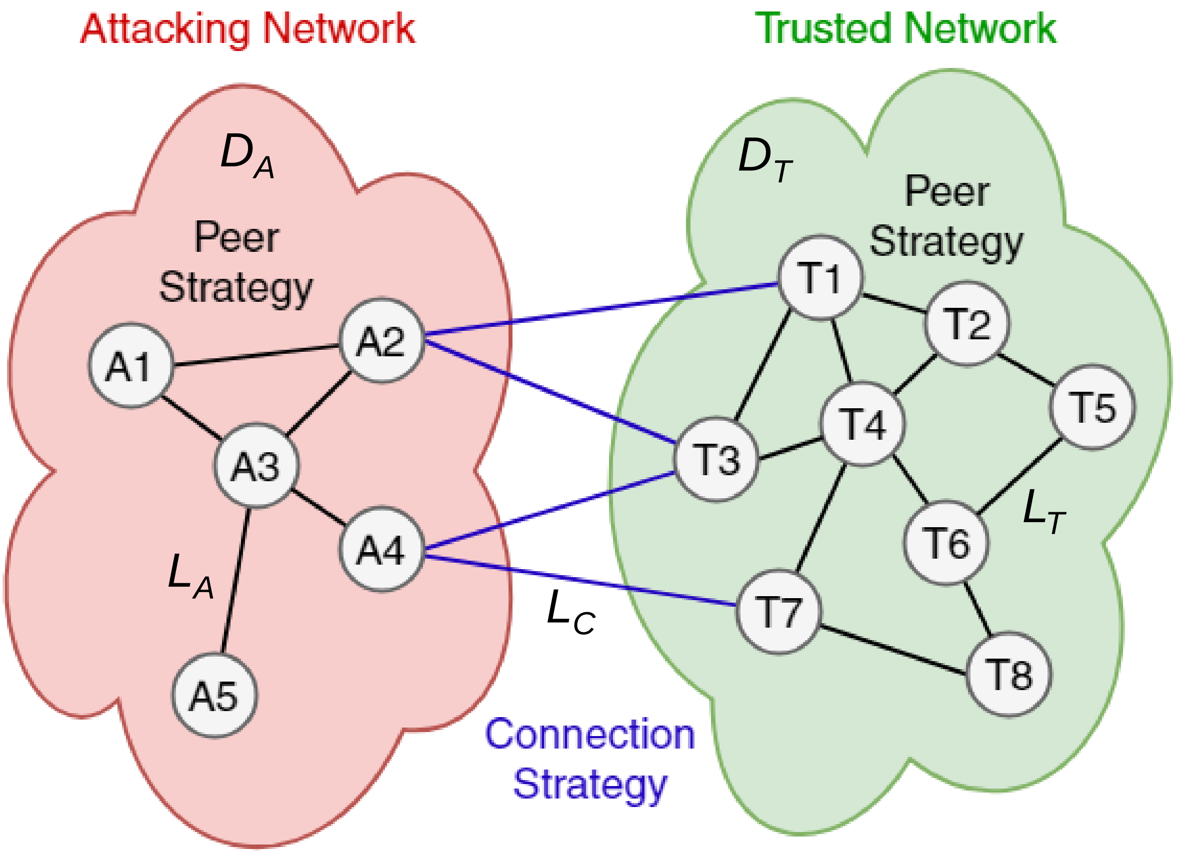
- *PDS*: Percentage of successful double-spend attacks
- *PSB*: Percentage of stale blocks

Outline

1. Context
2. Blockchain Simulation
   – Simulation Parameters
   – Simulation Model
3. Analysis
4. Empirical Model
5. Conclusion

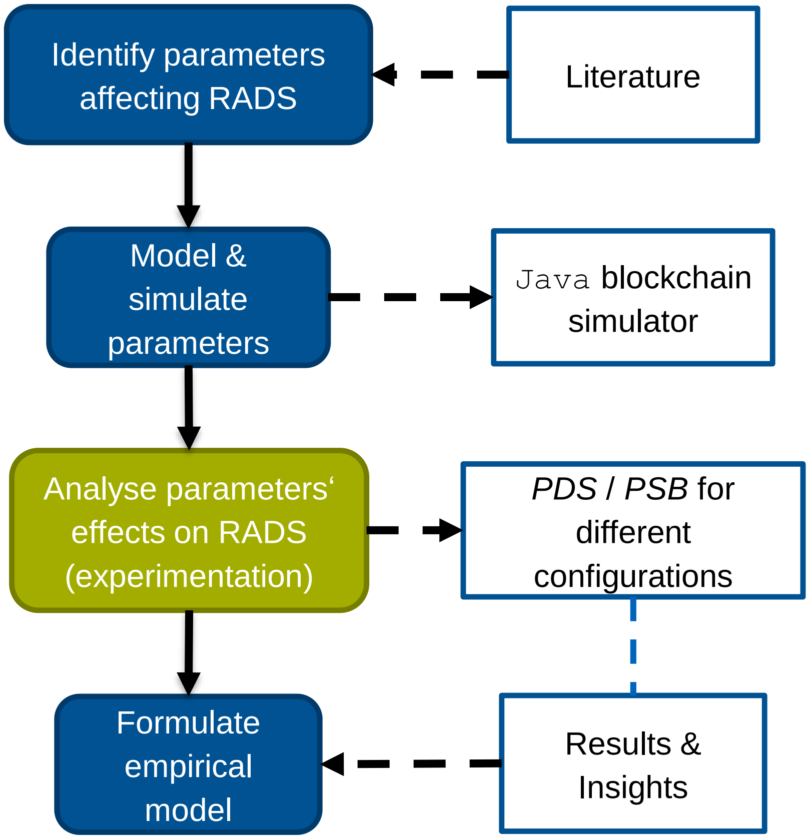


Simulation Model



Outline

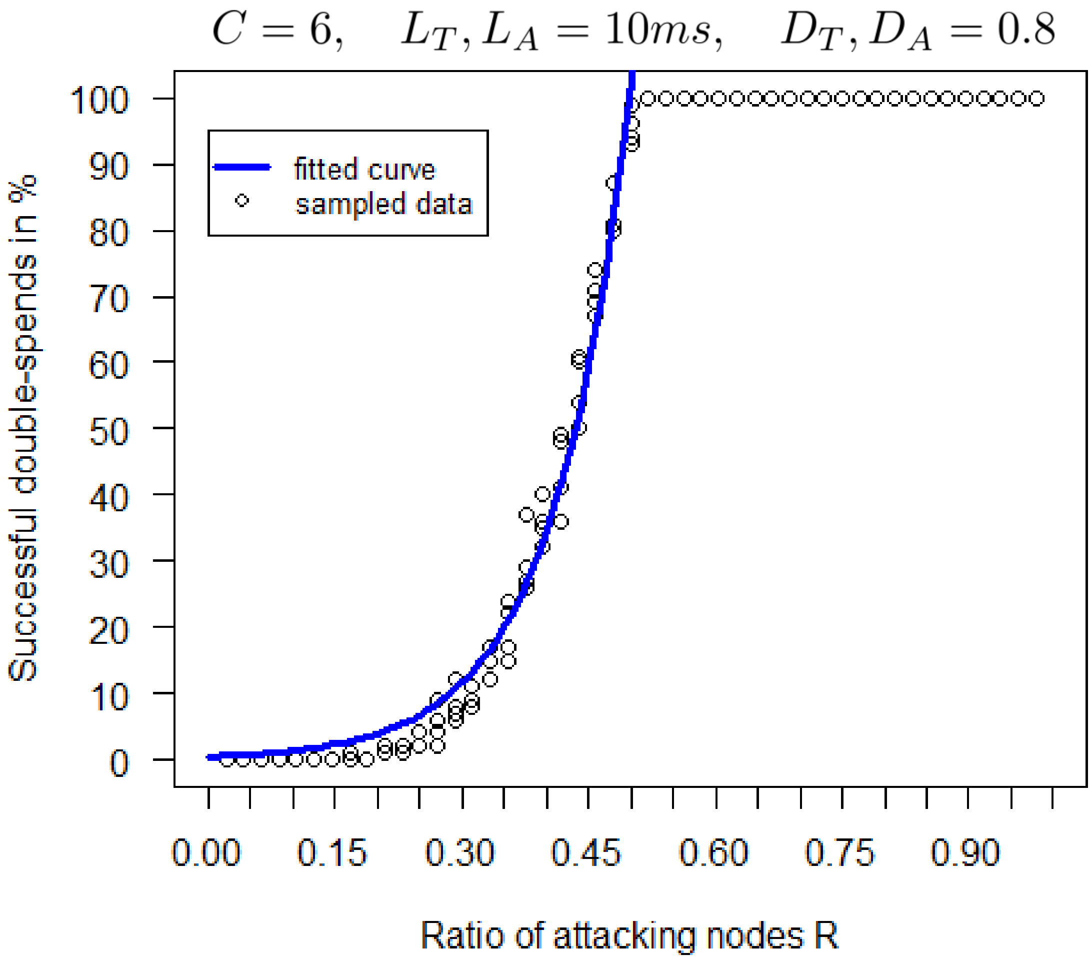
1. Context
2. Blockchain Simulation
3. Analysis
   – Experiments
   – Summary
4. Empirical Model
5. Conclusion



Ratio of attacking nodes *R*

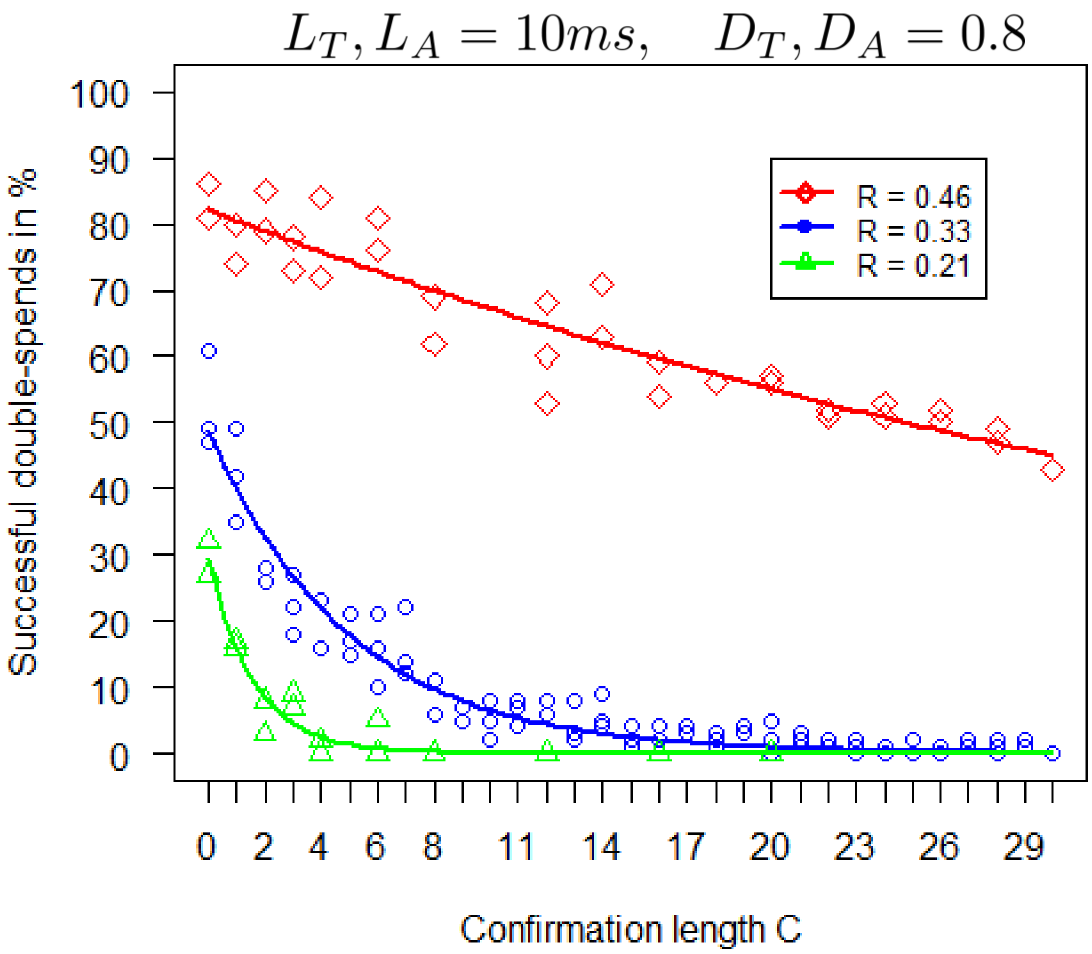
- Double-spends for *R* > 0.5 always succeed

- But: DSA at *R* = 0.5 not guranteed
  - ➢ Simulator end condition?
  - ➢ Influence of other Parameters?



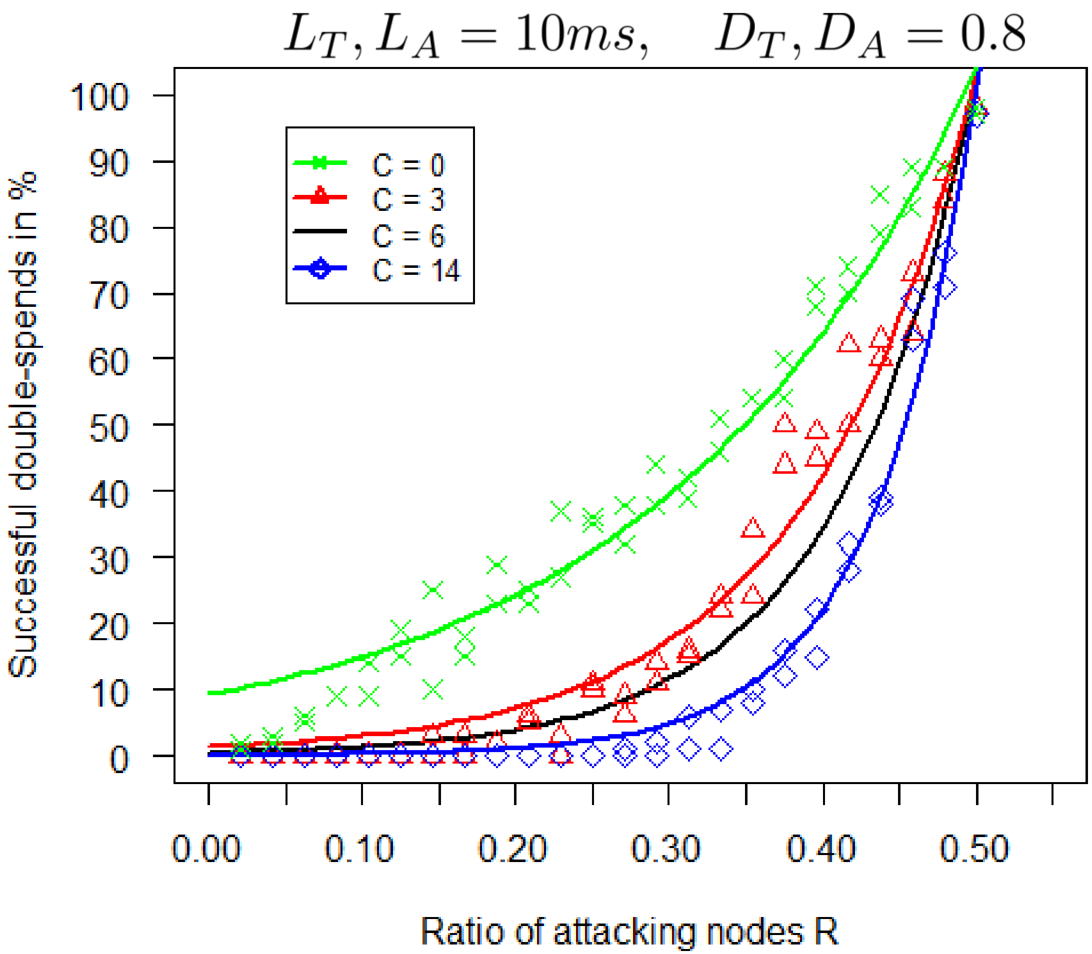
Confirmations C

- PDS decreases exponentially



Confirmations C

- PDS decreases exponentially
- No effect once majority of computing power under attackers‘ control



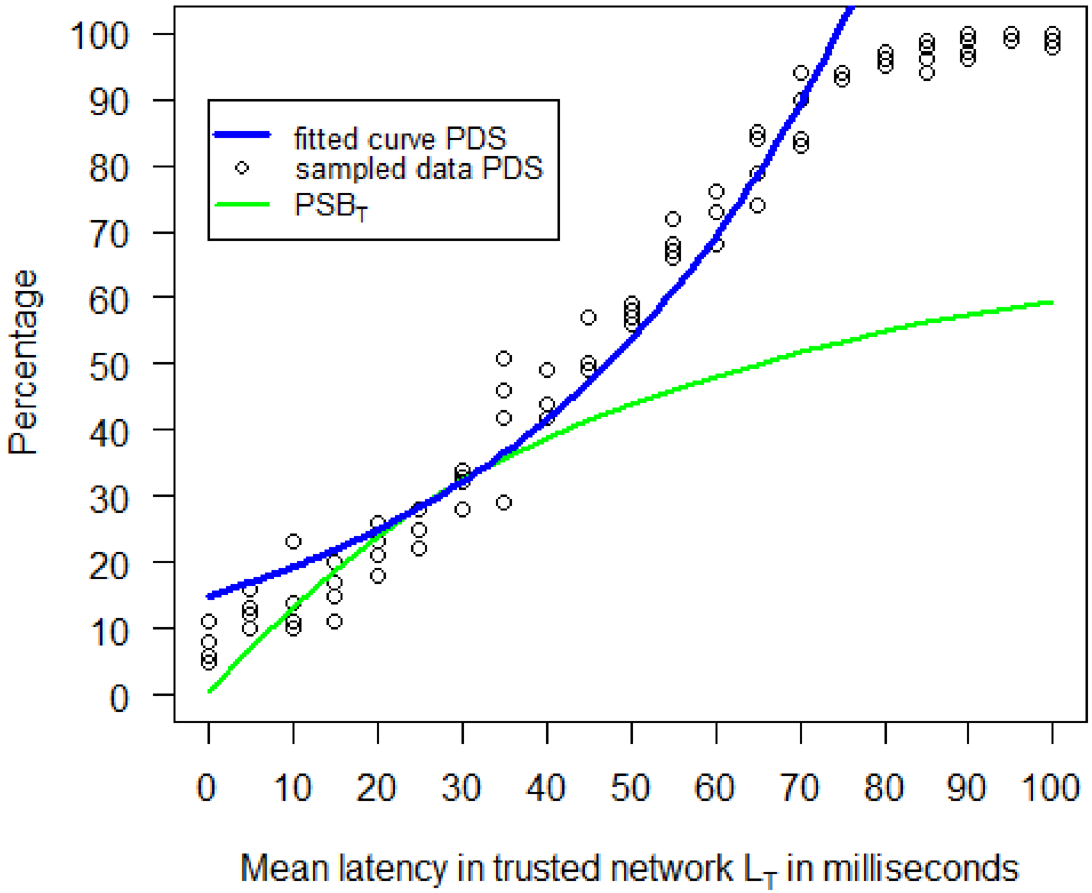
Trusted Latency L_T

- Percentage of stale blocks increases with rising latency

➢ More computing power is wasted on generation of stale blocks

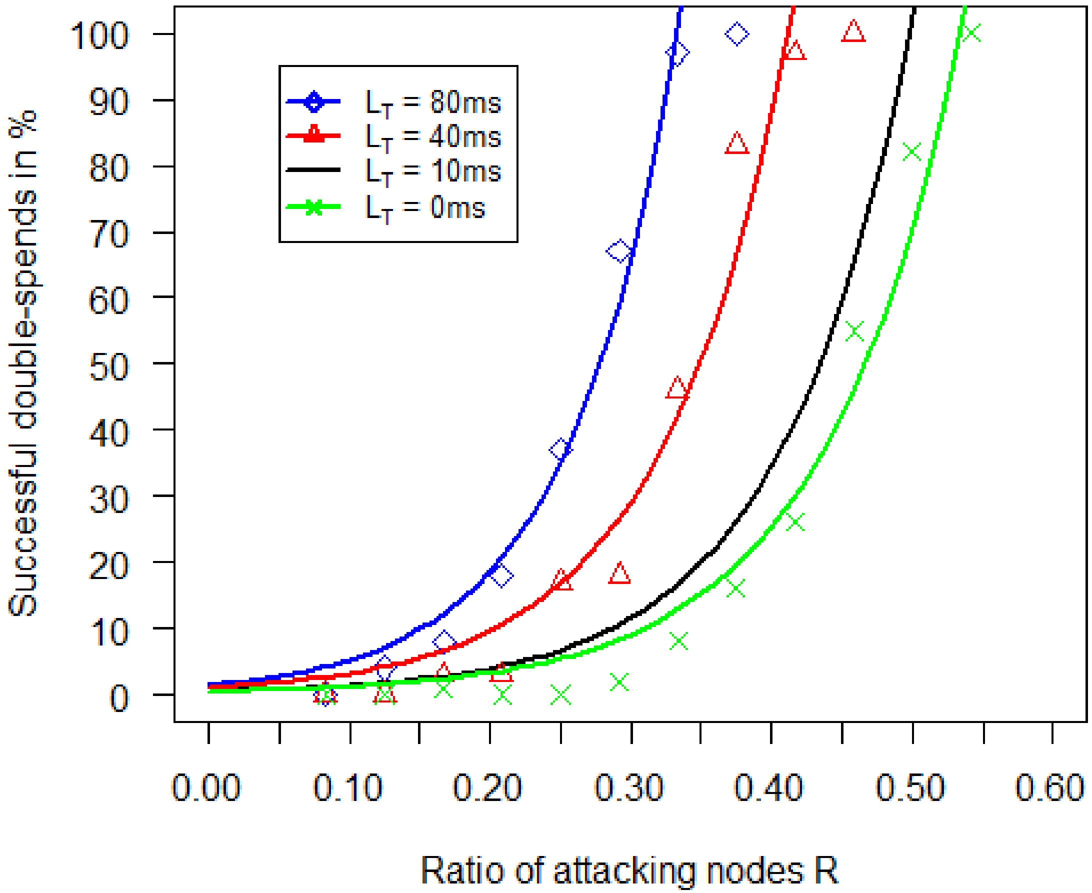
➢ Lower resistance against double-spend attacks

$$R = \frac{1}{3}, \quad C = 6, \quad L_A = 10ms$$



Trusted Latency L_T

- Direct effect on value of R
- Network density D_T produces similar effects
- Effect can be reduced by higher mining difficulty



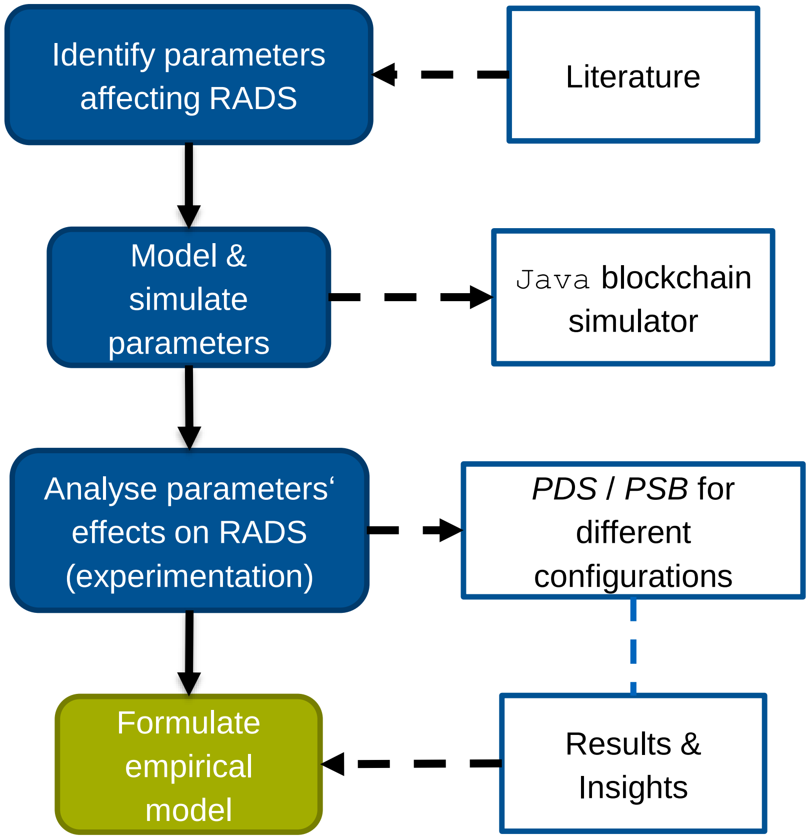
Summary

- *PDS* increases exponentially with increasing *R*
- Effective value of *R* is influenced by *PSB* of both networks
- *PSB* depends on latency, density and mining difficulty parameters

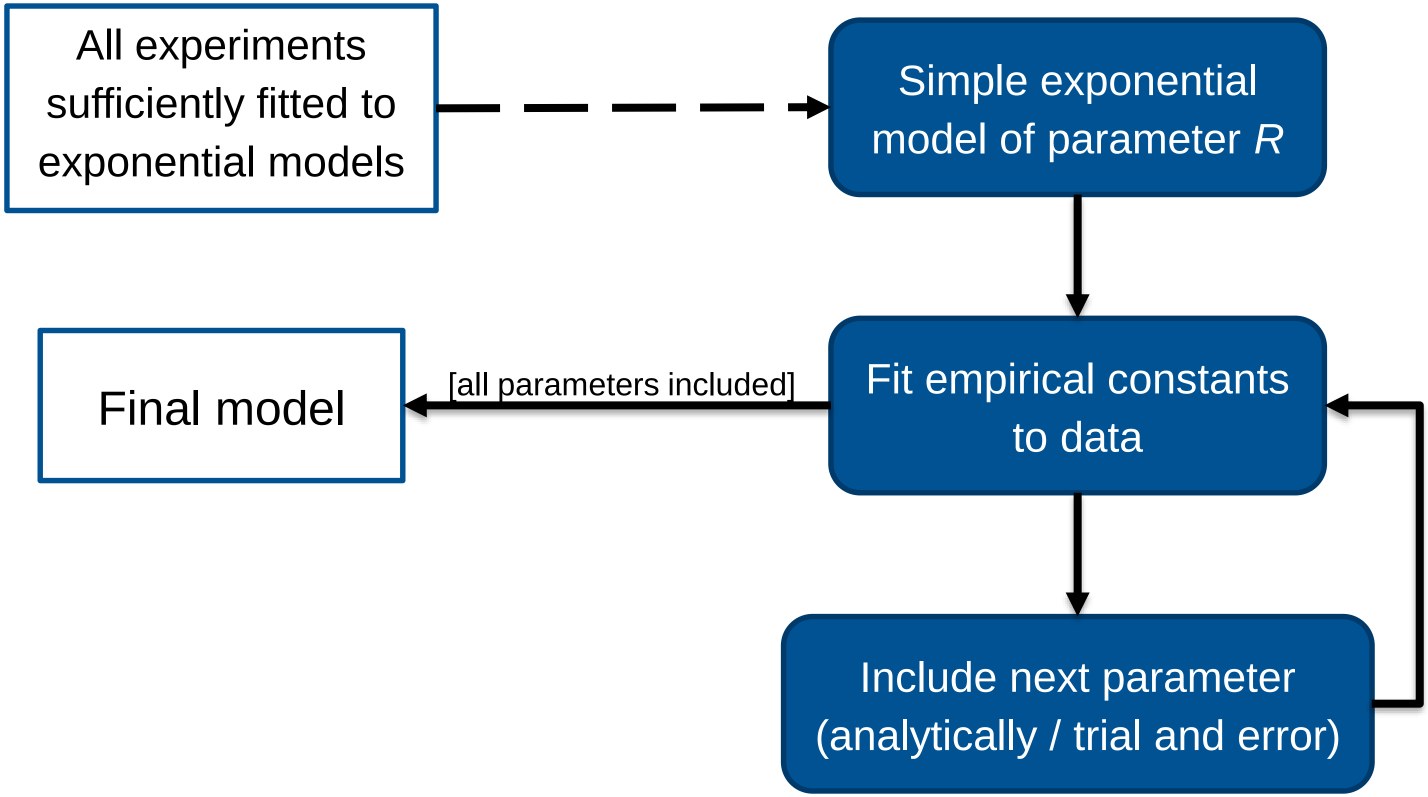
- Confirmations *C* successfully reduce *PDS* exponentially, as long as *effective* value of *R* less than 50%

Outline

1. Context
2. Blockchain Simulation
3. Analysis
4. Empirical Model
   – Building the Model
   – Model formula
5. Conclusion



Building the Model



Model

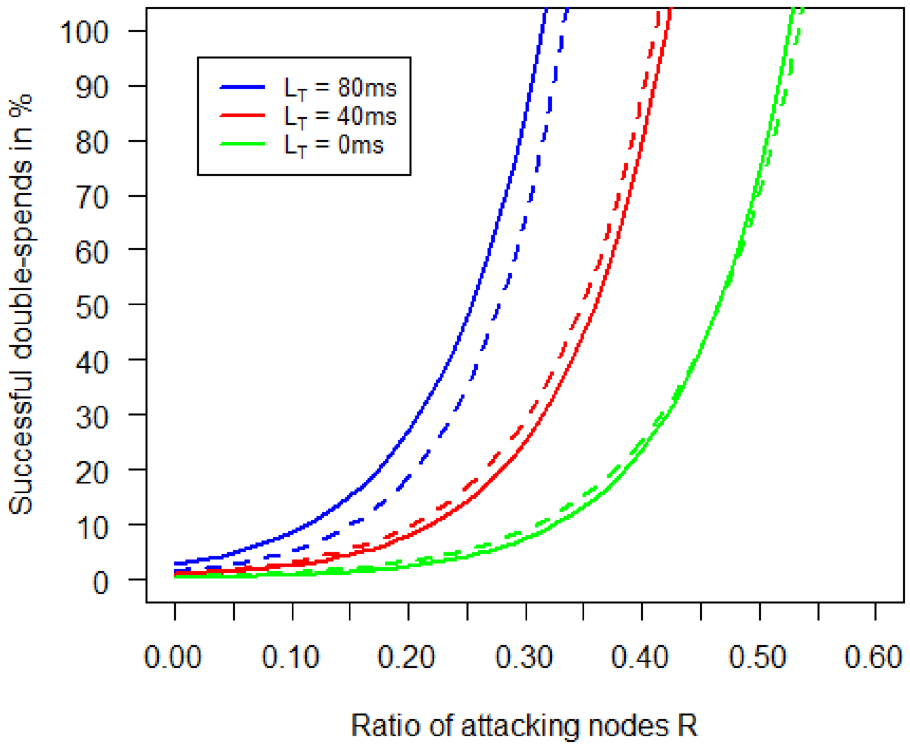
$$PDS = 100 \cdot exp\left(\left(R - \frac{T \cdot L_A}{D_A} + \frac{T \cdot L_T}{D_T}\right) \cdot C \cdot L_C\right)$$

- *PDS* indicative of RADS
- Empirical constants omitted
- Multiplied by 100 to receive percentage
- exp(x) corresponds to e^x

Model

$$PDS = 100 \cdot exp\left(\left(R - \boxed{\frac{T \cdot L_A}{D_A}} + \boxed{\frac{T \cdot L_T}{D_T}}\right) \cdot C \cdot L_C\right)$$

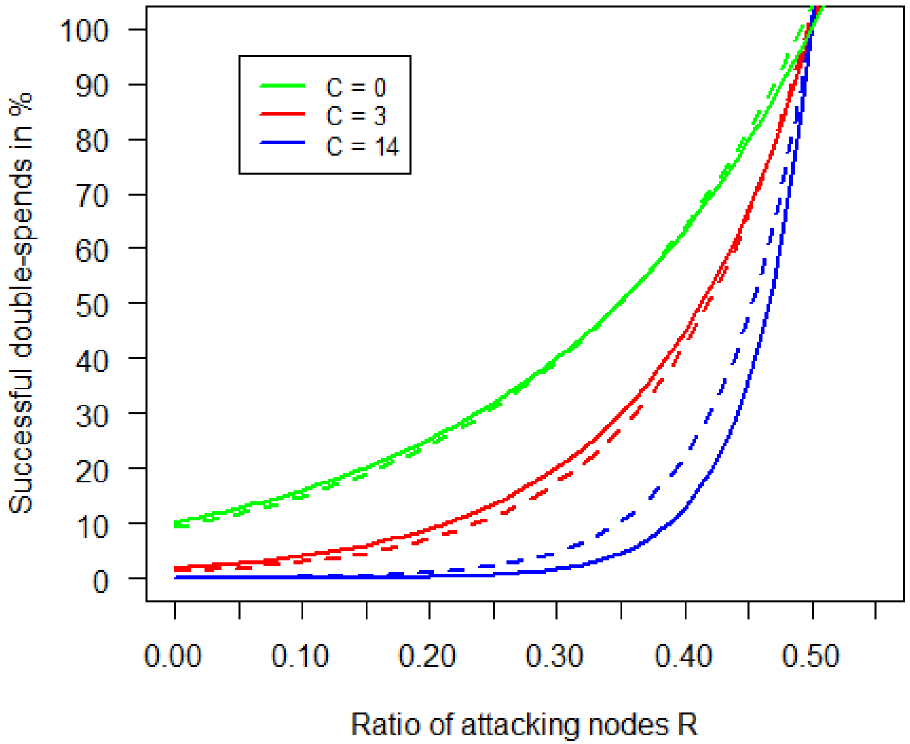
- Effects of latency are amplified by density and mining difficulty
- Computation of more stale blocks influences effective value of Nodes



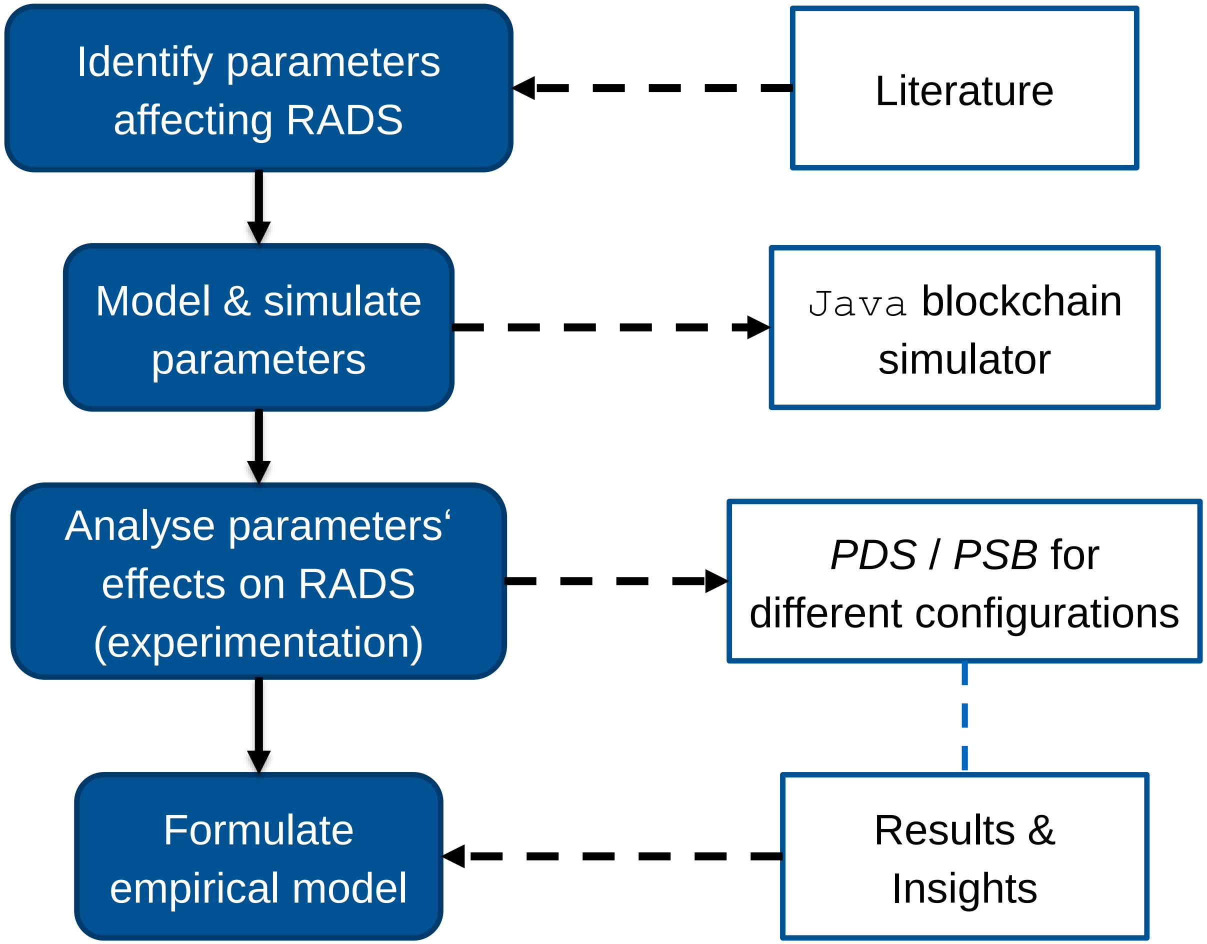
Model

$$PDS = 100 \cdot exp\left(\left(R - \frac{T \cdot L_A}{D_A} + \frac{T \cdot L_T}{D_T}\right) \cdot \boxed{C} \cdot \boxed{L_C}\right)$$

- C and L_C produce dampening effect
- No effect once majority of effective mining power controlled by attacker



Conclusion



Identify parameters affecting RADS
Literature
Model & simulate parameters
Java blockchain simulator
Analyse parameters' effects on RADS (experimentation)
PDS / PSB for different configurations
Formulate empirical model
Results & Insights

Conclusion

Additional findings:

- Definition of double-spend attack as 51% or majority attack is misleading
- Capability of conducting double-spend attacks depends on distribution of *effective* mining power
- Architectures with higher stale block rates are more vulnerable

Implication:

- Simulator and model can be used to predict a blockchain architecture‘s resistance against double-spend attacks

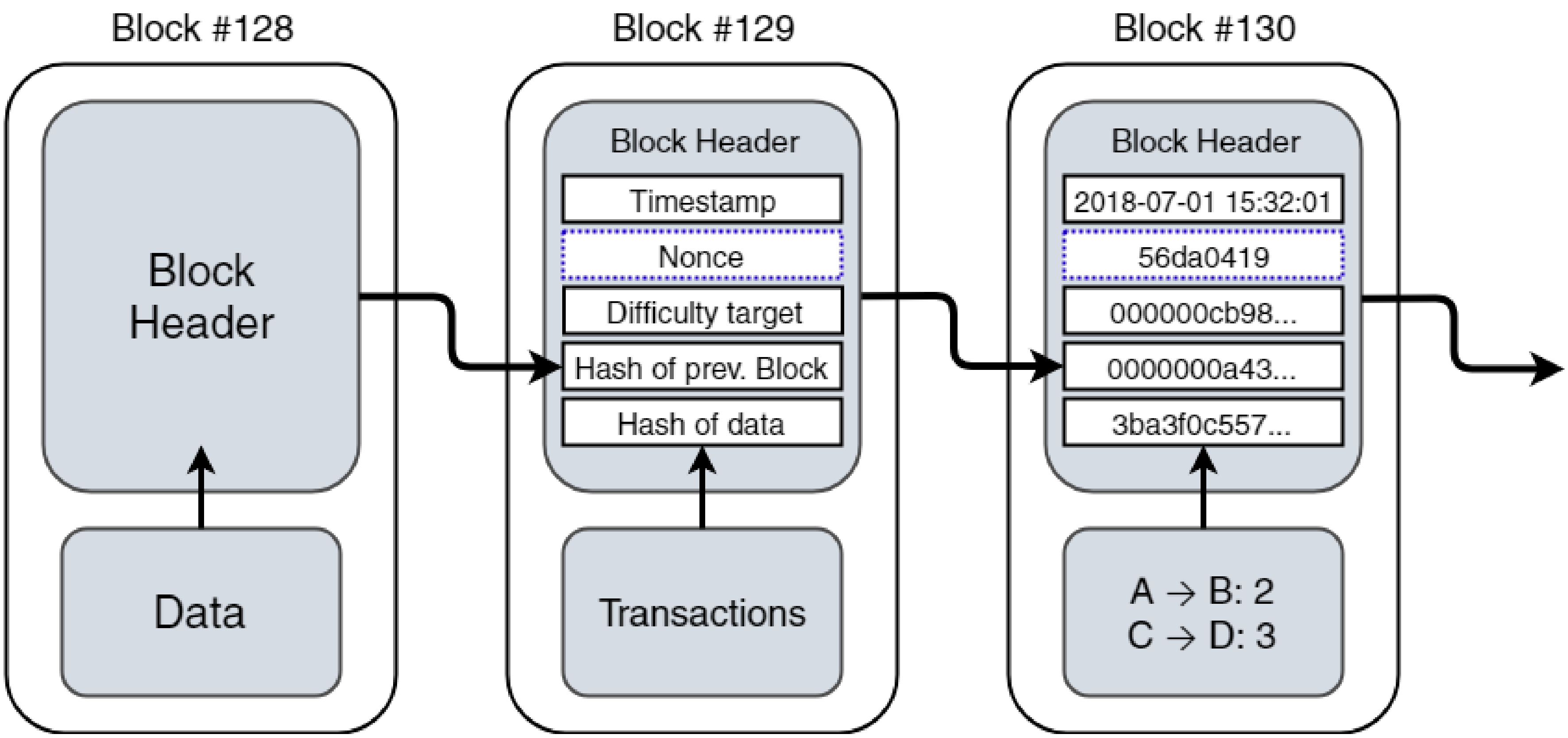
References

- Andreas M. Antonopoulos. *Mastering Bitcoin: Programming the Open Blockchain*. "O'Reilly Media, Inc.", 2017.
- C. Decker and R. Wattenhofer. Information propagation in the Bitcoin network. In *IEEE P2P 2013 Proceedings*, pages 1-10, 2013.
- Satoshi Nakamoto. Bitcoin: A peer-to-peer electronic cash system. 2008.
- Carlos Pinzon and Camilo Rocha. Double-spend attack models with time advantage for bitcoin. *Electronic Notes in Theoretical Computer Science*, 329:79-103, 2016. CLEI 2016 - The Latin American Computing Conference.
- Meni Rosenfeld. Analysis of hashrate-based double spending. *CoRR*, abs/1402.2009, 2014.

Thank you!

Backup

Blockchain Structure (Bitcoin)



Transaction Structure (Bitcoin)

Transaction 534				
Inputs			Outputs	
Transaction	Account	Value	Account	Value
198	A_0	1.70	B	2.00
432	A_1	0.26	A_0	0.10
258	A_1	0.16		
Transaction fee: 0.02				
Digitally signed by A_0 and A_1				

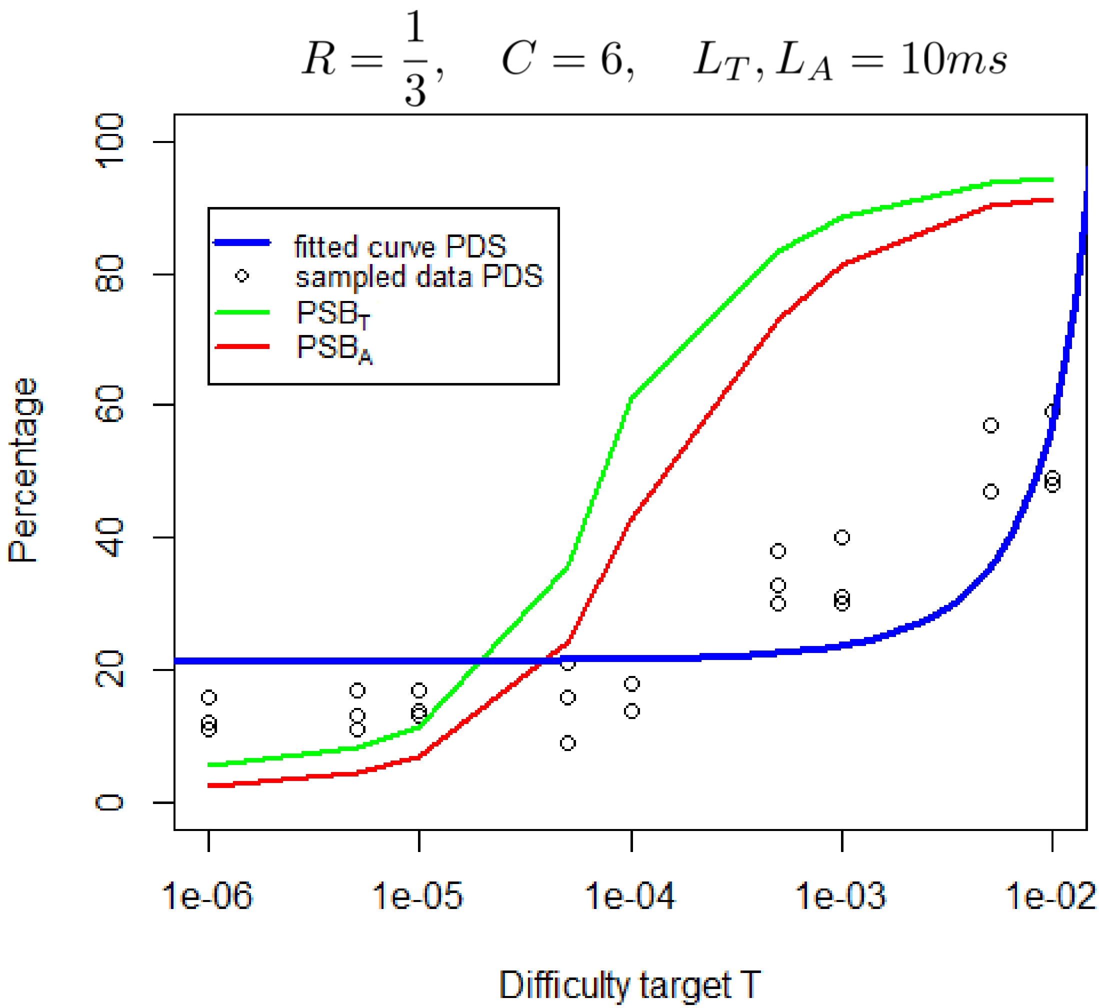
Transaction 817				
Inputs			Outputs	
Transaction	Account	Value	Account	Value
534	A_0	0.10	C	0.1
534	B	2.00	A_0	0.05
			B	1.95
Transaction fee: 0.0				
Digitally signed by A_0 and B				

Experiments

Difficulty target *T*

- High T increases the rate of new blocks

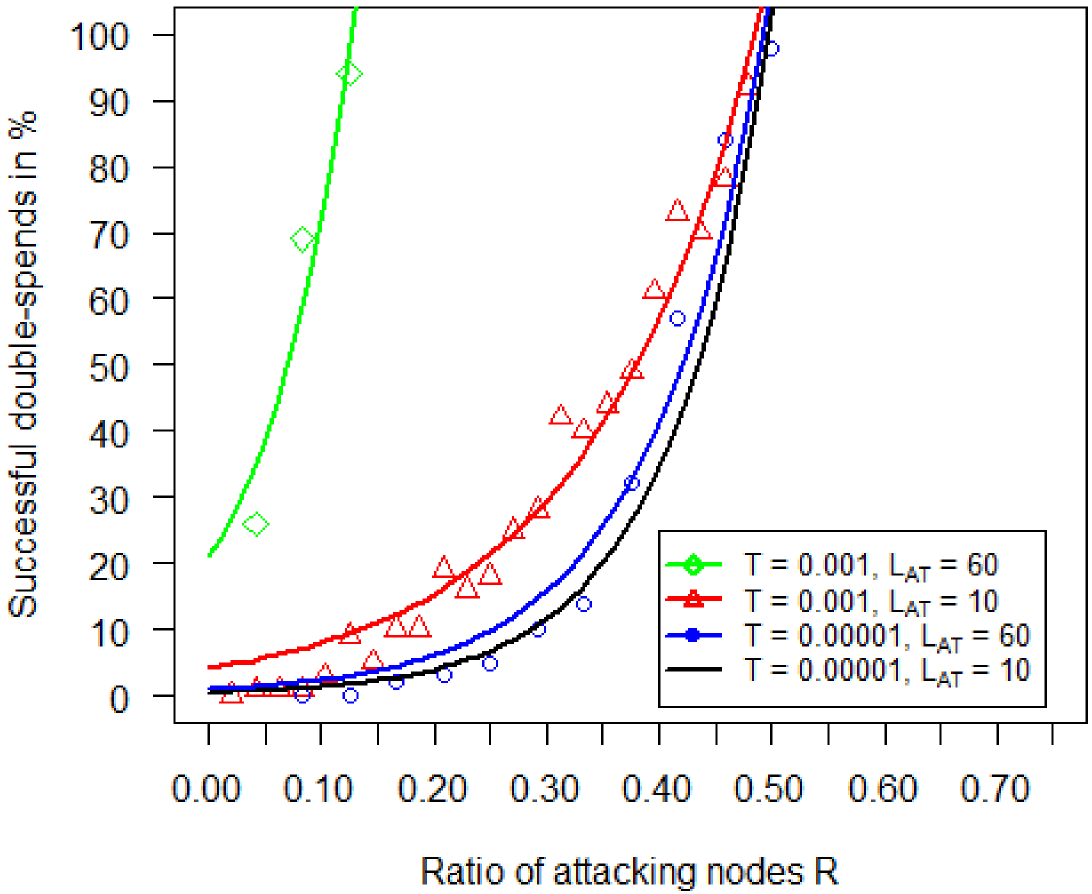
➢ Amplifies effect of network topology resulting in more stale blocks



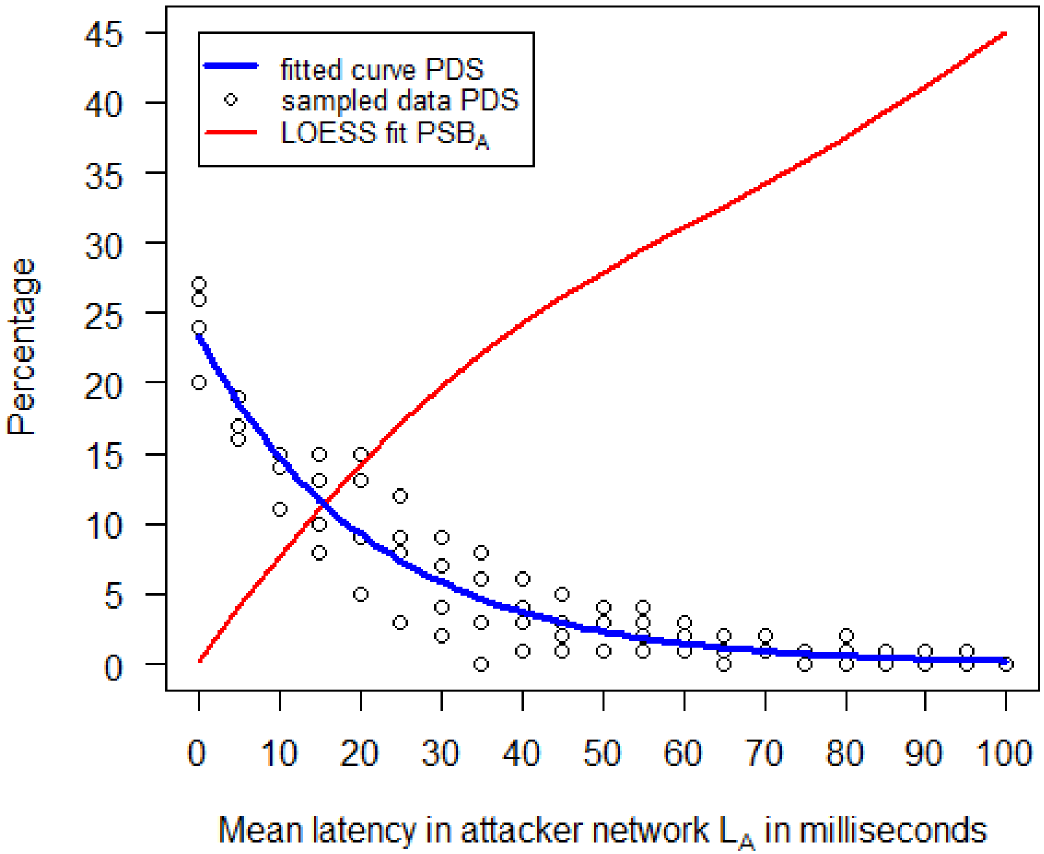
Difficulty target *T*

- T can be used to reduce effect of topology
- Low difficutly target creates more time between block creations

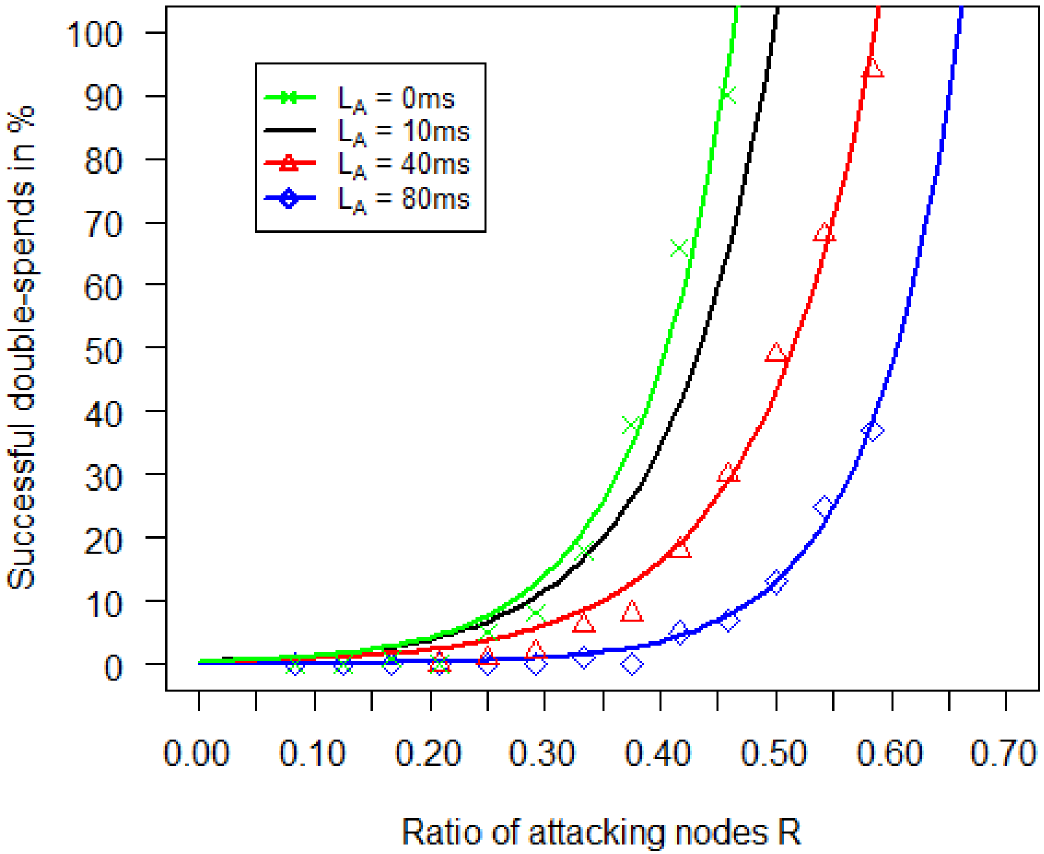
➢Less stale blocks even at higher latencies



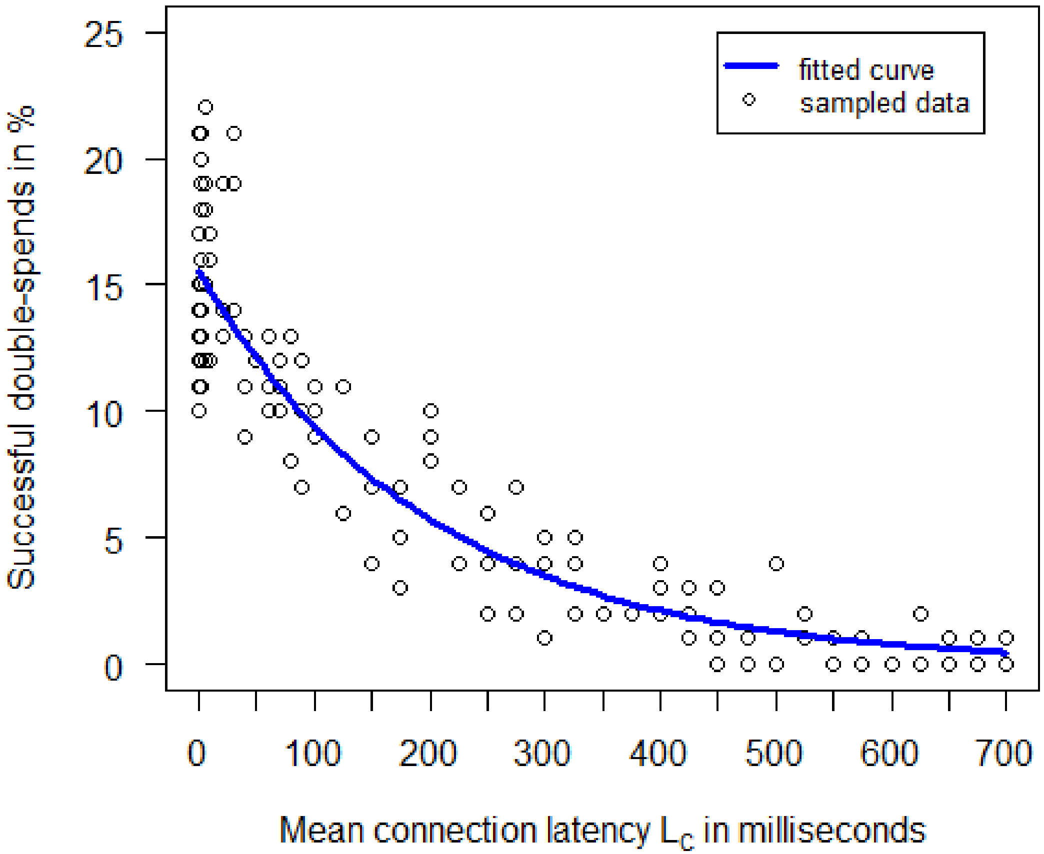
Attacker Latency L_A



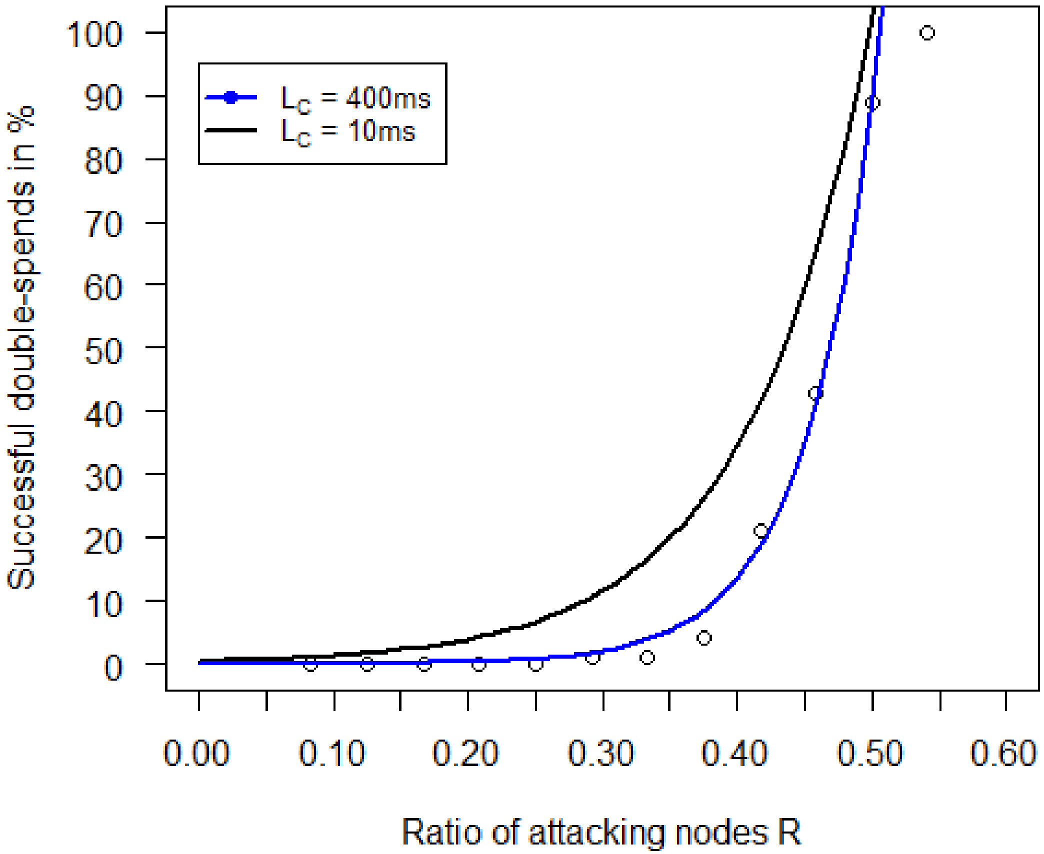
Attacker Latency L_A



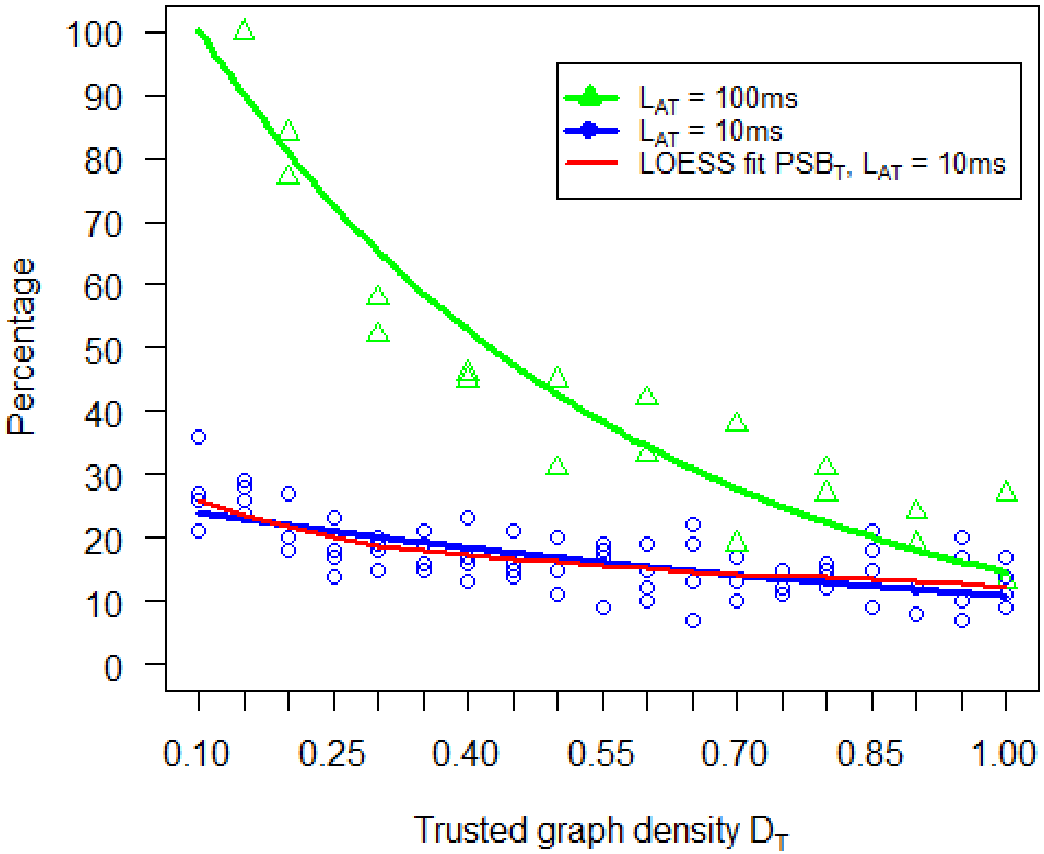
Connection Latency L_C



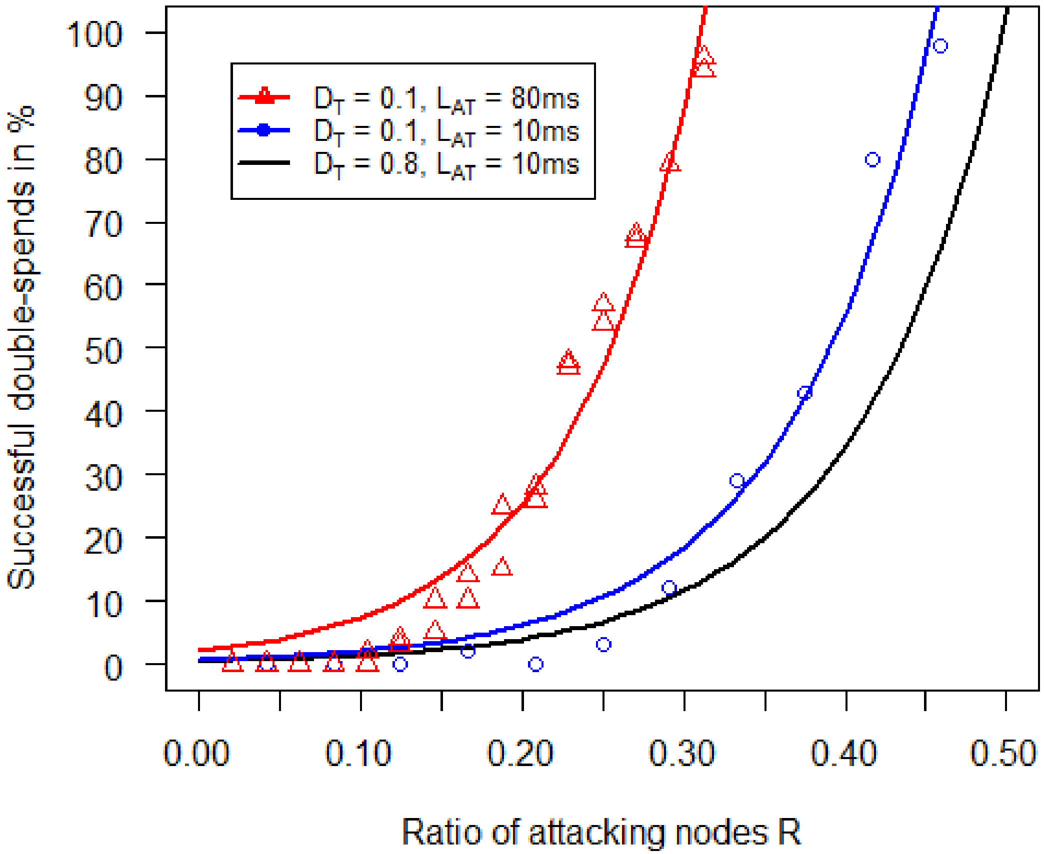
Connection Latency L_C



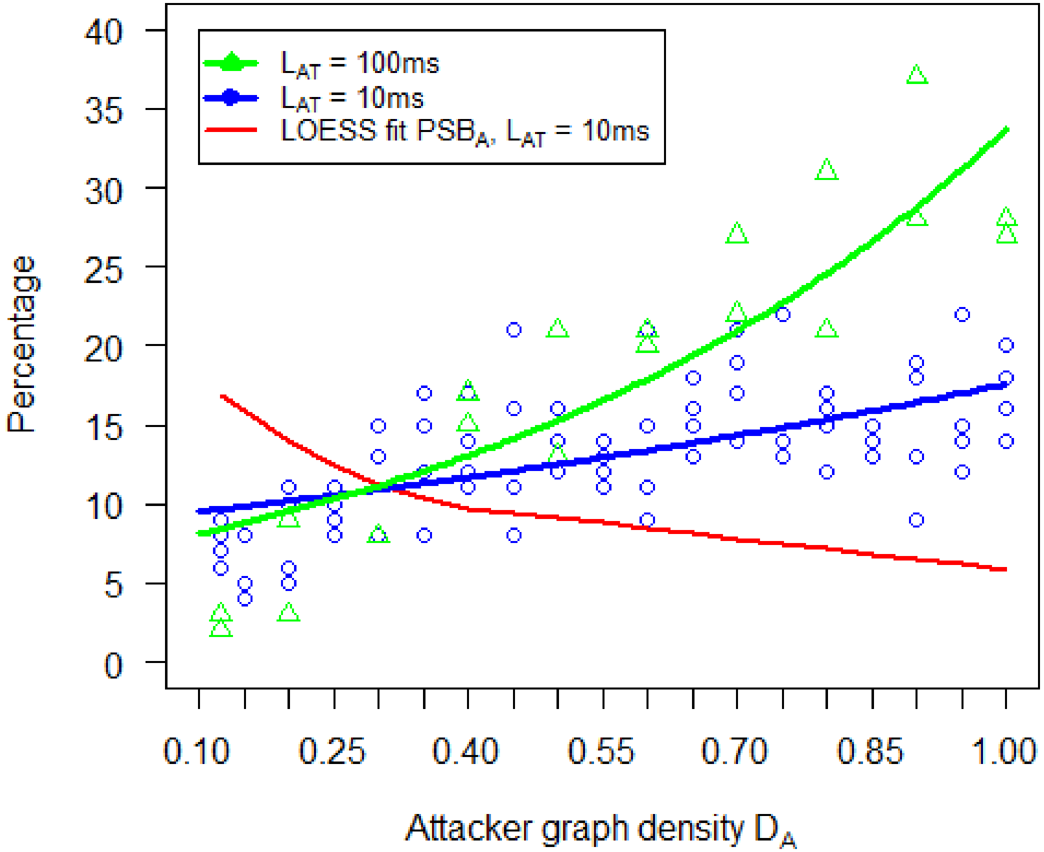
Trusted Density D_T



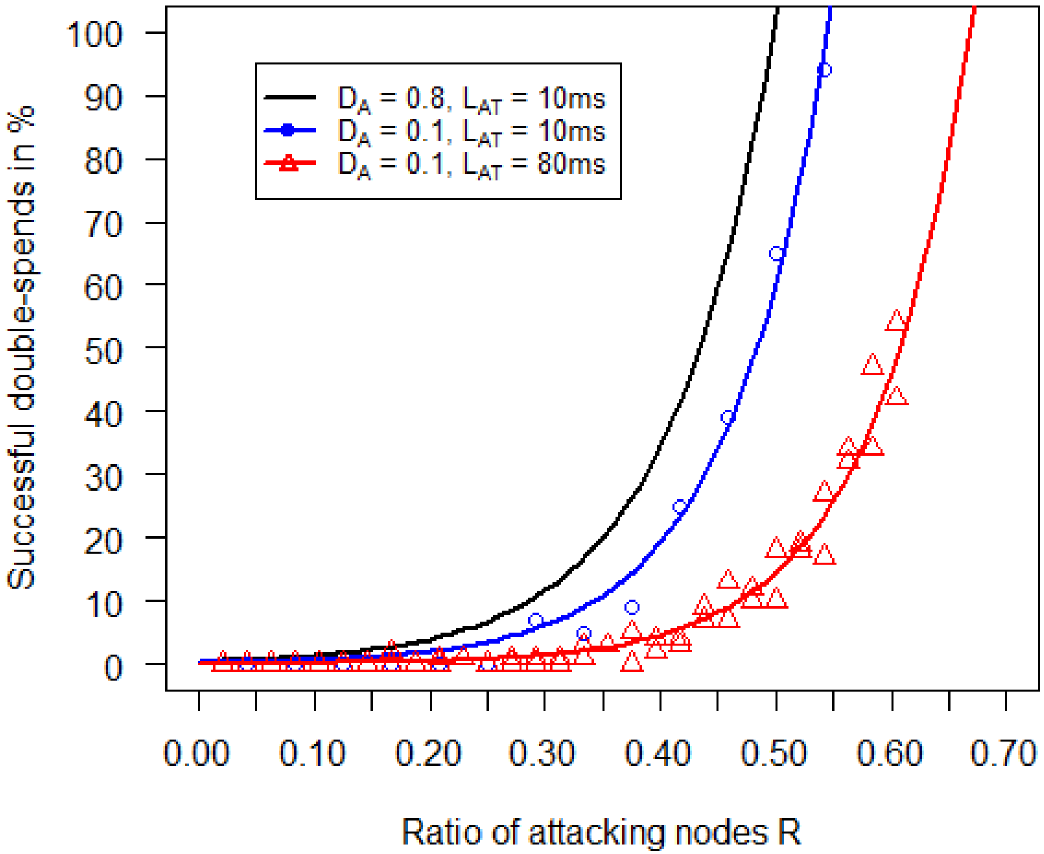
Trusted Density D_T



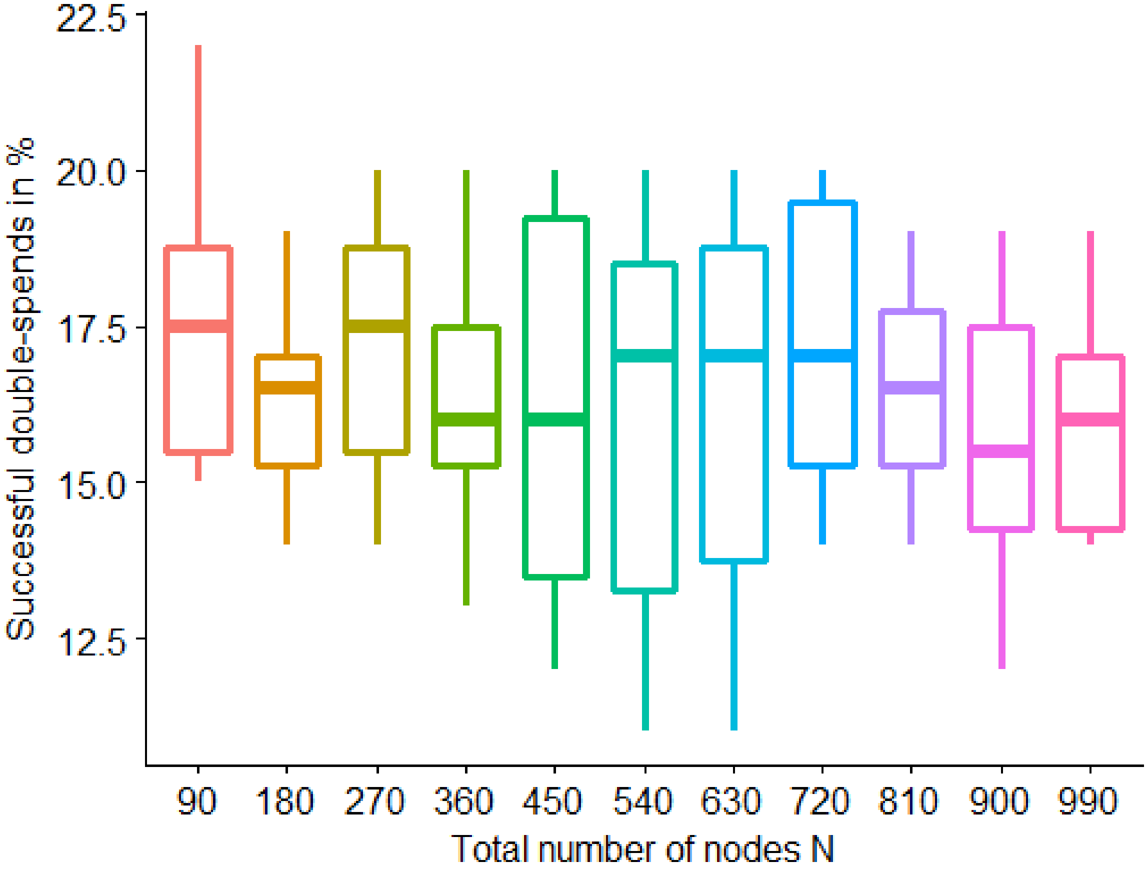
Attacker Density D_A



Attacker Density D_A

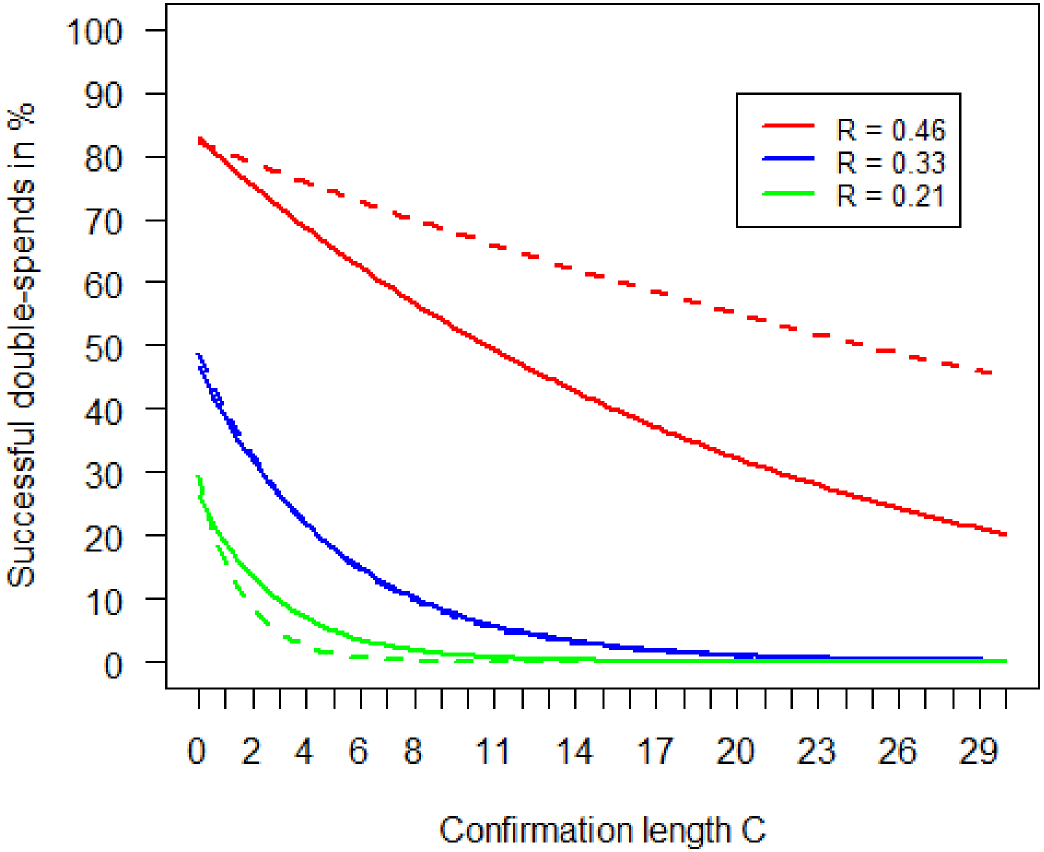


Number of Nodes *N*

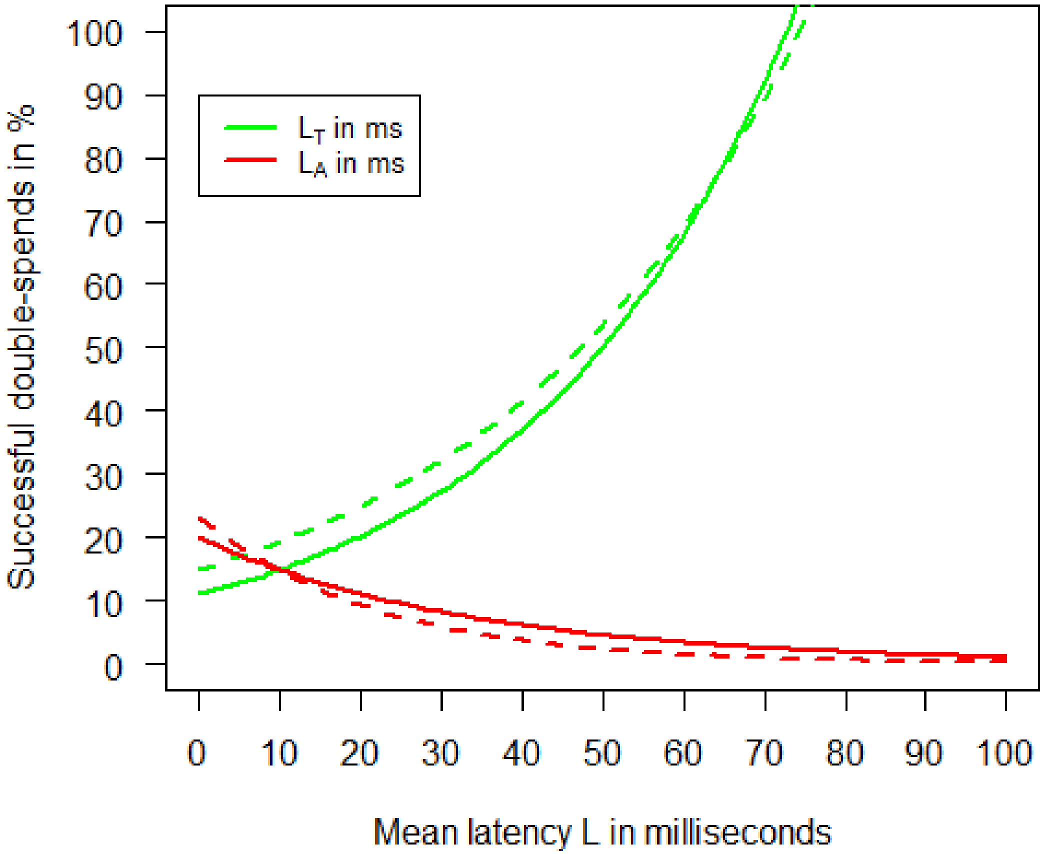


Comparison

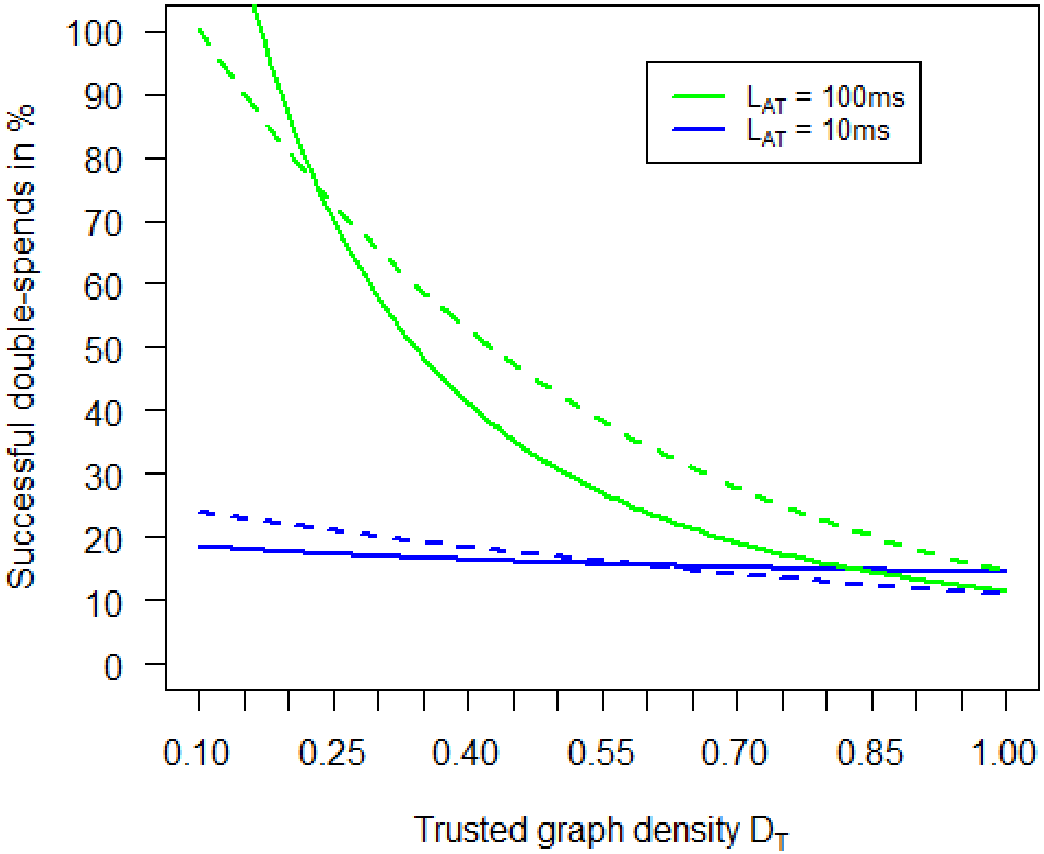
Confirmation length C



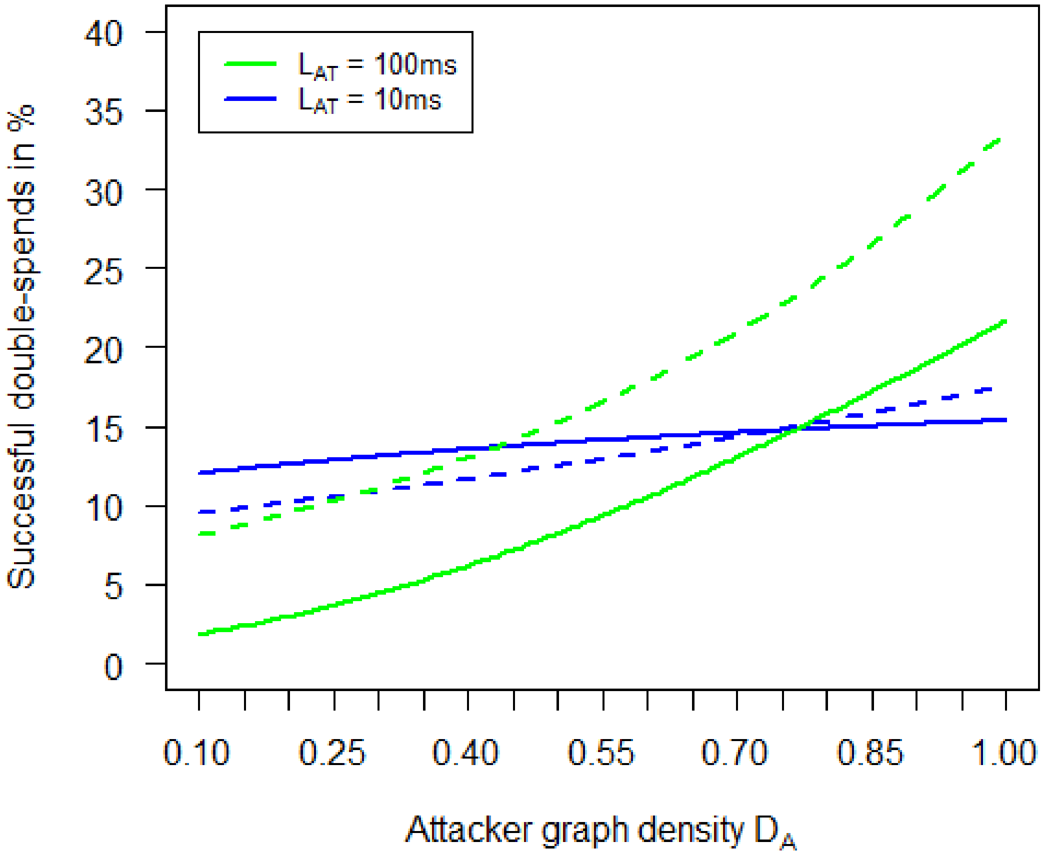
Latency L_T , L_A



Trusted Density D_T

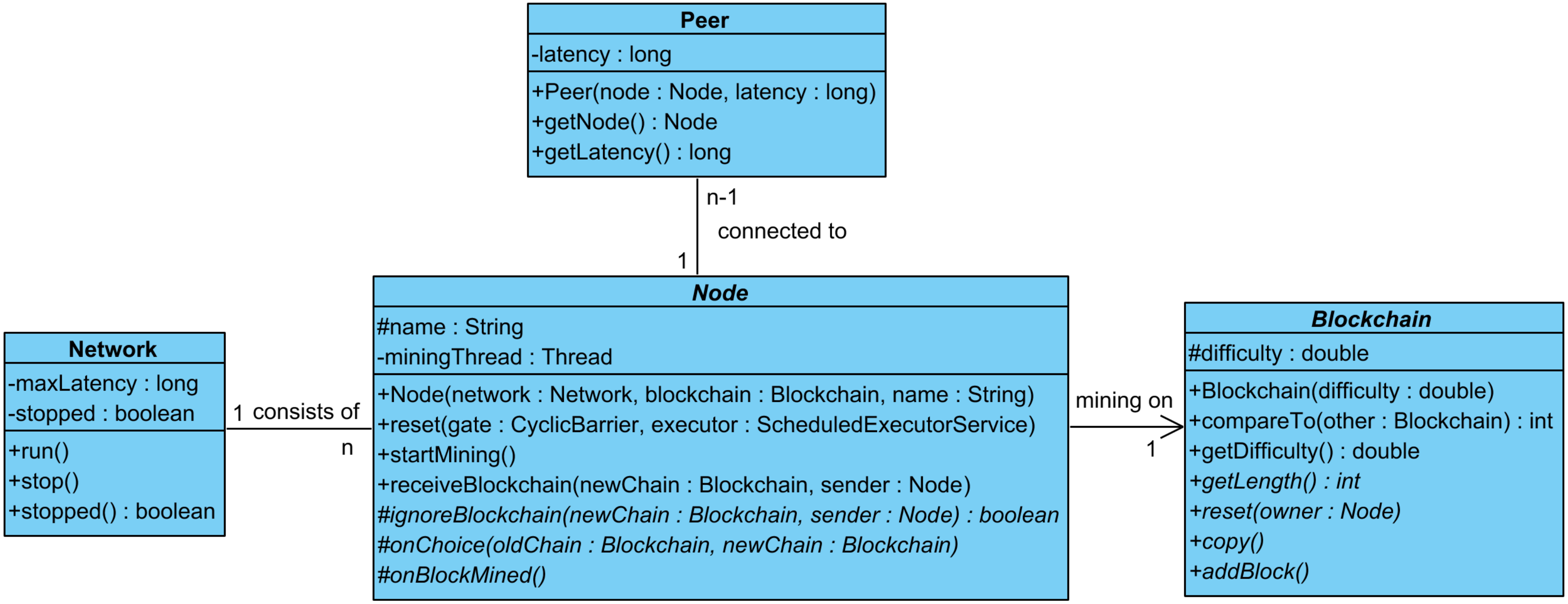


Attacker Density D_A

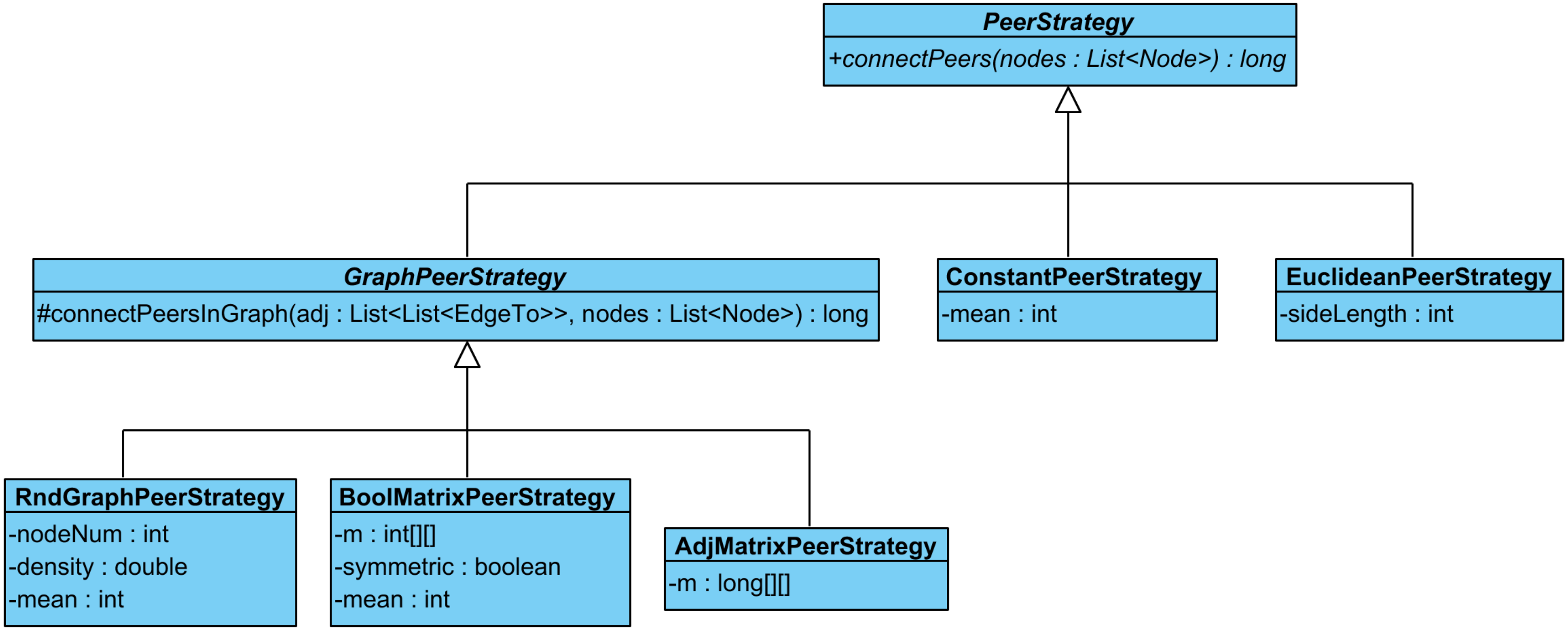


Simulator

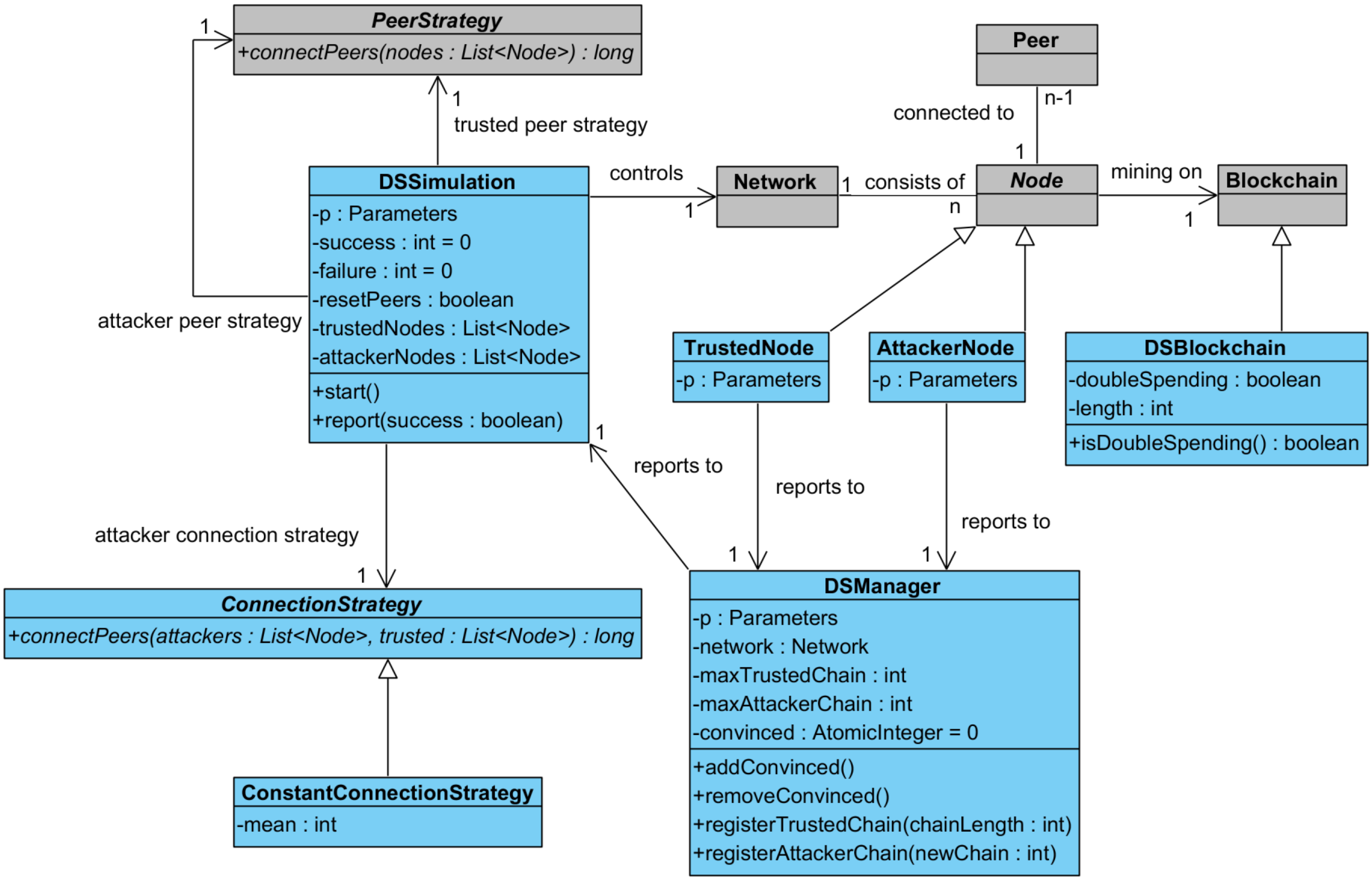
Simulator Framework



PeerStrategy



Double-Spend Simulator



Activity Diagram