Performance Analysis and Application of Mobile Blockchain

Kongrath Suankaewmanee[1], Dinh Thai Hoang[1], Dusit Niyato[1], Suttinee Sawadsitang[1], Ping Wang[1], and Zhu Han[2 1] School of Computer Science and Engineering, Nanyang Technological University, Singapore
[2] Department of Electrical and Computer Engineering, University of Houston, USA

International Conference on Computing, Networking and Communications (ICNC 2018)

중앙대학교 소프트웨어학과
20150249 김용재

Contents

Background of MobiChain

- **Mobile security** has become more and more important due to the boom of mobile commerce (m - commerce).
- Blockchain has been introduced as an **effective security solution** deployed successfully in many applications.

Background of MobiChain

- Mining process usually executed on **powerful devices** with high computational capacities and energy supply. (Server and Computers)
- **Improvement of mobile technologies and hashing algorithm** allow the mining process to implement on mobile devices efficiently.
- **MobiChain** allows the mining process to be performed on mobile devices effectively.

System Model and Assumptions

Mobile node

- Each mobile node has a **backlog and queue to store pending tasks.**
- The mobile node is connected to a server node either via **the Internet or a local direct connection.**

Server node

- All server nodes are installed **Sync Gateway to broadcast data.**
- Server nodes are always connected to other server nodes via the Internet.

* Sync Gateway : Maintains up-to-date copies of documents where user need them. On mobile devices for instant access and on servers in data centers for reasons such as synchronizing documents, sharing documents, and loss-protection and so on.

System Model and Assumptions

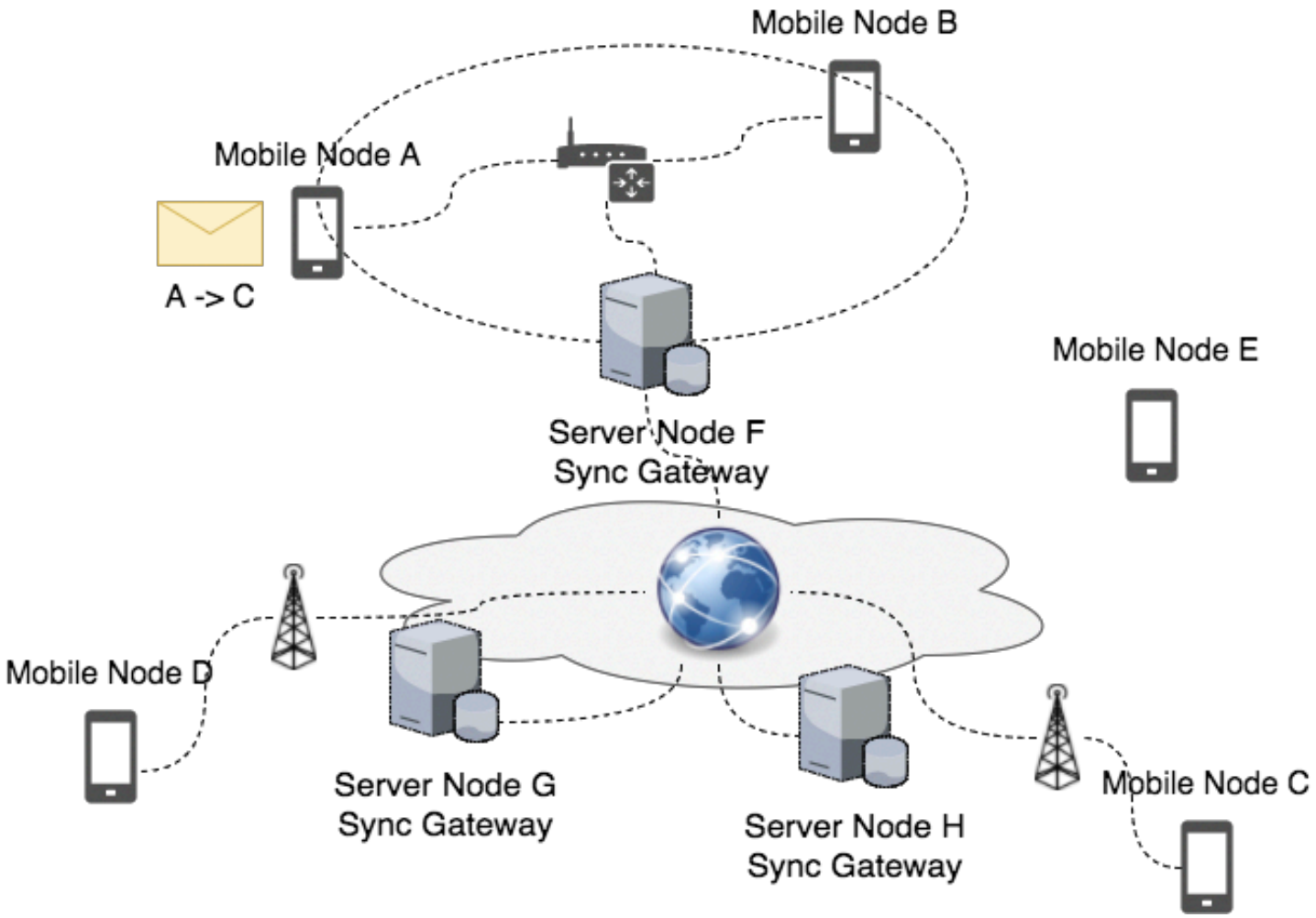


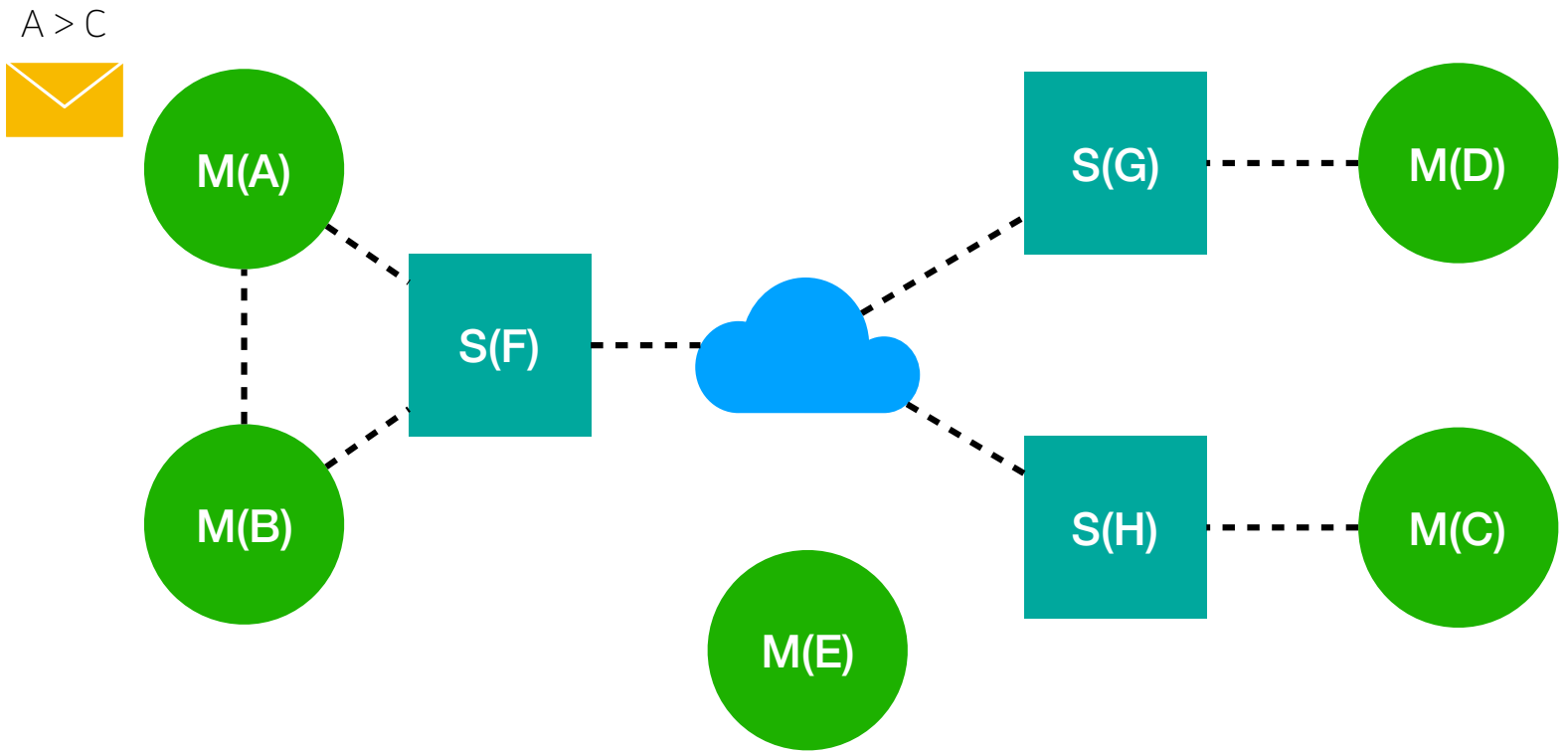
Figure 1 : The MobiChain system model.

Message Passing Process

- Mobile Node A sends a message to Mobile Node C.
- This message can be a data transfer request or a radio resource sharing request, **which needs to be verified. (transaction)**
- All mobile nodes are installed the MobiChain application.
- And nodes have already registered user accounts.
- These accounts are to provide public and private keys which are unique to the users.

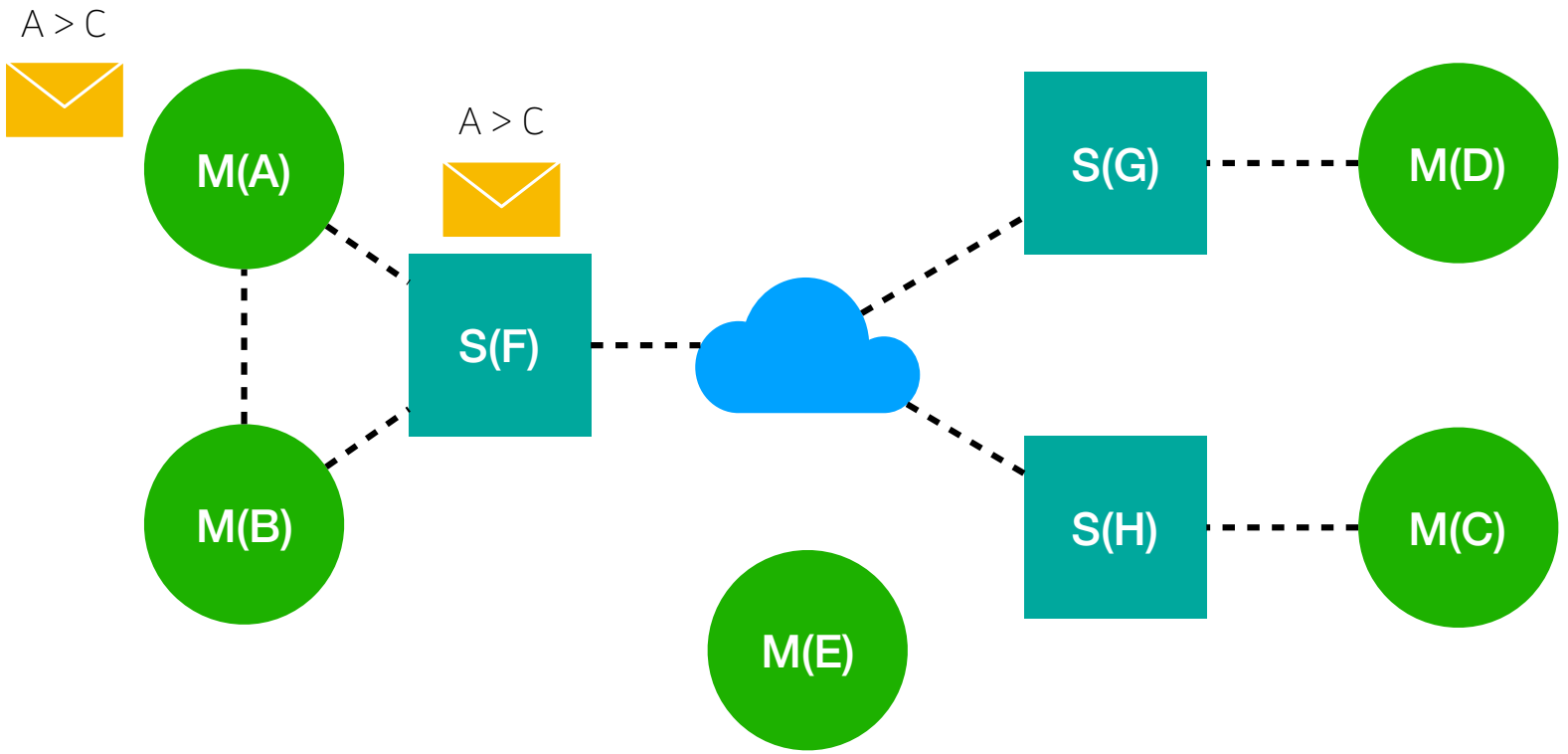
Message Passing Process

Step 1. Mobile Node A create a transaction. After that, the sender puts the transaction into the sender's backlog.



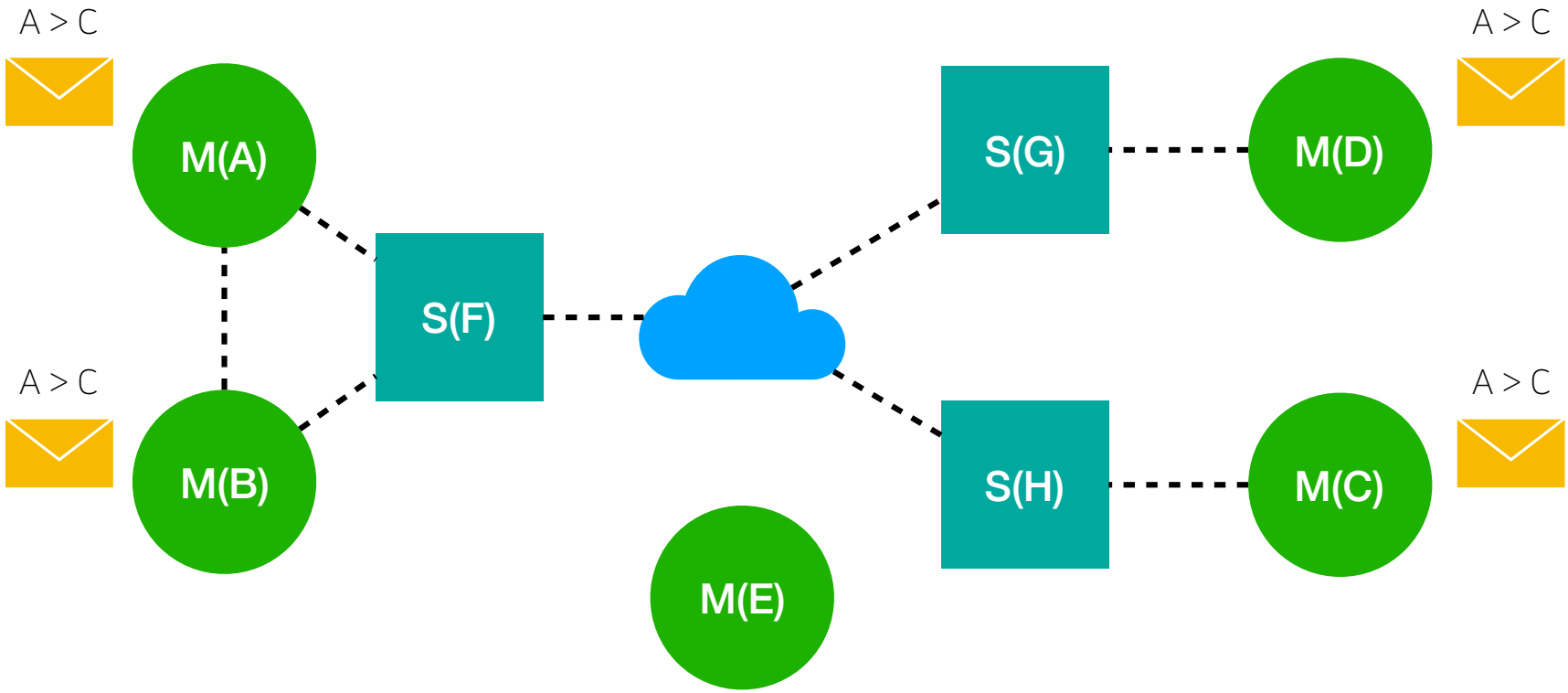
Message Passing Process

Step 2. The sender uploads the transaction to its connected server, i.e., Server node F.



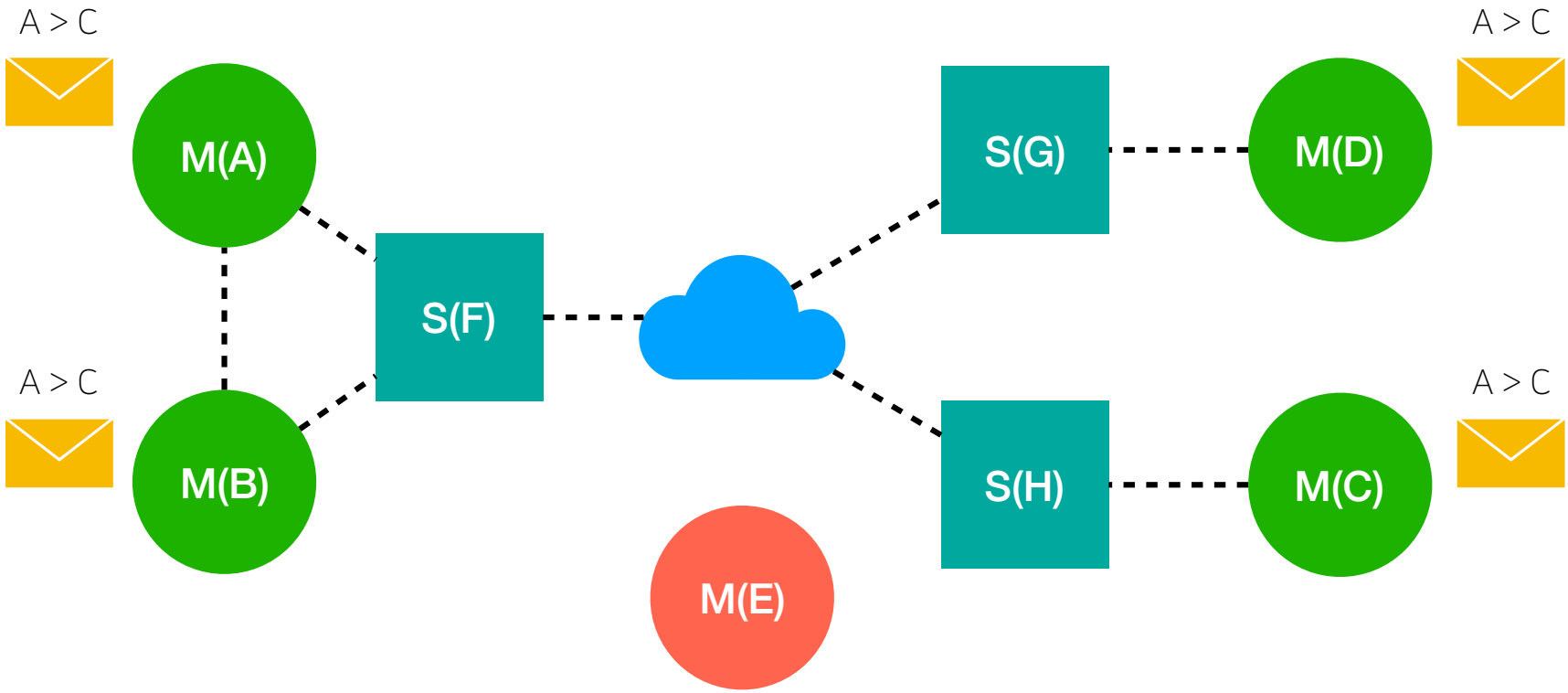
Message Passing Process

Step 3. The Sync Gateway broadcasts the transaction to every node in the network.



Message Passing Process

Mobile Node E does not have the same transaction because it is currently not connected to the network.



Mining Process

- Mobile Node B and C are dedicated or want to mine this transaction.

Step 1-1. A miner queries the transaction from its backlog, and verifies this transaction by checking

(1) whether the transaction is modified or not. **and**

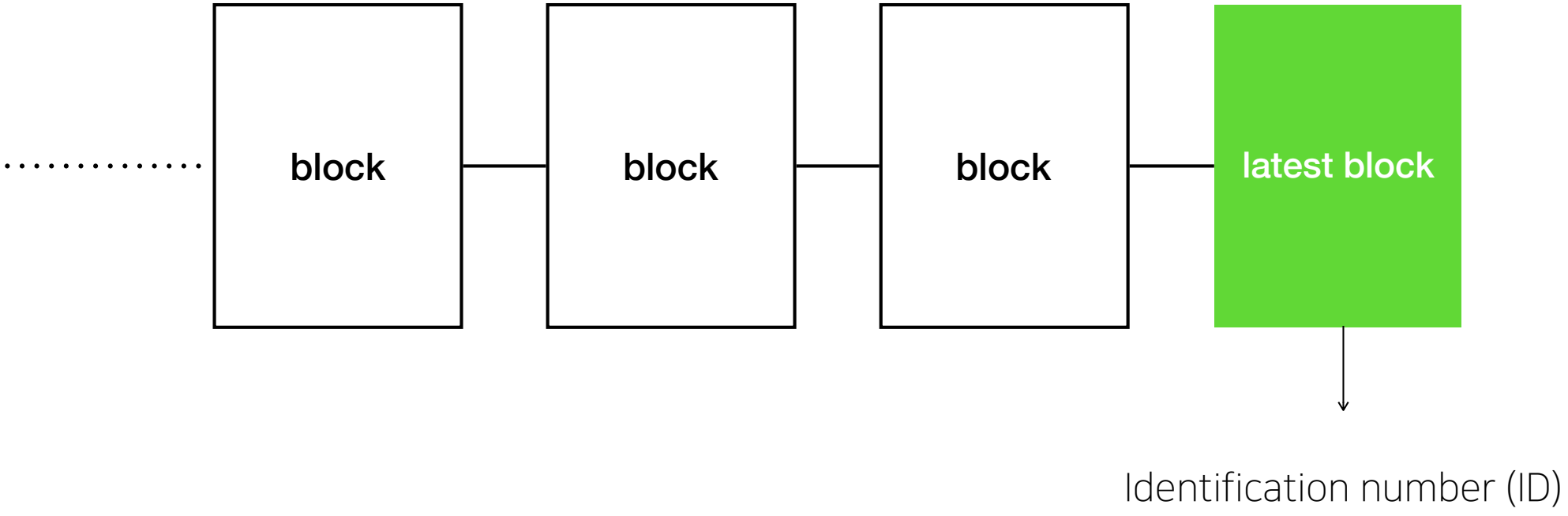
(2) whether the transaction exists in the blockchain or not.

Step 1-2. If the transaction is not modified and does not exist in the blockchain, the miner perform **Step 2.**

Otherwise, the mining process is terminated and the miner reports the problem to all other nodes in the network.

Mining Process

Step 2. The miner finds the latest block in the chain, and store the identification number (ID) of the latest block for the future use. **if the chain is empty, the previous block ID is set to be zero.**



Mining Process

Step 3. This step is referred to as Proof-of-Work (PoW) process. In the PoW process, the miner will create a new block by hashing the information iteratively.

Step 4. This step is to verify all information. This process is to guarantee that the information of all blocks in the chain has not been changed.

Step 5. The miner needs to check whether the created block has been added to the chain by other mobile nodes or not. If not, the miner will add the created block to the chain.

> Almost same as usual block chain mining process.

Implementation of MobiChain

- By using the proposed core module, all mining processes of the blockchain technology can be executed on Android devices.
- The function of the core module are to
  - create blocks
  - verify the correctness through mining processes
  - link the verified blocks to the chain.
- The core module requires an application to work as a front-end.
- In fact, the core module can be used in various applications, for example, **file sharing, smart contracts,** and **credit member systems.**

Database Function

- The main functions of chain are to **record the private data in the local device** and to **broadcast this chain to all connected devices in the network.**
- The chain includes three different data structures.
    1) account
    2) transaction
    3) block
- In MobiChain, The data structure is followed by the JSON format.
- The private key is stored locally at the mobile node, while the public key is broadcast to other nodes.

Database Function

```
{
 "type": "account",
 "username": /*String of username*/,
 "private_key": /*String of private key's account*/,
 "public_key": /*String of public key's account*/
 "create_date": /*Date time of creating*/
}
```

Account data structure

```
{
 "id": /*Result string after hashing everything inside
     transaction excluding signature */,
 "signature": /*String of the combination between
     transaction and private key's sender*/,
 "timestamp": /*Time of creating*/,
 "transaction": {
  "data": {
   "payload": /*Any string in JSON format*/,
   "uuid": /*String of the unique identification number*/
  },
  "owner": [/*String of public key's sender*/, /*String
     of public key's destination*/]
 }
}
```

Transaction data structure

```
{
 "id": /*Result string after hashing block_number,
     tx_hash, previous_block, and nonce*/,
 "block_number": /*Integer of the current block number*/,
 "votes":[
    {
     "node_pubkey": /*String of public key's miner*/,
     "signature": /*Result string after vote is signed by
      using private key's miner*/,
     "vote": {
      "is_block_valid": /*Boolean that present the block
     valid status*/,
      "previous_block": /*String ID of the previous block
     */,
      "timestamp": /*Time of block creating*/,
      "voting_for_block": /*Same with the ID*/
     }
    }
   ],
 "version": "1",
 "tx_hash": /*Result string after hashing  all
     transactions in the block*/,
 "block": {
    "transactions": [/*list of transactions*/],
    "voters": [/*list public key's of voters*/]
 },
 "nonce": /*Integer of the hashing time. Note that
     hashing is done iteratively until the conditions are
      met*/
}
```

Block data structure

Database Function

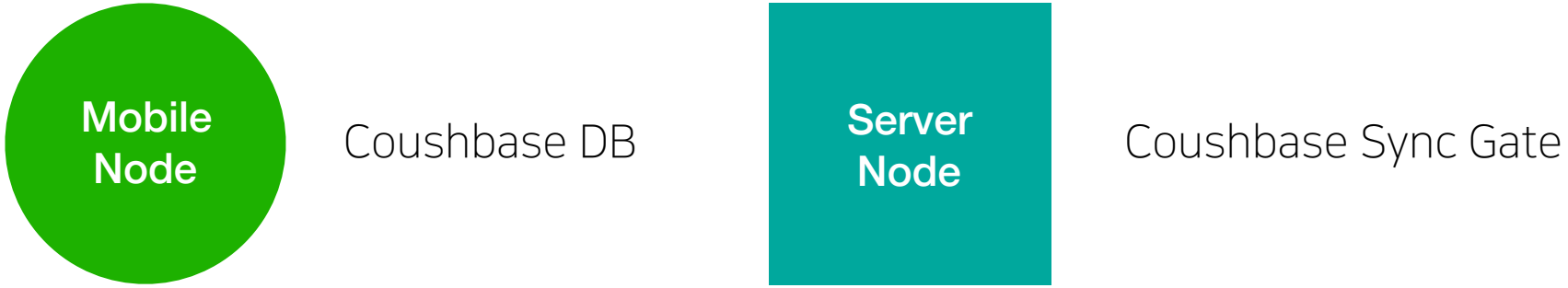
- In the MobiChain system, the blockchain is stored in a database. The database is implemented on both mobile devices and servers.

Mobile Node

- Coushbase Lite database instead of SQLite **(NoSQL)**
- It is **suitable for a real-time system.**

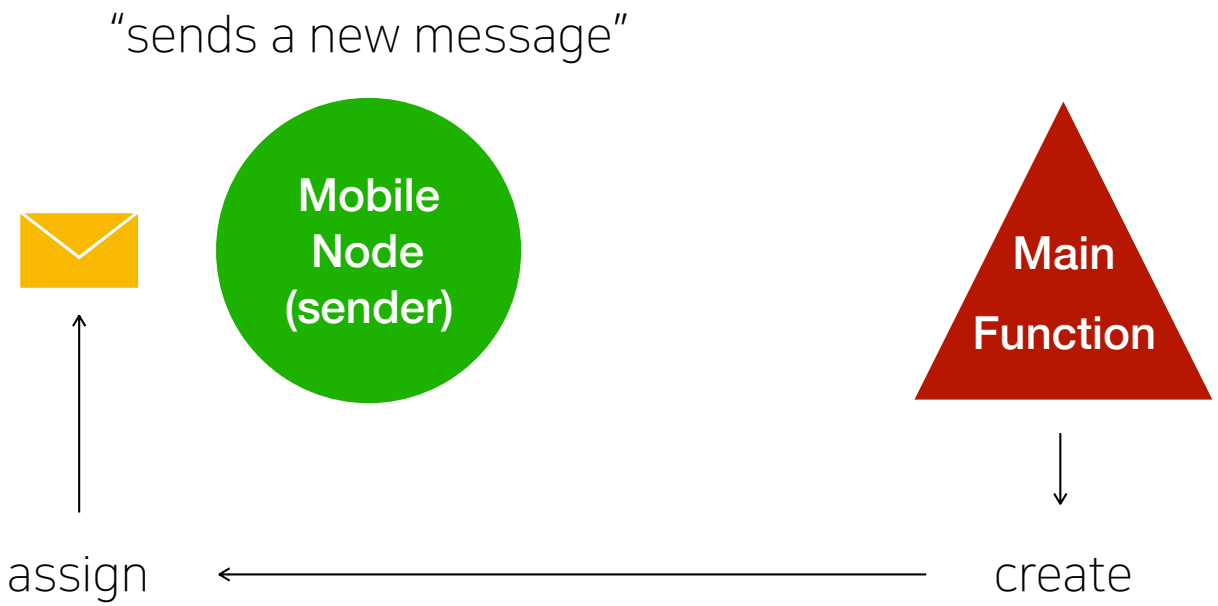
Server Node

- Coushbase Sync Gate is implemented to receive and broadcast data to the devices.



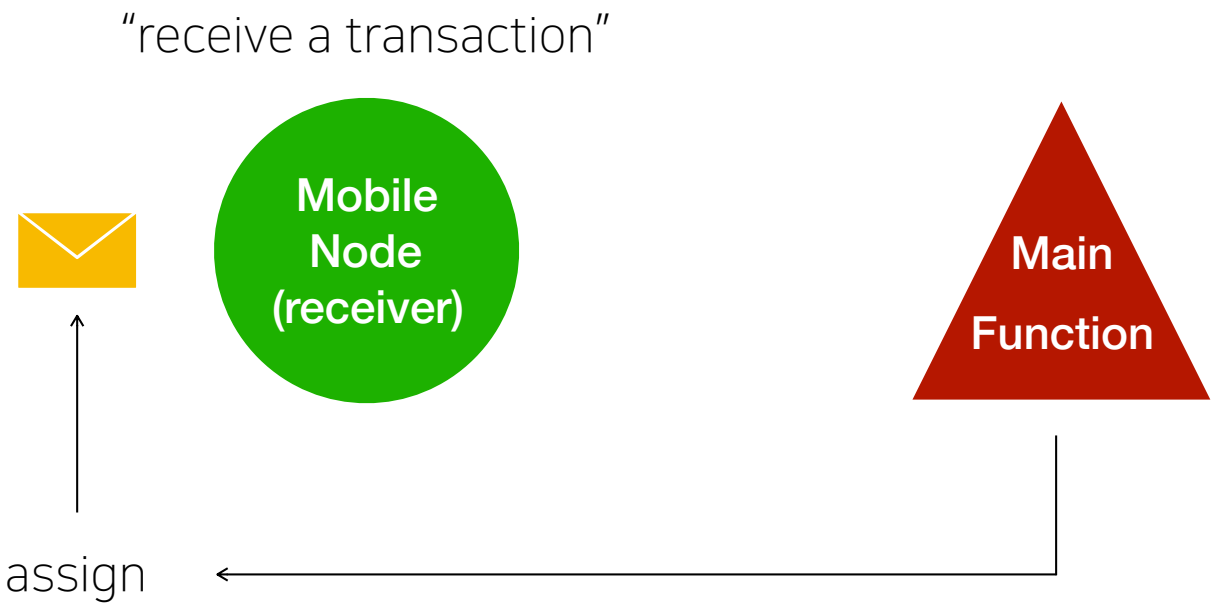
Main Function

1. When a mobile node sends a new message, the Main Function creates a transaction and assigns the transaction to the **sender's backlog**.



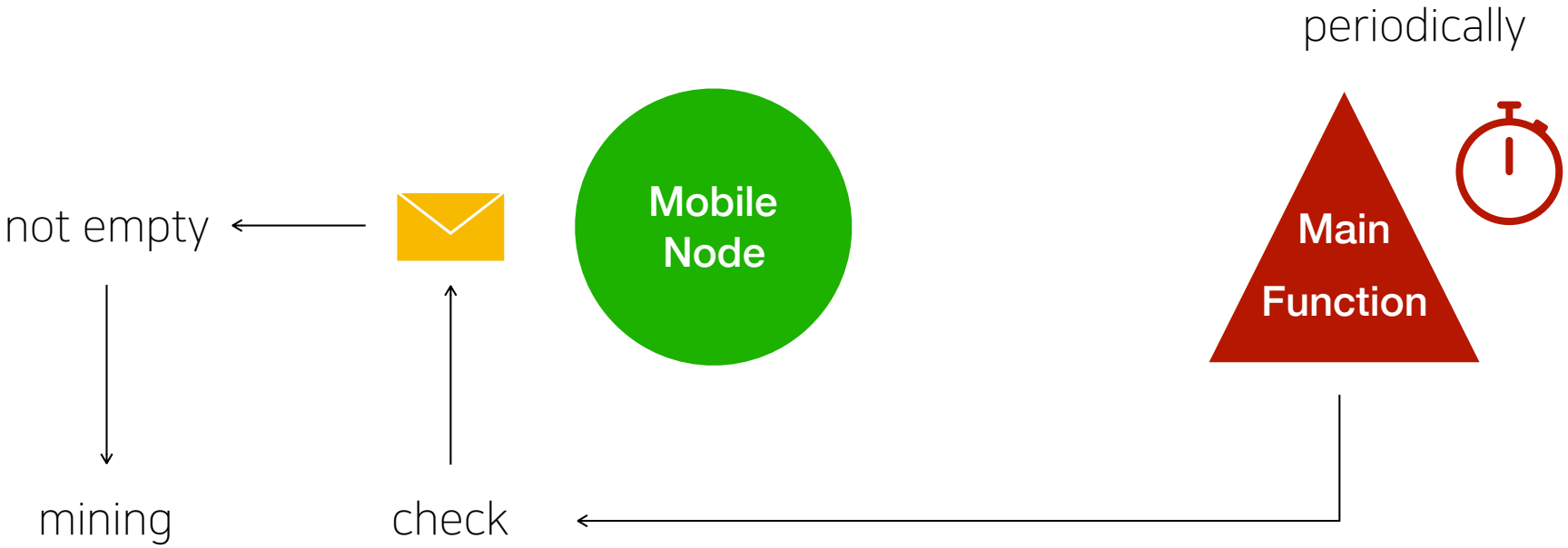
Main Function

2. When a mobile node receives a transaction, the Main Function assigns the new transaction to the **receiver's backlog.**



Main Function

3. The Main Function is executed periodically to **check whether a mobile node's backlog is empty or not.** If the backlog is **not empty**, the main function will **perform the mining process.**



Cryptography Function

- The cryptography function can be separated into three parts.
  - **cryptography-hashes** : SHA3-256 algorithm
  - **key-signature** : ED25519 public key signature system
  - **encode-decode** : Base58 schemes

Performance Evaluation

Experiment environment

- Mobile device (Mobile Node) : Samsung Galaxy Tab S2 8.0 (T715)
- Server (Sync Gateway) : Workstation with Intel Xeon CPU E5-1630
- Total energy consumption on the mobile device was measured by VideoOptimizer program
- If the nonce is not specific in an experiment, the nonce is set to be zero.

Memory Utilization

- The content of each transaction is fixed at 20 characters.

Test Case

1) 1 transaction per block
2) 3 transaction per block
3) 6 transaction per block

Test Result

- If we **increase the number of transactions in each block**, **the memory utilization can be reduced** remarkably.
- If we store 3 or 6 transaction in one block, the memory utilization can be reduced by 33% or 55%, respectively.

$$Memory\ \ Utilization\ =\ c_b\ +\ c_tT\ +\ c_dD$$

T = number of transactions in one block

D = number of digits of ***block_number***

Memory Utilization

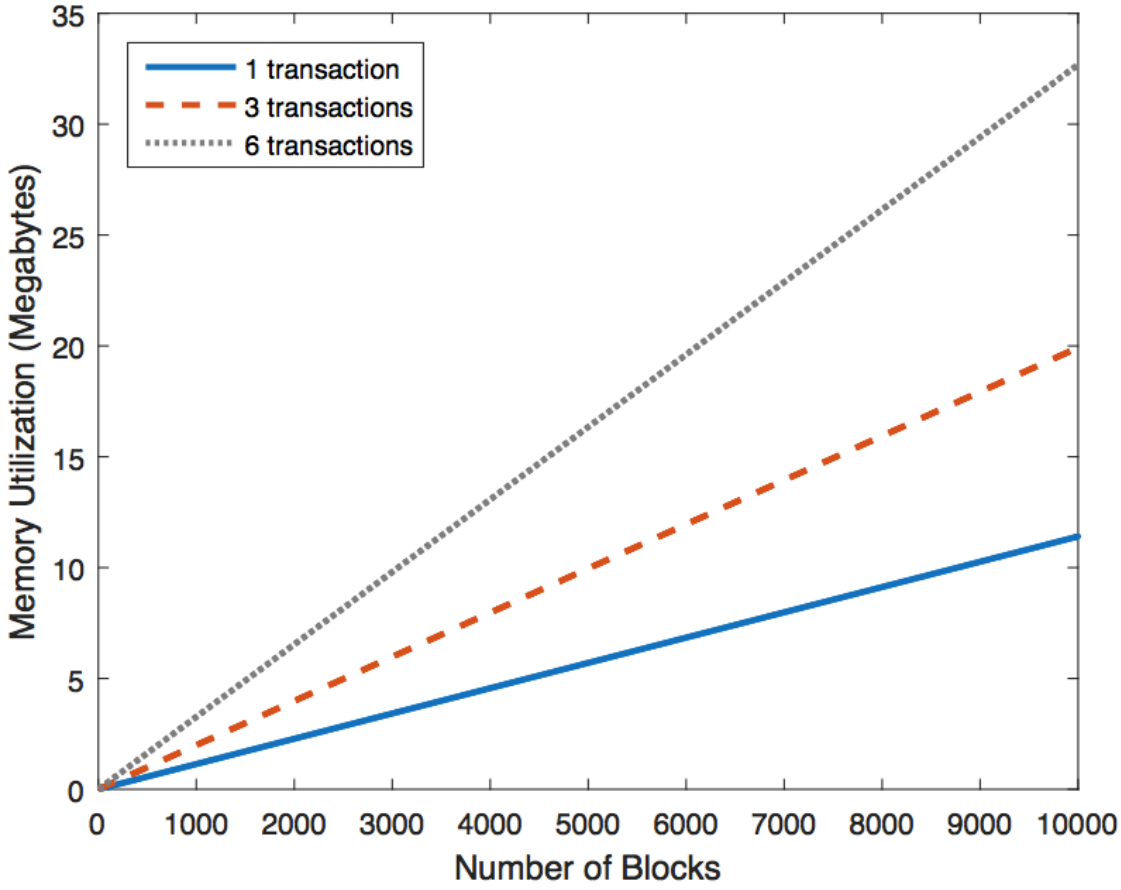


Figure 6: The memory utilization when the number of blocks increases.

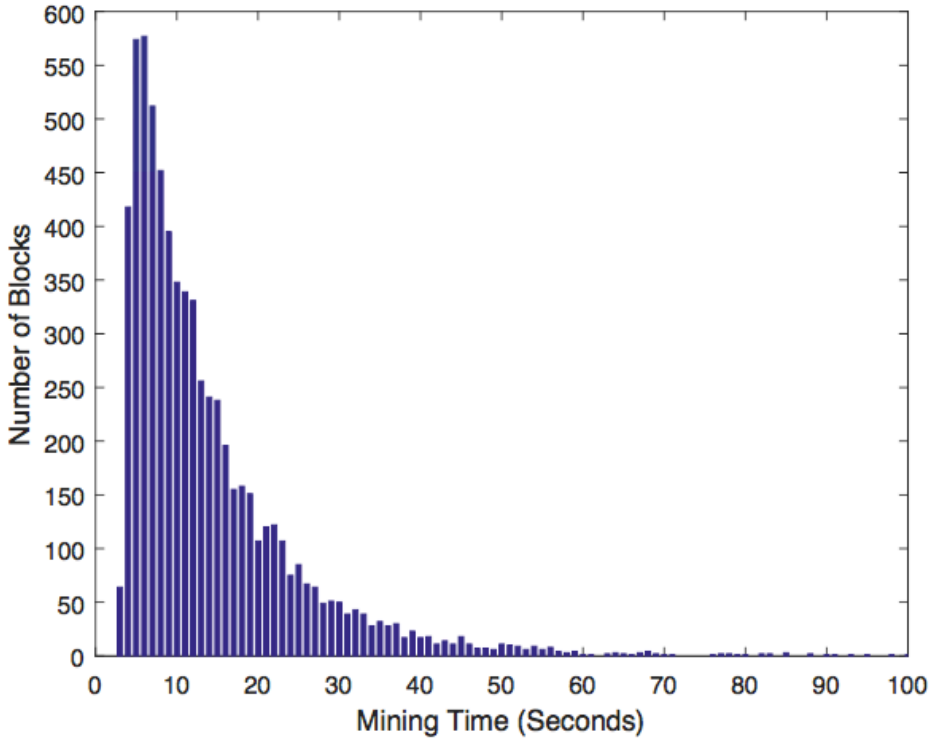
Proof-of-Work Process

- The hash process is executed iteratively until **the first three digits of the hash value equal zero.**

Test Condition

- 7,156 blocks were used for this test.
- These blocks were mined using mobile device.

Proof-of-Work Process



Test Result

- The test result is filtered to show only 0 to 100 seconds.
- **88.06% of blocks** need to **use 3 to 30 secs** to perform the PoW process.
- Only 4.79% of blocks perform longer than 100 secs.
- At the peak points, 23.23% of total blocks use 5 to 7 secs.
- 803 hashing iterations are executed per second.
- Peak points use around 4,015 to 5,621 hashing iterations before meeting the condition.

Chain Verification Process

- The execution time and energy consumption are measured from the beginning of the chain verification processes **until the end of this process.**
- **For multiple threads**, the measurement is from the beginning **until the last thread completes.**

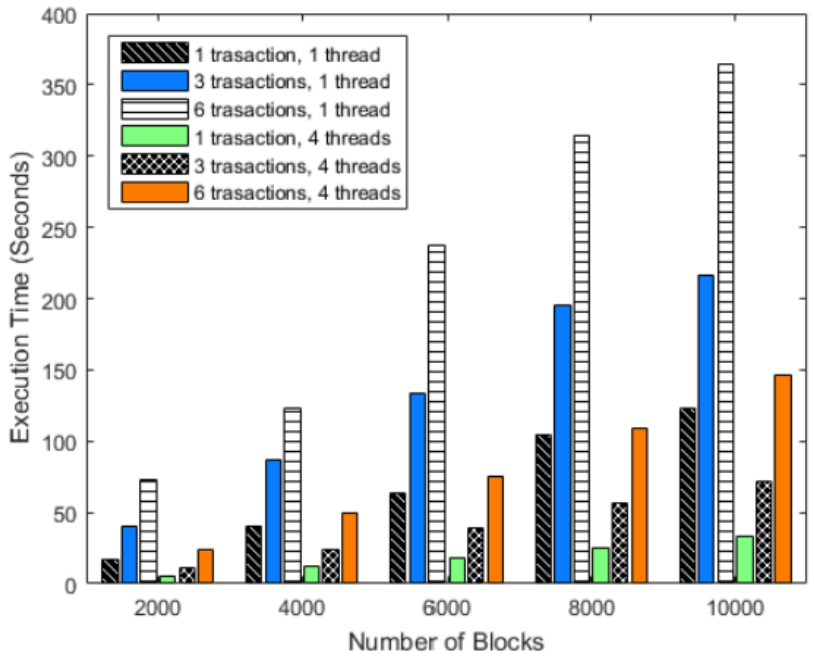
Test Case

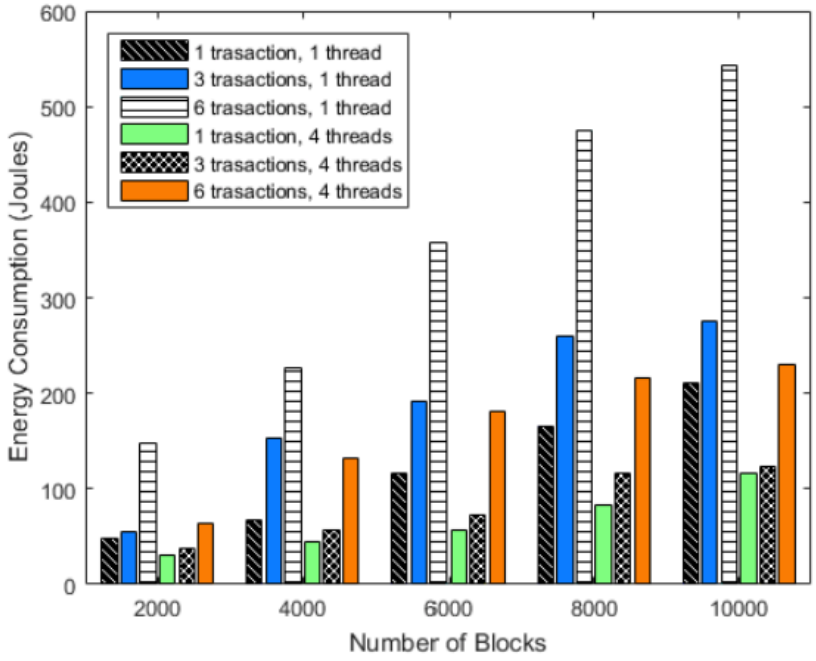
1) 1 thread
   - 1 transaction per block
   - 3 transaction per block
   - 6 transaction per block
2) 4 thread
   - same as above

Chain Verification Process

Test Result

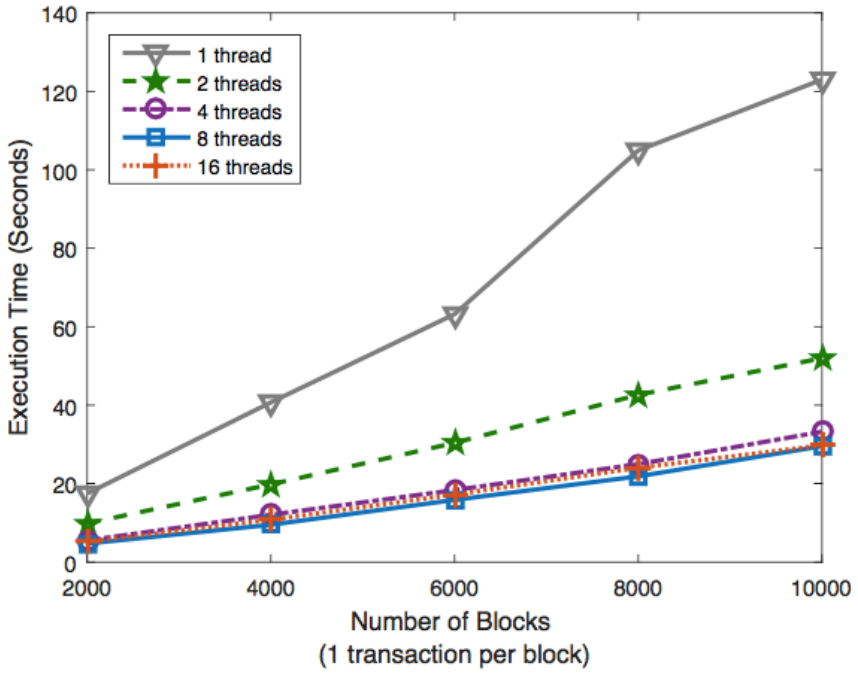
- As the **number of blocks in the chain increases, the execution time and energy consumption increase** accordingly.
- **Transaction are grouped together** in a block -> **faster / less energy.**
- In practice, having more number of transactions in a block can cause more delay if the transactions are generated randomly.





Chain Verification Process

- When we increase the number of threads, the execution time is not always reduced.
- If we keep increasing the number of threads, the execution time reduces insignificantly.
- Android device support 8 processing cores and each core has one thread.



Conclusion

- MobiChain, a new m-commerce application using blockchain technology for data security.
- It can perform mining process on mobile devices through.
- Experiments show that blockchain tech is a practical solution for
  - security
  - efficiency
  - scalability
  - processing…
- MobiChain system will be extended for **offline mining** and **propose data synchronization algorithms when mobile nodes are reconnected** to the network.