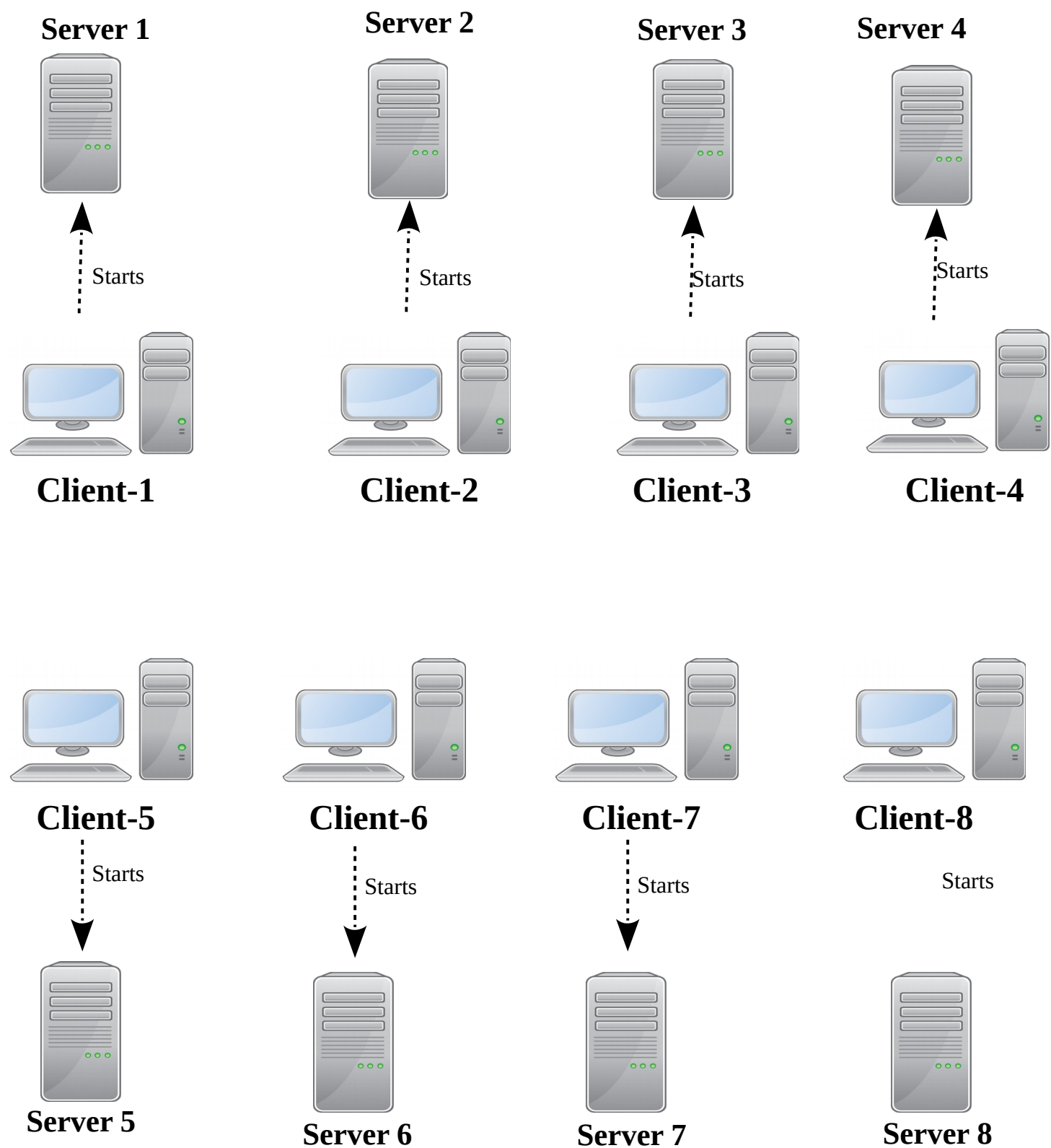
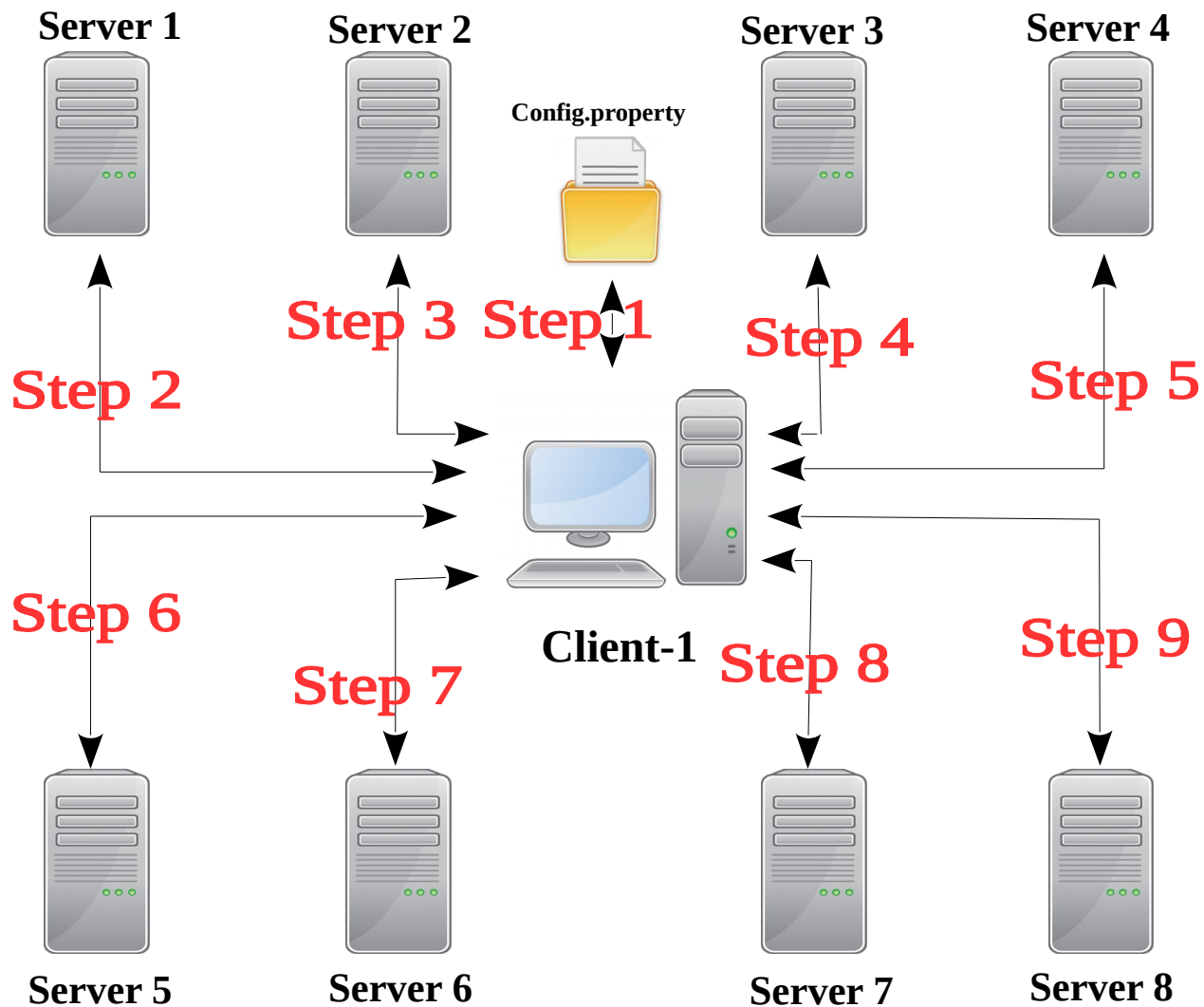


# Design Document :

1. At Start, all clients (Peers ) starts their own server.



**2. After that each client fetch information from 'config.property' file.** If it contains TotalPeer = 8, then it will try to connect to all 8 clients (including himself). Following figure shows, how client-1 establishes connection with all other Client's server. Similarly all other clients try to connect with each other.



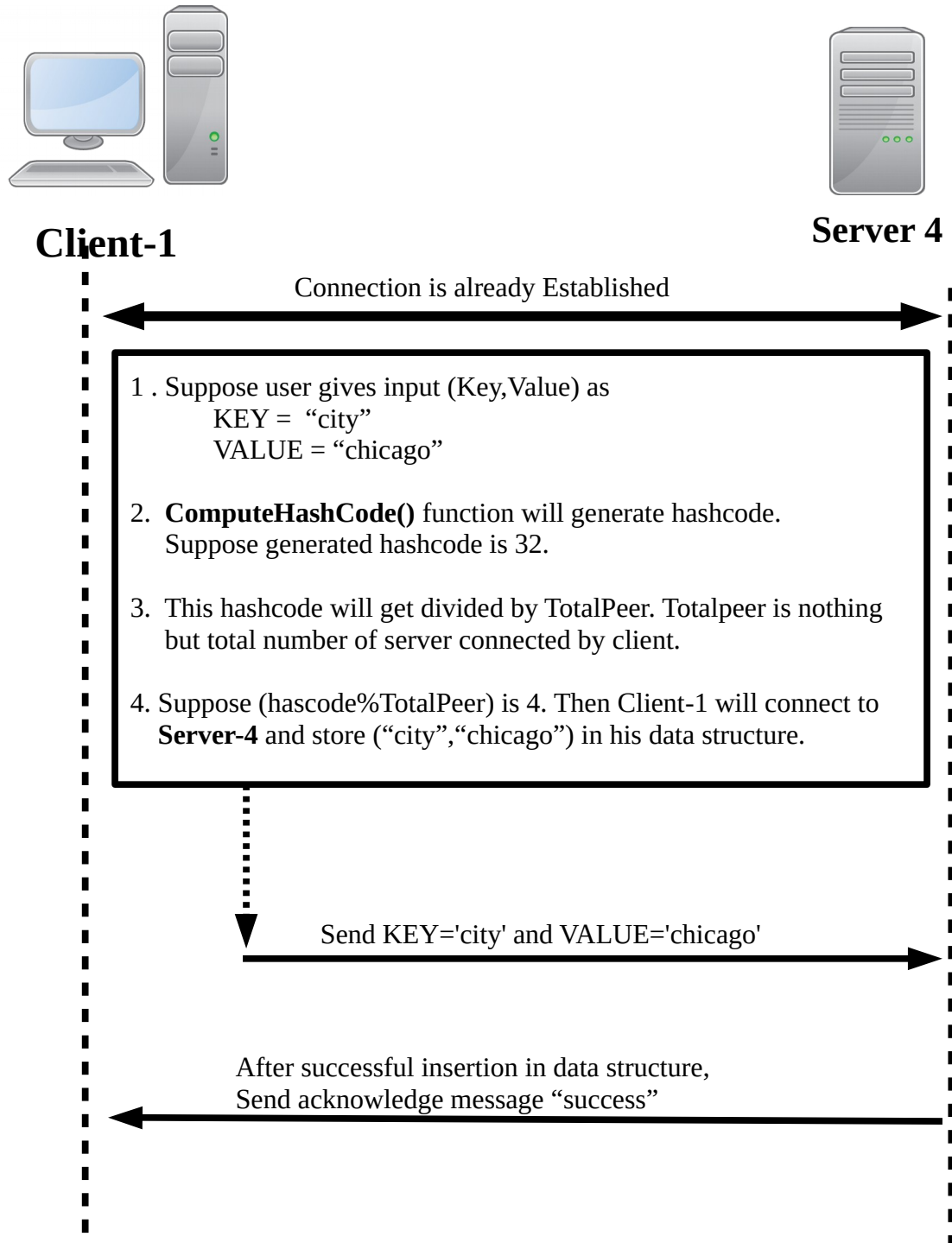
Step 1 : First Client-1 fetch information about all other server's IP address and PORT number from '**config.property**' file

Step 2 : According to design, Client will connect to all other servers in the sequential manner.  
In this step, Client-1 will connect to Server-1.

Step 3 : Client-1 will connect to Server-2.

Step 9 : Client-1 will connect to Server-8.

### 3. PUT request by client :



#### 4 . Condition where code will not work :

- i. To increase performance, all messages are passing to-and from server/client in String format.
- ii. Delimiter technique is used to distinguish GET/PUT/DELETE operation.
- iii. If user give KEY = “city” and VALUE = “chicago” for “PUT” operation, then String that will pass to server is as follows :  
“city@@chicago##PUT”
- iv. As shown above “@@” and “##”, this two delimiter are used.
- v. The code will not work, if user gives input which contains “@@” or “##” values.

#### 5. Future Scope :

- i. To overcome above delimiter limitation, String will be append with a header. This header will gives information about length of key and value. And last few bytes will be allocate for operation type.
- ii. So above (KEY,VALUE) pair (“city”,,”chicago”) will be stored as

Length of KEY	Length of VALUE	Space allocated for KEY	Space allocated for VALUE	OPERATION CODE
------------------	--------------------	----------------------------	------------------------------	-------------------

OPERATION CODE :

00 = PUT  
01 = GET  
10 = DELETE  
11 = INVALID