Peer Network Description

The overview of the network.

Gateway: The gateway of the network, a node want to join the network it needs to send the *"JOINING NETWORK "* message to the gateway. Gateway will save all nodes information, if a node join the network Gateway will pick some nodes to establish a *"route table"* and return back to the node. The peer and Gateway connect by TCP.

Peer: Once a node joined the network it will send message with other nodes base on the *"route table"* without the participate of Gateway. The *"route table"* only contains part of nodes in the network. The peers connection use UDP.

Peer ID: The id of the peer, which is the unique identify of peer in the network. In order to test, I use the join time as the parameter to hash. If the network already exist a peer hold same id, the return message will shows join fail.

Route Table: The *"route table"* is consist of *"leaf set"* and *"route set"*, same as the *"pastry"*, *"leaf set"* is the set which contains the node which is numerical close to the peer and the *"route set"* also works same as the *"pastry"*. The peer information saved in the *"route table"* are: peer id, peer ip, peer port.

Routing Processing: A peer send a chat message and input a tag, first hash the tag get a *"tag number"*, the target is to find a peer which *"peer id"* is same as the *"tag number"*. However it is common that can't find the peer which id is the *"tag number",* so this network will find the peer id is most numerical close to it. First peer will search the *"leaf set"*, then search the *"route set"*, if the peer found another peer which is numerical closest in it *"route table"* (of course closer than itself), the peer will send the chat message to the find peer, and the find peer will search as the same way(The detail search method can find in the *"peer"* class *"search_node_in_route_table"* method). Finally will find the best match peer, save the relevant chat message and return the chat record.

Test suggestion:

Tag: In order to test, I use the join time as the hash parameter so the max peer id is seven figures, so the tag better not exceed three letters.

Peer number: The more peer numbers the better test effect.

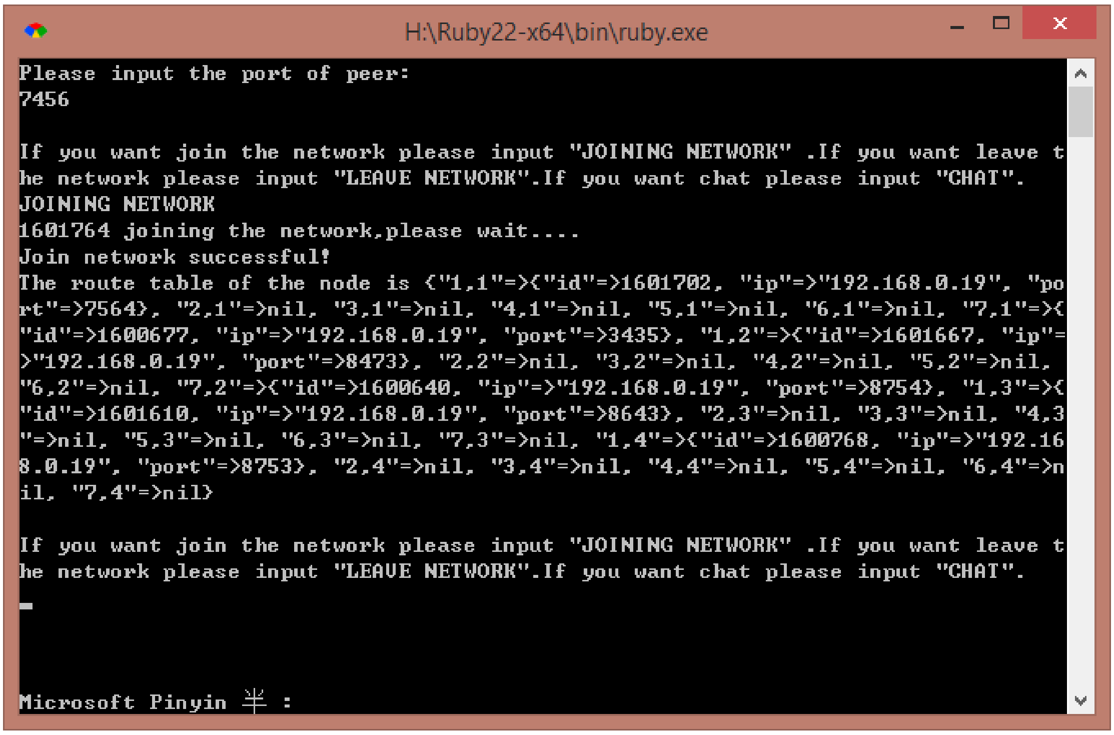
Self Evaluation:

This peer network achieve decentralised peer distributed(peer-to-peer communication, using route table to routing peer, each peer both client and server). The network can provide a reliability and large-scale function, support million peers. As for scalability the network support more peers by change some parameters.

In order to test, the ip I used is localhost, and change the port number as different clients.

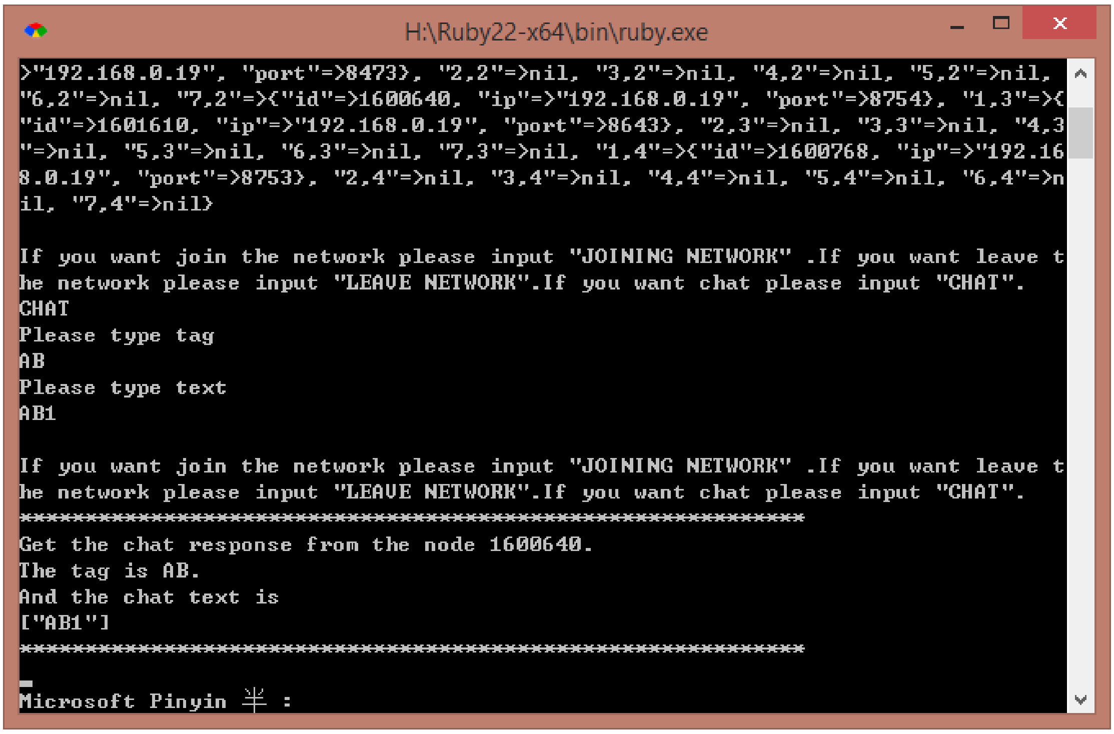
Test result:

JOIINING NETWORK:

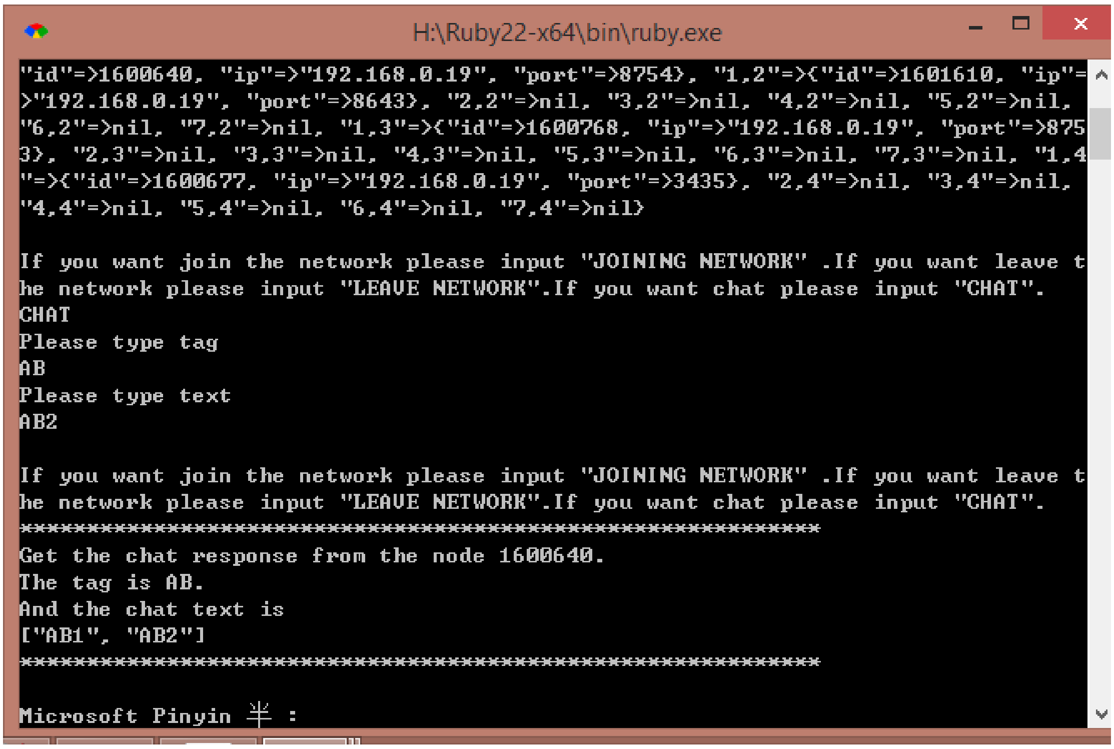


```
Please input the port of peer:
7456

If you want join the network please input "JOINING NETWORK" .If you want leave t
he network please input "LEAVE NETWORK".If you want chat please input "CHAT".
JOINING NETWORK
1601764 joining the network,please wait....
Join network successful!
The route table of the node is {"1,1"=>{"id"=>1601702, "ip"=>"192.168.0.19", "po
rt"=>7564}, "2,1"=>nil, "3,1"=>nil, "4,1"=>nil, "5,1"=>nil, "6,1"=>nil, "7,1"=>{
"id"=>1600677, "ip"=>"192.168.0.19", "port"=>3435}, "1,2"=>{"id"=>1601667, "ip"=
>"192.168.0.19", "port"=>8473}, "2,2"=>nil, "3,2"=>nil, "4,2"=>nil, "5,2"=>nil,
"6,2"=>nil, "7,2"=>{"id"=>1600640, "ip"=>"192.168.0.19", "port"=>8754}, "1,3"=>{
"id"=>1601610, "ip"=>"192.168.0.19", "port"=>8643}, "2,3"=>nil, "3,3"=>nil, "4,3
"=>nil, "5,3"=>nil, "6,3"=>nil, "7,3"=>nil, "1,4"=>{"id"=>1600768, "ip"=>"192.16
8.0.19", "port"=>8753}, "2,4"=>nil, "3,4"=>nil, "4,4"=>nil, "5,4"=>nil, "6,4"=>n
il, "7,4"=>nil}

If you want join the network please input "JOINING NETWORK" .If you want leave t
he network please input "LEAVE NETWORK".If you want chat please input "CHAT".



Microsoft Pinyin 半 :
```

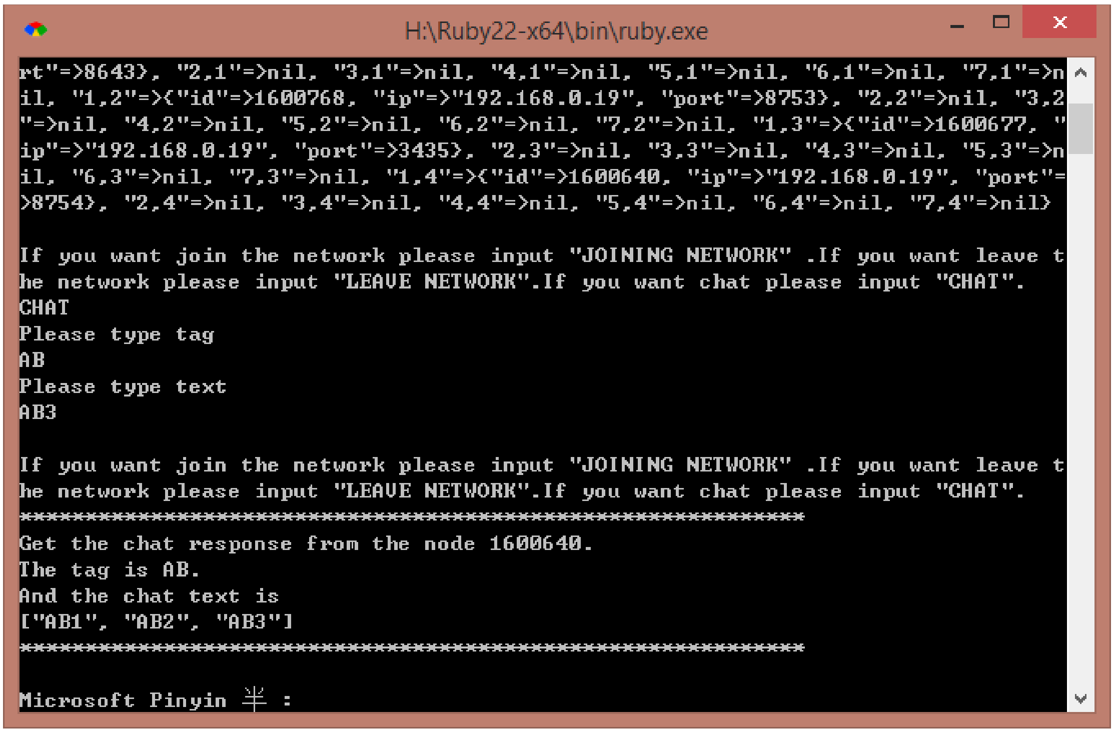
CHAT Test:



```
>"192.168.0.19", "port"=>8473}, "2,2"=>nil, "3,2"=>nil, "4,2"=>nil, "5,2"=>nil,
"6,2"=>nil, "7,2"=>{"id"=>1600640, "ip"=>"192.168.0.19", "port"=>8754}, "1,3"=>{
"id"=>1601610, "ip"=>"192.168.0.19", "port"=>8643}, "2,3"=>nil, "3,3"=>nil, "4,3
"=>nil, "5,3"=>nil, "6,3"=>nil, "7,3"=>nil, "1,4"=>{"id"=>1600768, "ip"=>"192.16
8.0.19", "port"=>8753}, "2,4"=>nil, "3,4"=>nil, "4,4"=>nil, "5,4"=>nil, "6,4"=>n
il, "7,4"=>nil}

If you want join the network please input "JOINING NETWORK" .If you want leave t
he network please input "LEAVE NETWORK".If you want chat please input "CHAT".
CHAT
Please type tag
AB
Please type text
AB1

If you want join the network please input "JOINING NETWORK" .If you want leave t
he network please input "LEAVE NETWORK".If you want chat please input "CHAT".
************************************************************
Get the chat response from the node 1600640.
The tag is AB.
And the chat text is
["AB1"]
************************************************************

Microsoft Pinyin 半 :
```



```
"id"=>1600640, "ip"=>"192.168.0.19", "port"=>8754}, "1,2"=>{"id"=>1601610, "ip"=
>"192.168.0.19", "port"=>8643}, "2,2"=>nil, "3,2"=>nil, "4,2"=>nil, "5,2"=>nil,
"6,2"=>nil, "7,2"=>nil, "1,3"=>{"id"=>1600768, "ip"=>"192.168.0.19", "port"=>875
3}, "2,3"=>nil, "3,3"=>nil, "4,3"=>nil, "5,3"=>nil, "6,3"=>nil, "7,3"=>nil, "1,4
"=>{"id"=>1600677, "ip"=>"192.168.0.19", "port"=>3435}, "2,4"=>nil, "3,4"=>nil,
"4,4"=>nil, "5,4"=>nil, "6,4"=>nil, "7,4"=>nil}

If you want join the network please input "JOINING NETWORK" .If you want leave t
he network please input "LEAVE NETWORK".If you want chat please input "CHAT".
CHAT
Please type tag
AB
Please type text
AB2

If you want join the network please input "JOINING NETWORK" .If you want leave t
he network please input "LEAVE NETWORK".If you want chat please input "CHAT".
************************************************************
Get the chat response from the node 1600640.
The tag is AB.
And the chat text is
["AB1", "AB2"]
************************************************************

Microsoft Pinyin 半 :
```



```
rt"=>8643}, "2,1"=>nil, "3,1"=>nil, "4,1"=>nil, "5,1"=>nil, "6,1"=>nil, "7,1"=>n
il, "1,2"=>{"id"=>1600768, "ip"=>"192.168.0.19", "port"=>8753}, "2,2"=>nil, "3,2
"=>nil, "4,2"=>nil, "5,2"=>nil, "6,2"=>nil, "7,2"=>nil, "1,3"=>{"id"=>1600677, "
ip"=>"192.168.0.19", "port"=>3435}, "2,3"=>nil, "3,3"=>nil, "4,3"=>nil, "5,3"=>n
il, "6,3"=>nil, "7,3"=>nil, "1,4"=>{"id"=>1600640, "ip"=>"192.168.0.19", "port"=
>8754}, "2,4"=>nil, "3,4"=>nil, "4,4"=>nil, "5,4"=>nil, "6,4"=>nil, "7,4"=>nil}

If you want join the network please input "JOINING NETWORK" .If you want leave t
he network please input "LEAVE NETWORK".If you want chat please input "CHAT".
CHAT
Please type tag
AB
Please type text
AB3

If you want join the network please input "JOINING NETWORK" .If you want leave t
he network please input "LEAVE NETWORK".If you want chat please input "CHAT".
***********************************************************
Get the chat response from the node 1600640.
The tag is AB.
And the chat text is
["AB1", "AB2", "AB3"]
***********************************************************

Microsoft Pinyin 半 :
```

In this test, three clients already join the network, the chat tag is "AB" and the chat text is shows above.

When a peer leaving the network it will send message to the gateway, gateway will delete the peer information.