

**Team : Bhavyan Mehta, Akshay Bhole, Meena Krishnappa, Honey Patodia**  
**Peer to Peer Review**

<u>Name</u>	<u>Comments</u>	<u>Rating</u>
Boada	<p>Service Discovery protocol is quite clear. Easy to implement.</p> <p>However, they can do away with Joinchannel as this has the overhead for new multicast group for each neighbors. Also can do without Preferred command if some information is present in the client itself about its parent.</p> <p>Message format is simple.</p> <p>Fault Tolerance is also easy to implement.</p> <p>Didnt find reference to resolving the name of the secondary clients</p>	3
Brian	<p>No description about Service Discovery protocol and Fault tolerance. Message format not that clear</p>	1
Dube	<p>Service Discovery protocol is quite clear. Easy to implement.</p> <p>However, for selecting neighbors use of generating random numbers on both the requesting server and other servers can be removed as this is an overhead. Also as the numbers generated are random, its quite possible that the requesting server would get a response from the other live servers pretty late.</p> <p>Message format is simple. Fault Tolerance is also easy to implement.</p> <p>Didnt find reference to resolving the name of the secondary clients</p>	4
Dugan	<p>Service Discovery protocol is quite clear. Easy to implement.</p> <p>However, the commands for sending the user information to other servers can be removed if some information about the server is stored in the client.</p> <p>Message format is simple. Fault Tolerance is also easy to implement.</p> <p>Didnt find reference to resolving the name of the secondary clients</p>	3

Hwang	<p>Cons</p> <ol style="list-style-type: none"> <li>1. The list of users is also being transferred every time when a new user get attach to the server to its neighboring servers, it is unnecessary and result in many updating and message passing.</li> <li>2. The network traffic will again increase due to the message transfer between server and agents due to the heartbeat message and the response message, this could be implemented easily with "I am alive message"</li> <li>3. There is no description of how the users will find the neighboring servers to get attached to when its own server crashes</li> </ol> <p>Pros:</p> <ol style="list-style-type: none"> <li>1. Node search has been described clearly.</li> </ol>	3
Jindal	<p>Cons:</p> <ol style="list-style-type: none"> <li>1. Unnecessary database management.</li> <li>2. No need of sharing the user information with other neighboring servers.</li> <li>3. No specification about how the servers will identify the neighboring servers.</li> </ol> <p>Pros:</p> <ol style="list-style-type: none"> <li>1. Searching had been handled properly.</li> <li>2. Alive messages properly handled.</li> </ol>	3
Nitin	<p>Cons</p> <ol style="list-style-type: none"> <li>1. No specification about the server discovery protocol, neighboring servers, server recovery</li> <li>2. "Alive message" are not handled.</li> <li>3. No measure to do load balancing.</li> <li>3. No name resolution.</li> </ol>	3

Pearson	<p>Cons:</p> <ol style="list-style-type: none"> <li>1. Server crash down scenario is not handled.</li> </ol> <p>Pros:</p> <ol style="list-style-type: none"> <li>1. Simple specification.</li> <li>2. Load balancing is handled properly</li> </ol>	4
Rohin	<p>Cons :</p> <ol style="list-style-type: none"> <li>1. Some of the conditions and specifications are missing like not specified how server will communicate agent list between them</li> <li>2. It seems that they are statically assuming the maximum number of servers so in case number of servers increase, they wont be able to find any neighbour</li> </ol> <p>Pros:</p> <ol style="list-style-type: none"> <li>1. Very clear and simple description</li> <li>2. Almost complete specification</li> </ol>	4.5
Roneck	<p>Cons:</p> <ol style="list-style-type: none"> <li>1. almost every where multicast messages are used, even in situations where they should not be used like replies. This will substantially increase the number of messages overhead in system</li> <li>2. There is no measure to do load balancing</li> <li>3. Search not defined properly</li> </ol> <p>Pros:</p> <ol style="list-style-type: none"> <li>1. Recovery and failure handled properly</li> <li>2. simple specification</li> <li>3. Alive messages properly handled</li> </ol>	3.5

Tapasvi	<p>Cons:</p> <ol style="list-style-type: none"> <li>1. Many specifications missing</li> <li>2. No name resolution</li> <li>3. Communication between servers about the list of agents is missing</li> </ol> <p>Pros:</p> <ol style="list-style-type: none"> <li>1. diagrams are very intuitive</li> <li>2. Clear explanation</li> <li>3. Recovery and Failure conditions properly handled</li> </ol>	4
Vaishnav	<p>Cons:</p> <ol style="list-style-type: none"> <li>1. Very complicated description. Tree structure is being specified which instead would be graph as nodes as there is bidirectional communication and the neighbors can be connected to any node making it a graph instead of a tree</li> <li>2. Find not specified properly</li> <li>3. No address resolution is done</li> <li>4. Server does not send Alive message and server's availability not discovered periodically. Only when Agent detects error in communication, it is discovered that server is missing.</li> <li>5. Server only have one neighbor initially</li> </ol>	3
Perez	<p>Flaw with technique that finds number of neighbors in the network. will yield huge difference in # of neighbors for 2nd server that comes online and for 100th server that comes online</p> <p>No history of chatrooms being maintained anywhere. How do the users come back to their original chatrooms?</p>	3
Reza	<p>Good server discovery mechanisms</p> <p>Good coverage of all cases and message formats</p> <p>BFS should be used rather than DFS</p>	4.5
Ritu	<p>exhaustive coverage of all standard communication messages</p> <p>If server goes down clients should be proactive in shifting to neighbors rather than neighboring servers being proactive. This is because the message from neighbors wont be received since the receiving server itself is down</p>	4

Ritwik	<p>Good standardized format for messages</p> <p>To check if Server is alive there is no need for Handshake between Clients and Server. It unnecessarily increases message traffic. Server sending 'Alive' messages should be enough. In that case there is no need for any feedback from clients.</p>	4