

## Bill-Donnie Protocol (BDP)

Version	isP2P	isSimple	SenderIP
Text	Future use		
IP1	isAlive	timeStamp	timeToLive
IP2	isAlive	timeStamp	timeToLive

### Fields:

- | **Version** – double – contains the version number of the protocol
- | **isP2P** – boolean – determines if the protocol is being used for peer-to-peer networks (*true*) or client-server networks (*false*)
- | **isSimple** – Boolean – determines if the message contains a list of IPEntry(s) that should be read by the receiver; if no list, then *true*; if there is a list, then *false*
- | **SenderIP** – String – contains the IP address of the originator of the packet
- | **Text** – String – contains the text to be printed when the packet is received
- | **Future use** – any newly implemented fields could exist here
- | **List of IPEntry(s)** – Hashtable<String, IPEntry>
  - Ø **IPAddress** – String – contains the IP address of a node, used as the key of the hashtable
  - Ø **isAlive** – boolean – determines if a given node is up (*true*) or down (*false*)
  - Ø **timestamp** – LocalDateTime – contains the time of the latest node update
  - Ø **timeToLive** – int – contains how long a node has left until it is considered dead; by default starts at 30 seconds

## **C/S**

In the client-server model, clients are supposed to be started on different machines and the server is required to be initiated before all clients.

### **Client:**

The client needs to know the IP address of the server before the launch and the IP address of the server has been configured in the serverIP.txt file.

The client initially will be guaranteed to send to the server at least a message and move into the loop waiting for the server's response.

Whenever the server receives the message, it will immediately reply with a list of the known IP addresses of other clients.

If the client does not receive any response for  $2 * \text{MAX\_TIME}$  (the 40s), then one of them will take over the server and assume the former server is dead.

The mechanism of choosing who to become the server is to let the client check if it is the first entry on the list, otherwise, the first place client will become its server and run the server code.

### **Server:**

The server is not necessary to know the IP address of the client beforehand since it is capable to accept new incoming clients.

The server will begin decrementing the timeToLive on the IPEntry of a node. If the timeToLive is 0, then the server will assume the client is dead. The latest message from a particular client can reset the timer in order to keep the server aware it is alive.

Whenever the server receives any information of any client, it will automatically broadcast and update the list to all other clients.

**P2P:**

Upon startup, a host will broadcast a simple (isSimple=1) message to a list of any nodes within the "ipList.txt" file. After which, the host will start broadcasting non-simple (IsSimple=0) messages at random intervals between 0 and 19 seconds, then once again at 20 seconds.

When a message is received, the host checks if has the sender's IP address stored in its hashtable of IPEntries and adds it if it is not present.

If the message is simple, then the host only marks it as alive and continues the loop. If the message is non-simple, the host compares its hashtable to the received hashtable. If any entries are more recent than the host's, the host updates their entry to match.

If no message is received for 30 seconds (i.e., timeToLive reaches 0), then the host marks the node as dead.

**How to run Client-Server:**

1. On all machines, make sure that "myIP.txt" contains a single line of text which is the machine's IPv4 address
2. On all client machines, make sure that "serverIP.txt" contains a single line of text which is the server's IPv4 address
3. Run the server first
4. Run any number of clients
5. If the server dies, the first living node in the list will take over and the rest of the nodes will start reporting to it

**How to run P2P:**

1. On all machines, make sure that "myIP.txt" contains a single line of text which is the machine's IPv4 address
2. On all machines, make sure that "ipList.txt" contains the IP address of any other node to be connected to the server. The P2P driver will dynamically add merge lists with nodes once it learns about them
3. Run any number of nodes