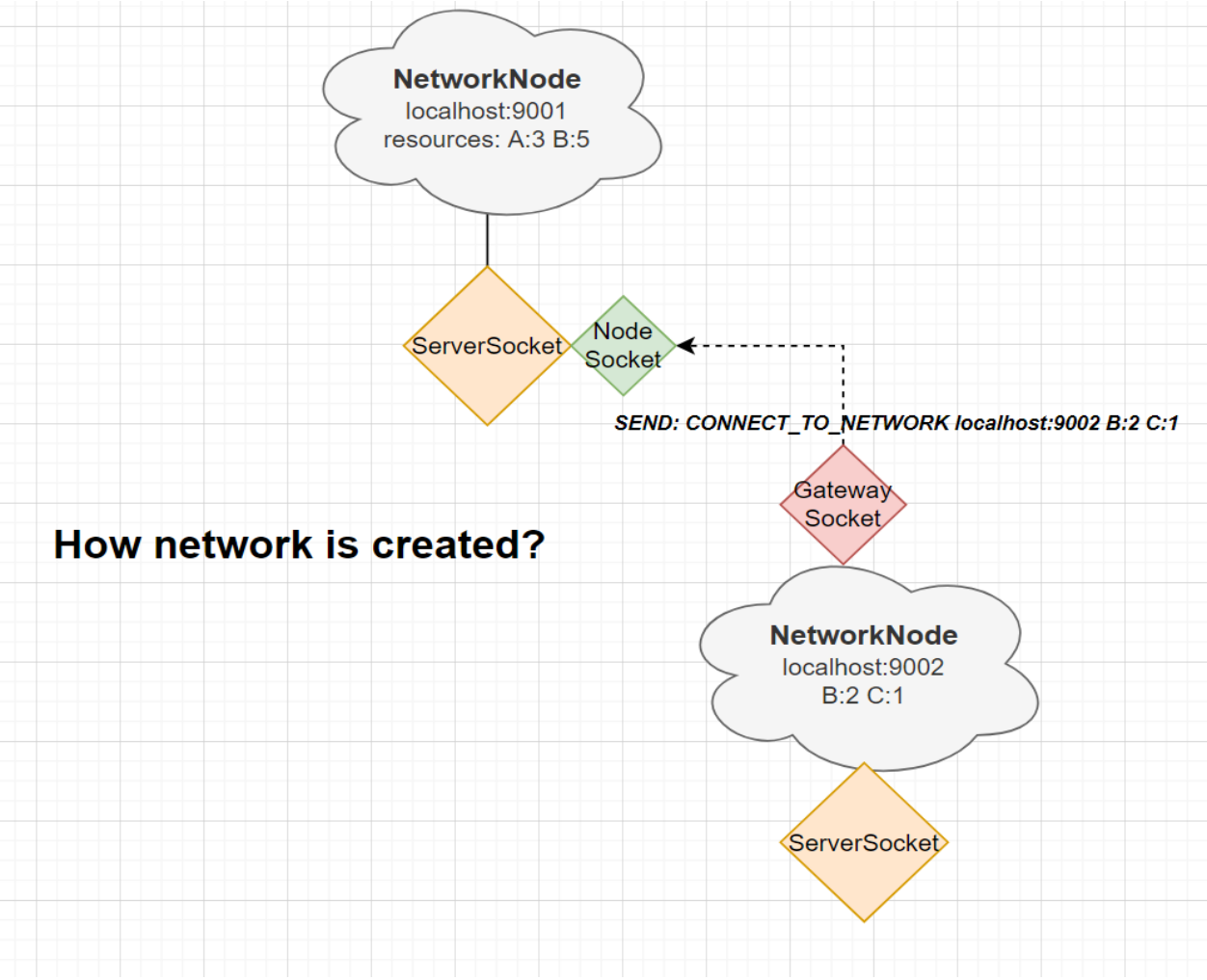
1. How Network is created?

The network is created by connecting logical NetwokNode to another NetworkNode further called Gateway (or it becomes gateway by itself if it first NetworkNode in the Network). Each NetworkNode has ServerSocket instance that listens all the time for new connection, which can be other NetworkNode or NetworkClient. In case of ServerSocket it accepts new connection, it creates new NodeSocket instance which can read and write to/from NetworkClient or another NetworkNode.

If NetworkNode is going to connect to the Network, this node creates its own GatewaySocket which connects to the ServerSocket of entry NetworkNode and establish connection with appropriate created NodeSocket and send *"CONNECT_TO_NETWORK ip:port <resource_name:quantity>"*

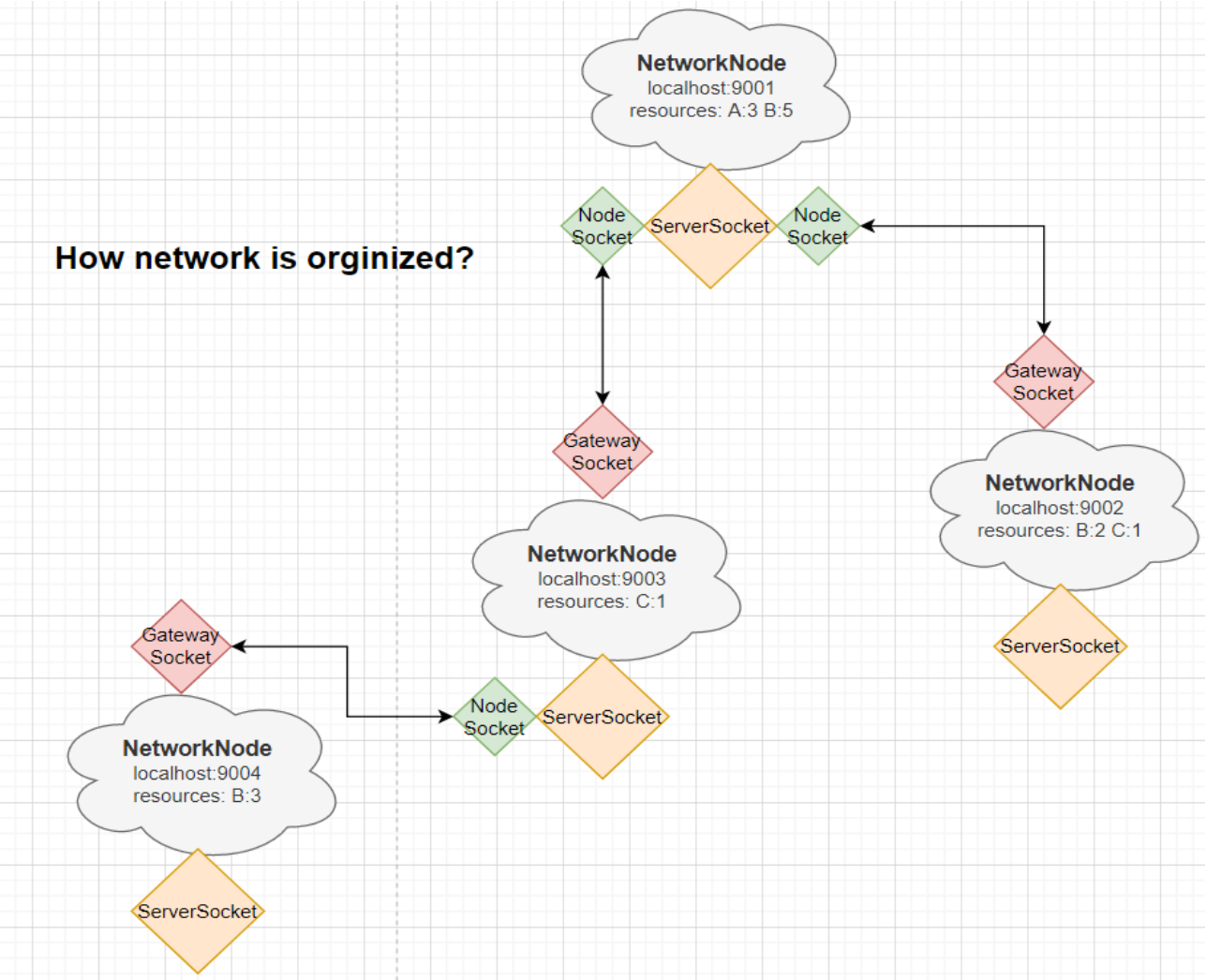


2. How Network is organized?

All communication is relied on TCP socket connections. Parent NetworkNode can communicate with connected NetworkNodes and communicate with its parent NetworkNode via GatewaySocket if existed.

The Network will always have only one NetworkNode which doesn't have any parent NetworkNode above it (let's call it Root node).

Each NetworkNode together with all recursive connected NetworkNodes creates a Subnetwork. Such NetworkNode has information about what NetworkNodes (and their resources) are connected to this subnetwork



3. Communication between nodes?

Important note: the implemented Network does not support messages with a specific address. Against of it, it sends messages from child to parents nodes, or sends broadcast from parent to child nodes.

Communication Protocol:

Requests:

- *CONNECT_TO_NETWORK ip:port <resource_name: quantity>*

New NetworkNode send it to the parent NetworkNode if it exists. Sends list of resources which are assigned to it.

For example: *CONNECT_TO_NETWORK localhost:9002 B:2 C:1*

- *ASK_RESOURCES who_ask <resource_name: quantity>*

New NetworkClient connected to the network and asked to allocate resources for him. who_ask is -ident argument of NetworkClient. First Subnetwork that satisfy requirements (reminder: each NetworkNode creates its own Subnetwork and knows about all connected nodes and their resources) will start to allocate resources.

For example: *ASK_RESOURCES 3 B:10 C:5*

- *ASK_TERMINATE*

NetworkClient sends command "TERMINATE" to the network. This message will be sent up the chain to the root to be broadcasted to all nodes in the network.

Responses:

- *ALLOCATE_RESOURCES <resource_name:quantity:ip:port>*

NetworkNode broadcast this message to all nodes in its Subnetwork. Each node parse this message and if this node is on the list - modify its resources appropriately.

For example: *ALLOCATE_RESOURCES A:10:localhost:9004 A:2:localhost:9001 B:3:localhost:9003 B:3:localhost:9002*

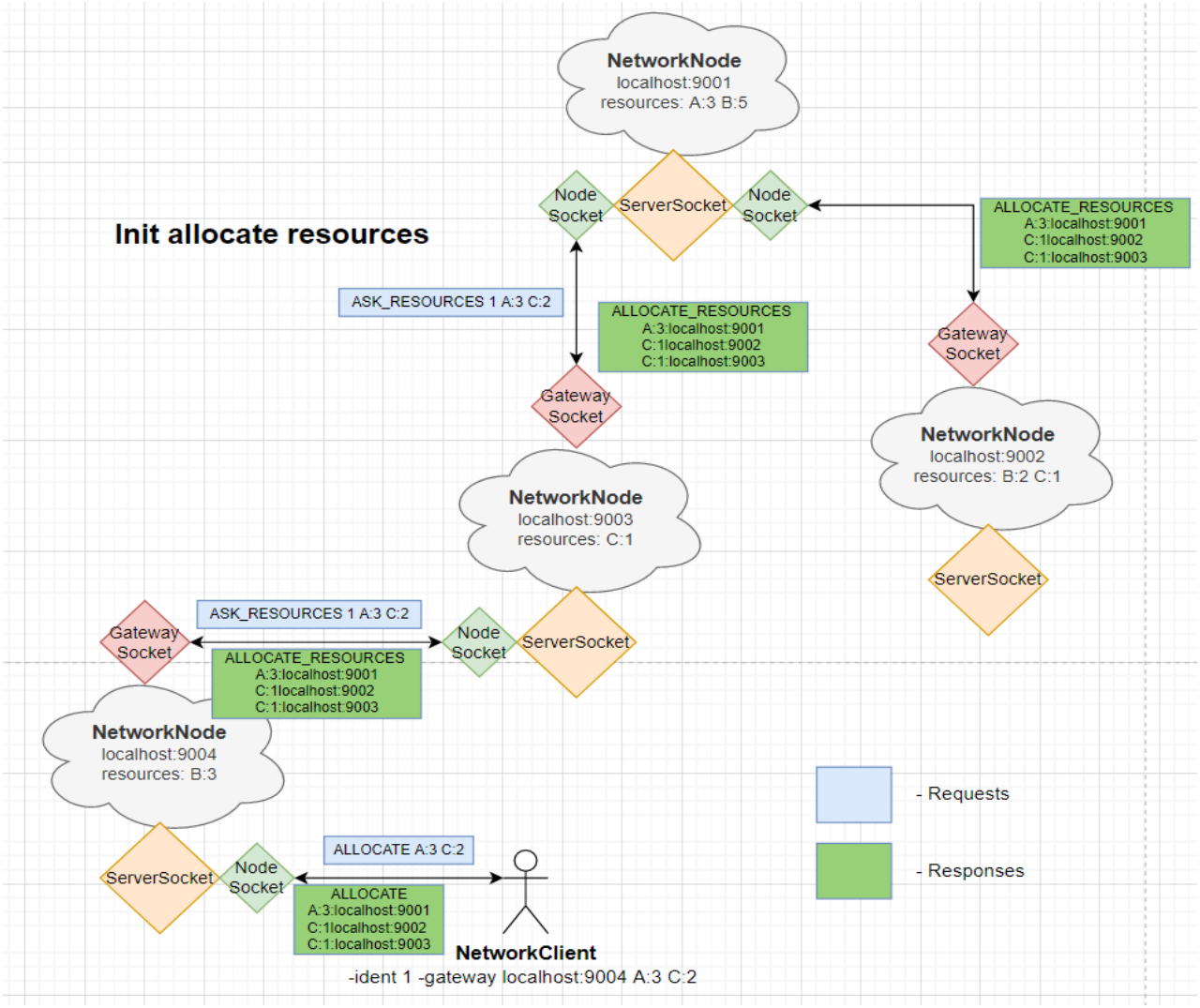
- *ALLOCATION_FAILED*

Network wasn't able to find required resources by NetworkClient. Root NetworkNode broadcast this message to all nodes and entry point of NetworkClient will format this message appropriately.

- *TERMINATE_NETWORK*

This message is broadcasted by Root to all Network nodes recursively. After parsing this message, a NetworkNode should disconnect all sockets connected to it and terminate its work.

Examples:



Let's assume a network is already created. NetworkClient instance was created in the following way:

NetworkClient -ident 1 -gateway localhost:9004 A:3 C:2

NetworkClient ask Network to allocate resources: A:3 C:2. The entry point to the Network is localhost:9004.

NetworkNode 9004 checks if there is enough resource in the current subnetwork (which currently consist of only 9004 itself). After negative check, it sends next command:

ASK_RESOURCES 1 A:3 C:2

to the parent node (9003). 9003 sends the same command to the 9001. 9001 check if there is enough resources in the Subnetwork (and it's the whole Network). Since node 9001 has information about all Subnetwork nodes, it decides there are enough resources to allocate.

localhost:9001 will need to allocate A:3

localhost:9002 will need to allocate C:1

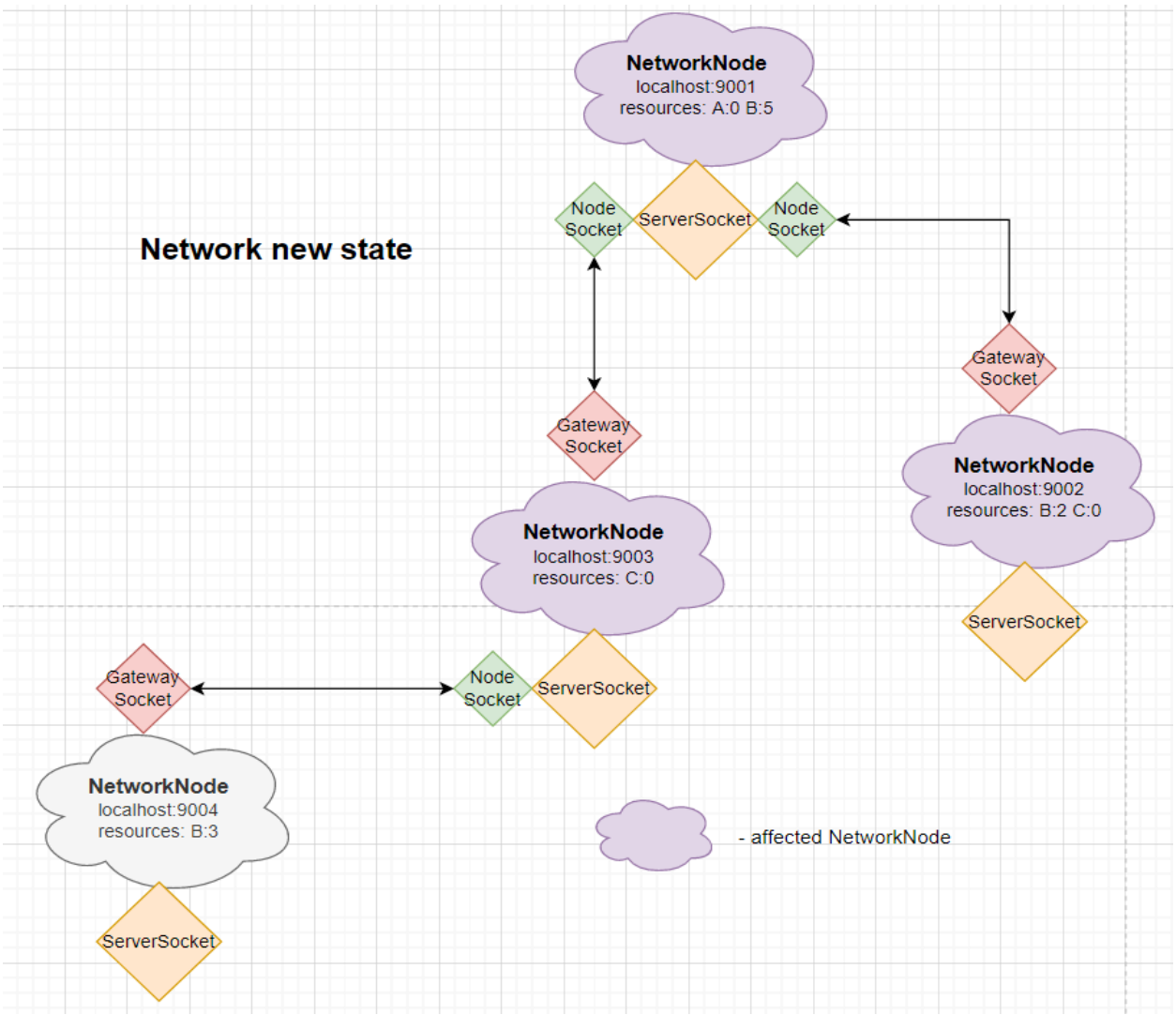
localhost:9003 will need to allocate C:1

9001 Starts broadcast the next message recursively to all nodes in the network:

ALLOCATE_RESOURCES A:3:localhost:9001 C:1localhost:9002 C:1:localhost:9003

Node 9004 takes this message from its parent 9003. Check if it needs to allocate some resources. This message also means the Network allocated resources successfully, and send ALLOCATE together with pairs which nodes allocate what resources to the client. After that, NetworkClient is being disconnected from the Network.

Each affected NetworkNode sends CONNECT_TO_NETWORK command again, to update parent knowlegde about resources in Subnetworks.


Network new state
NetworkNode
localhost:9001
resources: A:0 B:5
Node Socket
ServerSocket
Node Socket
Gateway Socket
Gateway Socket
NetworkNode
localhost:9002
resources: B:2 C:0
ServerSocket
NetworkNode
localhost:9003
resources: C:0
Gateway Socket
Node Socket
ServerSocket
NetworkNode
localhost:9004
resources: B:3
ServerSocket
- affected NetworkNode