Bug Report

Ticket 1: Undefined `msgmaxlength` for Unknown Prefix

Issue Type: Bug
Priority: High
Status: Open

Description:
The ConstructMsg procedure does not handle the case where an unknown prefix type is provided. When an unknown prefix type is passed, the variable `msgmaxlength` is not defined, leading to a crash when checking the length of the message.

Steps to Reproduce:

1. Call `ConectionManager::ConstructMsg` with an invalid `prefixType` value (e.g., 3).
2. Observe that the code attempts to use `msgmaxlength`, which was not defined.

Expected Results:
The code should handle unknown prefix types, without causing a crash.

Actual Results:
The script crashes due to the use of an undefined `msgmaxlength` variable.

Suggested Fixes:

1. Define a default value for `msgmaxlength` or ensure that it is only used when it is properly set.
2. Add an error message or handling mechanism for unknown prefix types.

Ticket 2: Unknown Suffix Handling

Issue Type: Bug
Priority: Medium
Status: Open

Description:
The `ConstructMsg` procedure does not handle unknown suffix types properly. If an unknown suffix type is provided, the message is sent with no suffix, which may not be the intended behavior.

Steps to Reproduce:

1. Call `ConectionManager::ConstructMsg` with an invalid `suffixType` value (e.g., 3).
2. Send the message and observe that it is sent without any suffix modifications.

Expected Results:
The code should handle unknown suffix types, potentially by logging an error or rejecting the message.

Actual Results:
The message is sent without any suffix when an unknown suffix type is used.

Suggested Fixes:

1. Add handling for unknown suffix types by logging an error or using a default suffix.

Ticket 3: Message Body Length Validation

Issue Type: Bug
Priority: Medium
Status: Open

Description:
The `ConstructMsg` procedure does not validate the length of the message body. There are no checks to ensure that the message body length is between 1 and 30 characters. As a result, empty message bodies are accepted, and bodies longer than 30 characters are simply trimmed.

Steps to Reproduce:

1. Call `ConectionManager::ConstructMsg` with an empty message body or a message body longer than 30 characters.
2. Observe that the message body is accepted and processed without validation.

Expected Results:
The message body should be validated to ensure its length is between 1 and 30 characters. An error should be raised if the length is outside this range.

Actual Results:
Empty message bodies are accepted, and longer message bodies are trimmed without validation.

Suggested Fixes:

1. Add validation for the message body length at the beginning of the `ConstructMsg` procedure.
2. Raise an error or log a message if the body length is outside the allowed range.

Ticket 4: Client Closure Handling and Client List Management

Issue Type: Bug
Priority: High
Status: Open

Description:
When a client is closed, it is not removed from the client list. Consequently, closed clients may still receive broadcast messages, leading to unnecessary operations and the server would waste resources trying to broadcast messages to clients that are no longer available.

Steps to Reproduce:

1. Start the server and connect multiple clients.
2. Close one or more clients using `ConectionManager::CloseClient1` or `ConectionManager::CloseClient2`.
3. Broadcast a message using `ConectionManager::BroadCast`.

Expected Results:
Closed clients should be removed from the client list, and they should not receive broadcast messages.

Actual Results:
Closed clients still appear in the client list and receive messages.

Suggested Fixes:

1. Remove closed clients from the `clientlist` in the `CloseClient` procedure.
2. Update the `BroadCast` procedure to skip closed clients.
