Università degli Studi di Bologna
Facoltà di Ingegneria

Principles, Models, and Applications for Distributed Systems M

Lab assignment 3
Connectionless Java Sockets

Luca Foschini

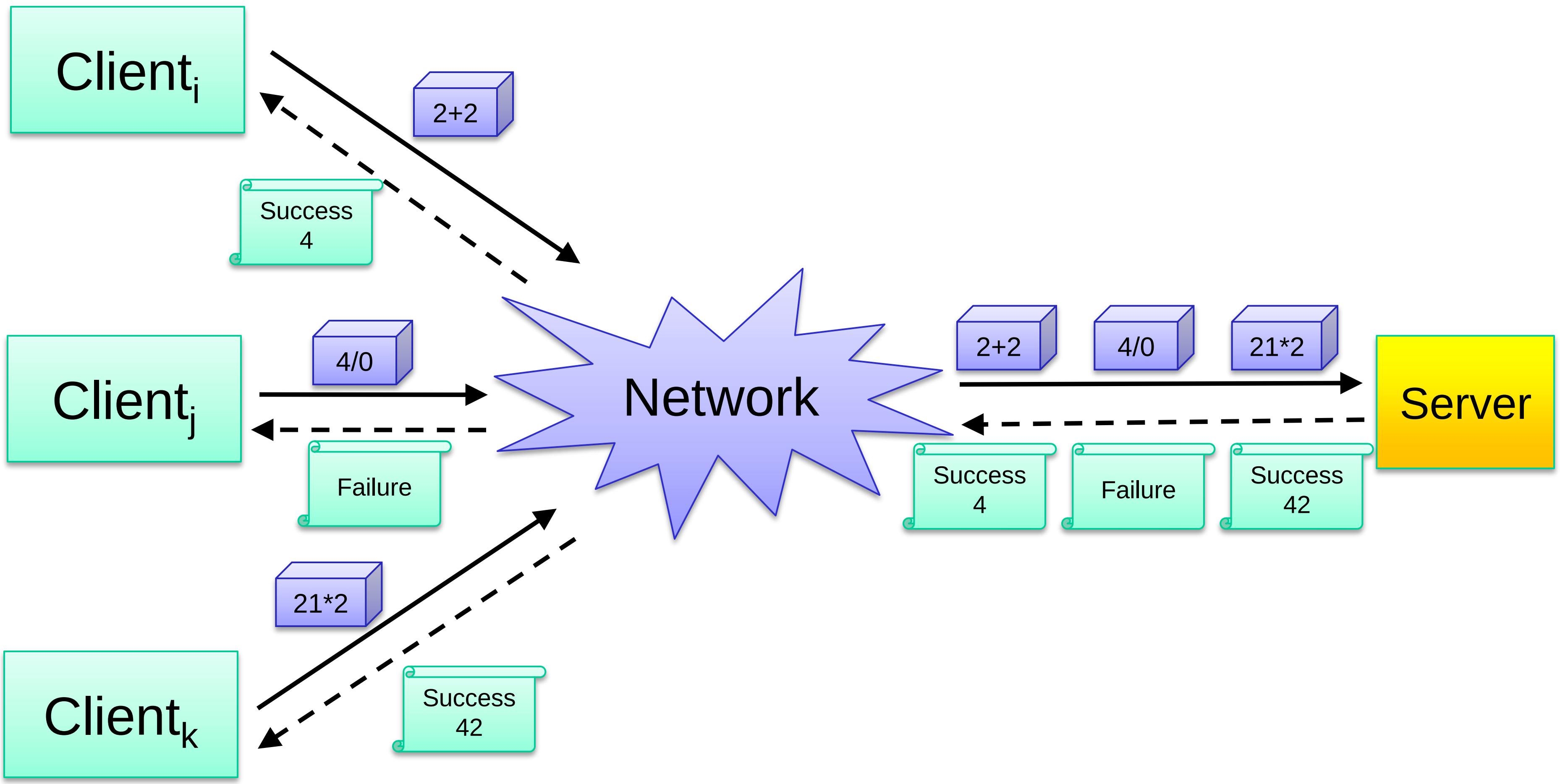
Remote Arithmetic Evaluation

Develop a C/S application that allows **remote evaluation of the four basic arithmetic operations**.

The **Client** sends to the server a request packet with the desired operation type (addition, subtraction, multiplication, division symbol) and the two operands to perform the arithmetic operation (two integers). The client cyclically asks the user to input these data using the console. Upon receiving the response, it prints the result to the screen.

The **Server** extracts the operation type and the operands from the request packet, performs the operation, and sends a datagram with a response string – "**success**" or "**failure**" – and the result of the operation (an integer).

Distribuited architecture



Implementation insights: Server

ArithmeticServer is the server: it replies to arithmetic operation requests and is run by using the following invocation interface from the command prompt:

ArithmeticServer udpport

The server waits incoming requests on port udpport; if the user does not specify a port, it has to be bound to port 1948.

When the server receives a datagram packet, it reads from it a **string with a single character that specifies the desired arithmetic operation** (“+”, “-”, “*”, “/”) and **two integers**. It tries to perform the operation and, if it is successful, it sends back a datagram packet with the string “**success**” and the **result of the operation (an integer)**; otherwise, for instance when the operation string is not valid or the client requests a division by 0, it sends back the string “**failure**” and an error indication.

Implementation insights: Client

ArithmeticClient is the client: it asks the user for operation requests, prepares the request packet and sends it to the server; then, it writes on the screen the received response. It is run by using an invocation interface with two arguments (IP address and UDP port of the ArithmeticServer):

ArithmeticClient ArithmeticServerIP ArithmeticServerPort

For each interaction, the client asks the user the input data (the **operation symbol** and the **two integers**), prepares the datagram packet, and sends it to the server on the specified UDP port. Then, it waits the response and **prints the outcome of the request** (success/failure) and the **result of the operation** on the screen. Finally, it asks the user for other requests, until EOF.