

Request-Reply Implementation Handout, CptS 464/564, 9/00

Examples from Section 4.4 in text, expanded

Client

```
Main(){
    ....
    // obtain object reference somehow
    args = // marshal them
    reply = doOperation(ref, method, args);
    // unmarshal reply into variables
}
```

```
doOperation(ref, methodID, args) {
    req.messageType=Request;
    req.requestID=sequence++;
    req.objectReference=ref;
    req.methodID=methodID;
    req.arguments = args
    send(req, ref.CommID);
    receive(replyMsg);
    return replyMsg;
}
```

(Contrast this with using CORBA.....)

(Now think of how to extend this to handle cases:

1. Request message lost (what should **DoOperation** do?)
2. Reply message lost (what should server-side do?)
3. Duplicate request received (what should server-side do?)

for idempotent and non-idempotent operations???)

Server-Side

```
Main(){
    loop forever
        req = getRequest();

        // dispatch loop
        Case req.methodID:
            Case 0:
                // unmarshall a1, a2 ...
                // from req.args
                method0(a1, a2, ...)
                // set return value in reply's
                // arguments if
                // method0 returns a value
                reply.arguments = ...
                break;
            Case 1:
                ....
            end case

        sendReply(reply, request.requestID);

    end loop
}
static CommunicationID senderID;

getRequest() {
    senderID = receive(reqMsg);
    return reqMsg;
}

sendReply(replyMsg, requestID) {
    reply.messageType = Reply;
    reply.requestID = requestID;
    send(replyMsg, senderID);
}
```