



```

void Source::handleMessage(cMessage *msg)
{
    if (msg==timeoutEvent){
        if (counter > 0) {
            EV << "Counter value = " << counter;
            EV << "\n";
            counter--;
            EV<< "Time out expired";
            cPacket *pkt_copy = pkt -> dup();
            send(pkt_copy, "out");
            scheduleAt(simTime()+timeout, timeoutEvent);
        } else {
            finish();
        }
    }
    else {
        cPacket* pktr = check_and_cast<cPacket *> (msg); //we assume that Destination
        int type = pktr->getKind();
        if (type == 0) {

```

```
#ifndef __ARQ2_SOURCE_H_
#define __ARQ2_SOURCE_H_

#include <omnetpp.h>

using namespace omnetpp;

class Source : public cSimpleModule
{
    private:
        simtime_t timeout;
        cMessage *timeoutEvent;
        cPacket *pkt;
        int counter;

    public:
```

```

Source::Source(){
    timeoutEvent = nullptr;
}

void Source::initialize()
{
    counter = par("num_messages");
    timeout = 1.0;
    timeoutEvent = new cMessage("timeoutEvent");
    EV<<"Sending initial packet";
    pkt= new cPacket("packet");
    pkt -> setBitLength(100);
    cPacket *pkt_copy = pkt -> dup();
    send(pkt_copy, "out");
    scheduleAt(simTime()+timeout, timeoutEvent);
}

```

simple Source

```

{
    parameters:
        @display("i=block/source");
        int num_messages = 100;
    gates:
        input in;
        output out;
}

```