

OMNeT++ 5.2

Omnet in a nutshell

- Discrete event simulator of modules that communicate via messages through channels.
- OpenSrc in C++; Commercial: OMNEST
- Nicht beschränkt auf Netzwerkprotokolle

→ Bestand:

* Simulation Kernel with utility classes: Topologie lässt sich abbilden via NED, Network Description Language

* .ini Files für unterschiedliche Startparameter

Syntax-Sample

```
module name{
  parameters: ..
  gates: ..
  input in;
  output out;
  submodules:
    app: EtherTrafficGen;
    llc: EtherLLC
    mac: EtherMac;
  connections:
    app.out -> llc.hlln;
    app.in <- llc.hlOut;
    llc.macIn <- mac.llcOut
    llc.macOut -> mac.llcIn;
    mac.phyIn <- in;
    mac.phyOut -> out;
}
```

Building the Programm: Übernimmt die IDE.

Cmessage: Data

cPacket: Packed Message (encapsulate(cMessage *msg) or decapsulate())

z.B.:

```
packet Datagram{
  int destAddr = -1;
  int srcAddr = -1;
  int ttl = 32;          // Time to live
}
```

=> wird mittels opp_msgc generiert in _m.h und _m.c

wichtig: initialize();

Working with OMNet

“Bestenfalls 20 LOC”

TicToc Tutorial

www.omnetpp.org/doc/omnetpp/tictoc-tutorial/

```
simple Txc1{
  gates:
    input in;
    output out;
}
```

```

network TicToc{
    submodules:
        tic: Txc1;
        toc: Txc1;
    connections:
        tic.out → toc.in;
        tic.in ← - toc.out;
}

```

In J:

```

public cSimpleModule{
    protected:
        virtual void initialize() override;
        virtual void handleMessage(cMessage *msg) override;
};
Define_Module(Txc1);

```

```

void Txc1::initialize(){
    if(strcmp("tic",getName()) == 0){
        cMessage * msg = new cMessage("tictocMsg");
        send(msg, "out");
    }
}

```

```

void Txc1::handleMessage(cMessage *msg){
    send(msg, "out");
}

```

.ini File ist notwendig.

>> Erweiterungen: Siehe Folie