



the OPEN Simulator

## OMNEST™ Simulation Environment

**OMNEST is an open, versatile, component-based simulation environment** with a modular architecture, strong GUI support and an embeddable simulation kernel. Source code is provided to give you full control of your software. OMNEST is the commercial version of the powerful **OMNEST** simulation environment, used worldwide at universities and research groups.

### Network Simulation



**OMNeT++/OMNEST is extensively used for R&D-type network simulations.** Somewhat similarly to ns2, OMNeT++ has an active user community in the Academia, who created a large body of open-source network simulation models and model frameworks, from ad-hoc and wireless simulations to MPLS, Diffserv and IPv6.

Simulation models are truly component-based, allowing you to freely combine existing protocol models to form hosts and network devices. There is support for real-time simulation and network emulation as well.

### Discrete Event Simulation



**With OMNeT++/OMNEST you can simulate** anything that can be mapped into a number of entities, sending messages to each other. What entities and messages mean is up to you. That is, one can simulate queueing networks ("jobs" or "customers" travelling among queues), computer networks (computers exchanging packets), businesses (employees sending documents and files to each other and making phone calls), hardware architectures (e.g. a CPU making requests to the disk, or units communicating on a bus) and many other systems.

### Embedding



**The Omnest™ simulation kernel can be embedded into 3rd party applications.** Your embedding application will link with the simulation kernel as a C++ library. Models can be developed and tested in the original OMNeT++/OMNEST GUI, then embedded unchanged into your application.



the OPEN Simulator

## Network Simulation

**OMNEST is extensively used for R&D-type network simulations.** Somewhat similarly to ns2, OMNEST has an active user community in the Academia, who created a large body of open-source network simulation models and model frameworks, from ad-hoc and wireless simulations to MPLS, Diffserv and IPv6.

Simulation models are truly component-based, allowing you to freely combine existing protocol models to form hosts and network devices. There is support for real-time simulation and network emulation as well.

### Simulation frameworks over OMNEST/OMNEST:

**INET Framework** - supports ad-hoc, wireless and wired (LAN, WAN) simulations (802.11, Ethernet, TCP, IP, IPv6, OSPF, MPLS, RSVP, etc.) The INET Framework is maintained by OMNEST people, with frequent contributions from the user community.

Cleanly programmed and extensively commented models lend themselves to experimentation with protocols and various architectures.

Several models are ported versions of real-life networking software like the Quagga routing daemon, or the AODV-UU implementation, which guarantees simulation accuracy.

**IPv6SuiteWithINET** - for IPv6, MIPv6 and wireless (802.11)

**Mobility Framework** - for wireless mobile and ad-hoc networks

**NesCT** - for simulating wireless motes running TinyOS

**SimSANS** - for simulating Storage Area Networks

These simulation frameworks have been programmed in C++, and are distributed in source code form. With the exception of the INET Framework, these models are maintained by people independent of Simulcraft Inc, so we naturally cannot provide any warranty for them. However, their large exposure to the Academia means that they get reviewed and bugfixed and improved by many people.

### OMNEST features relevant to network simulation

Existing protocol models can be freely combined to form hosts and network devices

Graphical tools for simulation building and evaluating results

Powerful GUI for tracing, debugging and animating your simulations

Emulation / Real-Time simulation / hardware-in-the-loop simulation support

Ask for help on the mailing list with 300+ active subscribers

Responsive and expert commercial support



the OPEN Simulator

## Discrete Event Simulation

**With OMNeT++/OMNEST you can simulate** anything that can be mapped into a number of entities, sending messages to each other. What entities and messages mean is up to you. That is, one can simulate queueing networks ("jobs" or "customers" travelling among queues), computer networks (computers exchanging packets), businesses (employees sending documents and files to each other and making phone calls), hardware architectures (e.g. a CPU making requests to the disk, or units communicating on a bus) and many other systems.

You assemble the simulated "network" from components written in C++ (by you or by someone else). Once the components are present, no further C++ programming necessary to build simulations from them - you will use a high-level declarative language for this task, or the graphical editor. OMNEST/OMNeT++ provides the simulation library (simulation kernel) and the execution environment, and various GUI and non-GUI tools for creating and running the model, and evaluating the results.

The OMNEST/OMNeT++ simulation kernel is very extensible, and can accommodate special needs such as parallel/distributed simulation, real-time simulation (emulation), database integration, embedding, comes with source code to increase your degree of freedom, and is backed with superior support.

### Features

- Hierarchical, component-based modelling
- C++-based, high performance simulation kernel
- Wide range of applicability
- Models are self-documenting
- Source code is provided
- Standards support, open interfaces
- Graphical tools for simulation building and evaluating results
- Powerful GUI for tracing, debugging and animating your simulations
- Responsive and expert support
- Parallel simulation and other advanced features



the OPEN Simulator

## Embedding

**The OmneST™ simulation kernel can be embedded into 3rd party applications.** Your embedding application will link with the simulation kernel as a C++ library. Models can be developed and tested in the original OMNeT++/OMNEST GUI, then embedded unchanged into your application.

### Features

- Hierarchical, component-based modelling
- C++-based, high performance simulation kernel
- Source code is provided
- Parallel simulation and other advanced features
- Responsive and expert support

## OMNEST™ Features

### General

- **Hierarchical, component-based modelling**  
You build models from self-contained components using a high-level declarative language (NED), with arbitrary levels of nesting. Use the graphical editor or your favourite text editor.
- **C++-based, high performance simulation kernel**  
Atomic components are programmed in C++, using a well-defined API to the simulation library. The use of C++, together with the streamlined simulation kernel, provides high event/sec throughput.
- **Wide range of applicability**  
OMNEST has proven itself in the simulation of queuing networks, business processes and high-level architectures. In the academia, it is mostly used for simulating wired and wireless communication networks.
- **Models are self-documenting**  
The documentation tool generates high-quality documentation from commented model source code, with diagrams, tables and cross-references. Integrates well with the Doxygen C++ documentation tool.
- **Source code is provided**  
Well-documented source code to increase your understanding, help debugging, and enable modifications is provided
- **Standards support, open interfaces**  
Input and output can be plain text and/or XML, making it easy to process with 3rd party tools. Database integration is also possible.
- **Graphical tools for simulation building and evaluating results**  
Apart from the GUI, command line, batch and API access is provided to all



## the OPEN Simulator

features, allowing for extremely versatile and powerful simulation control and management.

- **Powerful GUI for tracing, debugging and animating your simulations**  
In addition to animation and sophisticated logging, you can peek into objects and variables in the C++ code, even change them on the fly.
- **Responsive and expert support**  
Your requests are answered directly by the developers, not by techsup personnel.

### Network simulation

#### **Wide range of open-source simulation models**

TCP/IP, IPv6, MPLS, wireless, mobile and ad-hoc simulations, TinyOS, port of the FreeBSD networking stack, queuing and several other open-source models are available from [www.omnetpp.org](http://www.omnetpp.org) and other websites.

#### **Emulation / Real-Time simulation / hardware-in-the-loop simulation support**

The simulation kernel supports real-time and hardware-in-the loop simulation via a plugin interface. A functioning and extensively commented source code example will help you to quickly implement your own application-specific hardware-in-the-loop simulation.

### Advanced

#### **Embeddable simulation kernel**

The simulation kernel can be embedded into 3rd party products as a C++ class library. See our References for companies who have already done that.

#### **Parallel simulation**

Parallel simulation on clusters or multiprocessors, with MPI and powerful conservative synchronization. Using named pipes or other communication means instead of MPI is also possible.

#### **Extensibility**

C++ plug-in interfaces are made available to customize features of the simulation kernel.

#### **Database support**

Network topology and model configuration may come from a database, and results can be recorded into a database -- without changing a line in the model itself.

the OPEN Simulator



## Company Information

Simulcraft Inc.  
Crystal Offices, OT Center  
Victoria, Mahé  
Republic of Seychelles

Phone: +36 20 3535 116  
Fax: +44 20 7681 1502  
E-mail: [info@omnest.com](mailto:info@omnest.com)

Authoried Distributor & support partner

ViRE Technologies Pte Ltd  
Tel: +65 6100 4310  
Fax: +65 6100 1370  
Web: [www.vi-re.com.sg](http://www.vi-re.com.sg)