



## NETWORK SIMULATION

### 2.1. Steps Involved In Developing A Simulation

In general the steps involved in developing a simulation model, designing a simulation experiment, and performing simulation analysis are:

**Step1:Identify the problem:** this step consists of defining the problem that we must solve.

**Step2:Formulate the problem:** This involves identifying the controllable and uncontrollable inputs, identifying constraints on the decision variables, defining a measure of performance.

**Step3:Collect and process real system data:** Decide what data to collect about the real system,and how much to collect. This decision is a tradeoff between cost and accuracy.

**Step4:Formulate and develop a model:** Acquire sufficient understanding of the system to develop an appropriate conceptual, logical model of the entities and their states, as well as the events codifying the interactions between entities (and time). This is the heart of simulation design.

**Step5:Validate the model:** The term validation is applied to those processes that seek to determine whether or not a simulation is correct with respect to the “real” system. More precisely, validation is concerned with the question “are we building the right system?”

**Step6:Select appropriate experimental design:** The step trying to run the simulation with different input parameters (i.e., under different scenarios”) to see how performance measures are affected.

**Step7:Establish experimental conditions for runs.**

**Step8:Perform simulation runs.**

**Step9:Documents model for future use.**

**Step10:Interpret and present results.**



---

## Step11:Recommend further course of action.

Although this is a logical ordering of steps in as imulation study, many iterations at various sub-stagesmay be required before the objectives of a simulation study are achieved. Not all the steps may be possible and/or required. On the other hand, additional steps mayhave to be performed.

### 2.2 Structure of a Simulator

- Simulation engine (core): Engine is the most important part in simulator . this part is responsible about specifies how to handle simulations ( everything is handled in uniform way).
- Collection of models (link, buffer, etc.).
- Frontend (interface).
- Preprocessing tools (hidden for a user).
- Data post-processing tools.

### 2.3. Areas of Application

- Manufacturing Applications
- Semiconductor Manufacturing
- Construction Engineering and project management
- Military application
- Logistics, Supply chain and distribution application
- Transportation modes and Traffic
- Business Process Simulation



- Health Care
- Automated Material Handling System (AMHS)
  - Test beds for functional testing of control-system software
- Risk analysis
  - Insurance, portfolio,...
- Computer Simulation
  - CPU, Memory,...
- Network simulation
  - Internet backbone, LAN (Switch/Router), Wireless, PSTN (call center),...

## 2.4 Simulation versus Emulation

At the beginning we must understand the mean of emulation and simulation with respect to each other.

The term "emulation" comes from the verb "emulate," which means to imitate or reproduce. Therefore, computer emulation is when one system imitates or reproduces another system. This can be done using hardware, software, or a combination of the two. However, since hardware is expensive to reproduce, most emulation is done via software. Therefore, the purpose of emulation is to mimic the original system and reproduce the every event that happens in every network elements and applications. From this point, **Emulation** can be defined as the process of imitating a hardware/software program/platform on another program or platform. This makes it possible to run programs on systems not designed for them. **Emulators**, as the name implies, emulate the functions of one system on another. Thus, the second system behaves like the original system, attempting to exactly reproduce the external behaviors of the first system.

A simulator, on the other hand, sets up a similar environment to the original device's OS, but doesn't attempt to simulate the real device's hardware. Some programs may run a



little differently, and it may require other changes (like that the program be compiled for the computer's CPU instead of the device's), but it's a close enough match that you can do most of your development against the simulator. In simulation, the goal is to generate statistical results that represent the behavior of certain network elements and their functions. The differences between the emulator and simulator can be listed as shown in table 1:

**Table 1:** Differences between Emulator and Simulator

NO	Emulator	Simulator
1	emulator is one that replicates the usage as on the original device or system	A simulator is an environment which models the behavior of real system
2	An emulator will always have to operate close to real-time	Simulator operate using simulated time
3	An emulator can replace the original for 'real' use Lets consider	A simulator cannot be substituted with real system
4	An emulator works by duplicating every aspect of the original device's behavior. It basically simulates all of the hardware the real device uses, allowing the exact same software to run on it unmodified.	A simulator, sets up a similar environment to the original device's OS, but doesn't attempt to simulate the real device's hardware
5	Emulation is a replica of the internal system functions on a different host (for example on a Mac OS X I can write a software emulator of a Windows OS where a Win program can run). Emulators can also be strictly hardware-based and is normally based on a partial or complete "reverse engineering" phase.	Simulation is based on a software implementation of a model where the internal functions of the original systems are not taken into consideration (for example a "flight simulator" does not have any "component" of an actual aircraft).
6	Emulator is exactly like something and adheres to all of the rules of the system being emulated	Simulator is similar to somethingbut is implemented in an entirely different way
7	Faster	Slower

