Chapter 21
Evaluation of Simulation Models

Sattar J Aboud
Iraqi Council of Representatives, Iraq

ABSTRACT
Simulation is employed widely to evaluate complicated systems involving telecommunication networks. This chapter reviews and evaluates different types of discrete event simulation models that are widely used for modeling network models in a practical time-frame. Different methods are presented, but special attention is given to systems that use spatial decomposition. This involves data that may have to be transmitted by mediator tiles to its purpose depending on the decomposition post. For congruent simulation the challenge is to decompose the tool in order to make effective use of the original processor design. This chapter reviews a number of methodologies and architectural design that have been developed for efficient simulation model decomposition.

INTRODUCTION
Simulation is the implementation of a model, denoted by a program code that provides information regarding the scheme being inspected. Simulation models are employed to carry out tests that are costly, risky or time consuming to be performed using analytical or experimental approaches. There are a number of methodologies that can be used to classify the existing simulation models. In particular, simulation models can be classified according to one of the following criteria, which are widely-used in classifying the different types of simulation models: time-variation of the state of the system variables, simulation termination procedure, and input-traffic pattern. Based on the time-variation of the state of the system variables, simulation models can be classified into continuous-valued or discrete-event (Al-Bahadili, 2010; Sinclair, 2004; Law & Kelton, 2000; Roth, 1987). Discrete simulation is the modeling of a scheme; it changes over time by which state variables alters in time at a countable number of points. At these stages an event is happen and is...
taken to be an immediate event this could alter the situation of a scheme. The standard simulation of packet-switched networks typically includes the use of discrete event simulators which is model every separate packet during the network, usually entitled packet-level simulation. Every packet departure from or arrival at a network position is denoted by an event. Too big number of packets should be simulated to get trust in the results. This needs long simulation period, usually totaling many hours to simulate a few packets.

**Discrete Event**

Traditionally, discrete-event system simulations have been done in a sequential manner. A variable clock holds the time up to which the physical system has been simulated. A data structure, called the event list, maintains a set of messages, with their associated times of transmissions, that are scheduled for the future. Each of these messages is guaranteed to be sent at the associated time in the physical system, provided the sender receives no message before this message transmission time. At each step, the message with the smallest associated future time is removed from the event list, and the transmission of the corresponding message in the physical system is simulated. Sending this message may, in turn, cause other messages to be sent in the future, which then are added to the event list or cause previously scheduled messages to be canceled which are removed from the event list. The clock is advanced to the time of the message transmission that was just simulated (Moorsel, 1991).

This form of simulation is called event driven, because events, message in the physical system are simulated chronologically and the simulation clock is advanced after simulation of an event to the time of the next event. There is another important simulation scheme, time-driven simulation in which the clock advances by one tick in every step and all events scheduled at that time are simulated.

Discrete event simulation model is dynamic physically, thus there is a requirement for a method to move forward the simulation time from one event to another. This needs a simulation clock which provides the existing state of the elapsed simulated value. Also, certain means of saving the present predictable events queued to happen within the scheme is needed. This data structure usually takes the type of a specific time ordered cycle of event minutes title an event-list. The simulation clock can be later consistent with either of two main time-progression policies, entitled fixed-increment time progression. The fixed-increment time progression technique of time has been commonly employed in the simulation of schemes such that events are known to happen at fixed regular time.

The simulation clock increase by constant numbers, named ticks. After every tick, a test is generated to determine when any events must happen in the preceding time period. When one or more events are queued, they are considered to happen at the end of the period (Walrand, 1987). The scheme situation is then updated. The system can be unproductive when the tick degree is not carefully measured. Selecting a small value decreases the chance of events happening with a tick age and thus raises the possibility of simulator operation cycles that have no practical event work. Choosing a big tick time decreases the event time solution. With subsequent event time progression the times of happening of next events are created. Then, the simulation clock is moved to the time of the initial event, which is then handled. The scheme is updated to display the results of this event then the simulation clock is moved to the following most pending event and so on. This is the most general method used, since times of used in the scheme are neglected because the entire state modifies only happen at event times. This exclusion of used periods permits the simulation to carry on effectively, particularly if the time between following events is big.