

Introduction of an Integrated Environment for Modeling and Evaluation of Performance of Computer Networks

Rahim Rashidi
Department of Computer, Boukan
Branch, **Islamic Azad University**,
Boukan, Iran
rahim.rashidy@gmail.com

Keyhan Khamforoosh
Department of Computer, Sanandaj
Branch, **Islamic Azad University**,
Sanandaj, Iran
k.khamforoosh@iausdj.ac.ir

Farooogh Ashkavti, Behnam Salehyazad
Department of Computer, Mahabad Branch,
Islamic Azad University, Mahabad, Iran
ashkooti_f,behnam.salehyazad@yahoo.com

ABSTRACT

SimEvents extends Simulink with a discrete-event simulation (DES) model of computation. With SimEvents you can develop activity-based models of systems to evaluate system parameters such as congestion, resource contention, and processing delays. You can configure entities with user-defined attributes, and then aggregate entities and attributes to model data hierarchy and transport in applications such as packet-based networks, mission planning, supervisory control, real-time operating systems, and computer architecture. In this paper we introduce a comprehensive integrated environment for modeling and evaluation of computer network.

Keywords

Evaluate the performance, computer networks, modeling discrete event, MATLAB, SimEvents, Simulink

1. INTRODUCTION

Due to the increasing use of computer networks, the need for a comprehensive integrated environment for modeling and evaluation of computer networks and telecommunications is inevitable. It is a way to understand the needs of the system and network performance modeling system reliability necessary for induction it seems. In this paper, components for computer networks, as well as how to build networks with tools SimEvents are reviewed [1].

SimEvents software incorporates discrete-event system modeling into the Simulink® time-based framework, which is suited for modeling continuous-time and periodic discrete-time systems. In time-based systems, state updates occur synchronously with time. By contrast, in discrete-event systems, state transitions depend on asynchronous discrete incidents called events [1].

With regard to the visual environment, hierarchical structure tools, dynamic environment and network implementations in SimEvents, we show that SimEvents provide an integrated and comprehensive environment for modeling, computer networks assessment.

2. WHAT IS DISCRETE-EVENT SIMULATION?

Informally, a discrete-event simulation, or event-based simulation, permits the system's state transitions to depend on asynchronous discrete incidents that are called events. By contrast, a simulation based solely on differential equations in which time is an independent variable is a time-based simulation because state transitions depend on time. Simulink is designed for time-based simulation, while SimEvents is designed for discrete-event simulation. Your choice of a different simulation style can depend on the particular phenomenon you are studying and/or the way you choose to study it. Some examples illustrate these differences [1]:

Suppose you are interested in how long the average airplane waits in a queue for its turn to use an airport runway, but not

interested in the details of how an airplane moves once it is cleared for takeoff. You might use discrete-event simulation in which the relevant events include the approach of a new airplane to the runway and the clearance for takeoff of an airplane in the queue.

Suppose you are interested in the trajectory of an airplane as it takes off. You would probably use time-based simulation because finding the trajectory involves solving differential equations.

Suppose you are interested in how long the airplanes wait in the queue but you want to model the takeoff in some detail instead of using a statistical distribution to model the length of time that each plane uses the runway. You might use a combination of time-based simulation and discrete-event simulation, where the time-based aspect controls details of a plane's takeoff and the discrete-event aspect controls the queuing behavior[1][2][3].

3. MODEL AND SIMULATE DISCRETE-EVENT SYSTEMS

SimEvents provides a discrete-event simulation engine and Component library for Simulink. You can model the event-driven communication between the components to analyze and optimize end-to-end latencies, throughput, packet loss, and other performance characteristics. Libraries of predefined Blocks, such as queues, servers, and switches, enable you to accurately represent your system and customize routing, processing delays, prioritization, and other operations.

With SimEvents you can design the distributed control systems, Hardware architectures, and sensor and communication networks for aerospace, automotive, and electronics applications. You can also simulate the event-driven processes, such as the execution of a mission plan or the stages of a manufacturing process, to determine the resource requirements and identify bottlenecks [2][1].

4. KEY FEATURES OF SIMEVENTS

- Discrete-event simulation engine for multidomain modeling of complex systems in Simulink
- Predefined block libraries, including queues, servers, generators, routing, and entity combiner/splitter blocks
- Entities with custom data attributes for flexible representation of packets, tasks, and parts
- Built-in statistics aggregation for obtaining delay, throughput, average queue length, and other metrics
- Library blocks for defining domain-specific constructs, such as communication channels, messaging protocols, and conveyor belts
- In-model animation for visualizing model operation and debugging

5. DEFINITION OF TERMS IN SIMEVENTS

What is an Entity?

Discrete-event simulations typically involve discrete items of interest. By definition, these items are called entities in SimEvents software. Entities can pass through a network of queues, servers, gates, and switches during a simulation.

Entities can carry data, known in SimEvents software as attributes.

Examples of entities in some sample applications are in the table [1][2][6].

Context of Sample Application	Entities
Airport with a queue for runway access	Airplanes waiting for access to runway
Communication network	Packets, frames, or messages to transmit
Bank of elevators	People traveling in elevators
Conveyor belt for assembling parts	Parts to assemble
Computer operating system	Computational tasks or jobs

A graphical block can represent a component that processes entities, but entities themselves do not have a graphical representation. When you design and analyze your discrete-event simulation, you can choose to focus on:

- The entities themselves. For example, what is the average waiting time for a series of entities entering a queue?
- The processes that entities undergo. For example, which step in a multiple-step process (that entities undergo) is most susceptible to failure?

What is an Event?

In a discrete-event simulation, an event is an instantaneous discrete incident that changes a state variable, an output, and/or the occurrence of other events. Examples of events that can occur during simulation of a SimEvents® model are: The advancement of an entity from one block to another [1].

- The advancement of an entity from one block to another
- The completion of service on an entity in a server
- A zero crossing of a signal connected to a block that you configure to react to zero crossings. These events are also called trigger edges
- A function call, which is a discrete invocation request carried from block to block by a special signal called a function-call signal

Relationships among Events

Events in a simulation can depend on each other:

One event can be the sole cause of another event.

For example, the arrival of the first entity in a queue causes the queue length to change from 0 to 1.

One event can enable another event to occur, but only under certain conditions. For example, the completion of service on an entity makes the entity ready to depart from the server. However, the departure occurs only if the subsequent block is able to accept the arrival of that entity. In this case, one event makes another event possible, but does not solely cause it [1] [6] [9].

Events that occur at the same value of the simulation clock are called simultaneous events, even if the application processes sequentially. When simultaneous events are not causally related to each other, the processing sequence can significantly affect the simulation behavior.

Role of Attributes in SimEvents Models

You can attach data to an entity using one or more attributes of the entity. Each attribute has a name and a numeric value. You can read or change the values of attributes during the simulation.

For example, suppose your entities represent a message that you are transmitting across a communication network. You can attach the length of each particular message to the message itself, using an attribute named length.

6. SIMEVENTS LIBRARY

SimEvents library includes block shown in Figure 1. The user can also create and add the desired block.



Fig 1: SimEvents Library

7. CREATING NEW BLOCKS IN THE TOOL OF SIMEVENTS

In SimEvents by block shown in Figure 2, you can create new blocks. After double-clicking on it, the edit window appears and you can change the number of inputs and outputs based on the workflow within the programming instructions written in MATLAB.

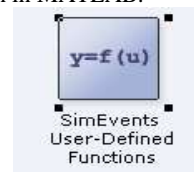


Fig 2: User-Define Function

8. CREATE NEW BLOCKS IN THE TOOL SIMULINK

In Simulink Library by block shown in Figure 3, user can create new blocks.

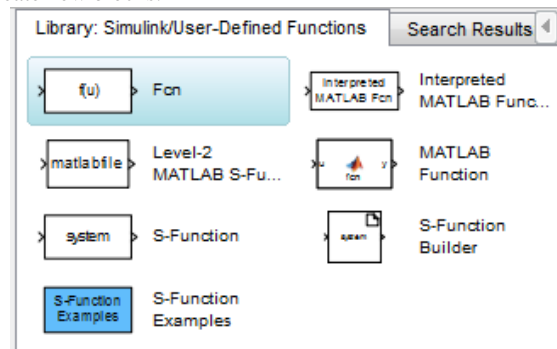


Fig 3: Simulink Library, User-Define Block

After double-clicking on block of interest, as shown in Figure 4, the edit window appears and you can change the number of inputs and outputs based on the workflow within the programming instructions written in MATLAB. Section edit it as follows [1].

```

1 function y = fcn(u)
2 % This block supports the Embedded MATLAB subset.
3 % See the help menu for details.
4
5 y = u;

```

Fig 4: Edit Window

9. COMMUNICATION PROTOCOL MODELING IN ETHERNET LAN

In this simulation it is assumed that a computer site includes a workgroup network, and in which each node is both a client and a server.

This simulation shows how to model an Ethernet LAN with three computers and a shared channel operated at 10 Mbps. The MAC Controller block implements the CSMA/CD (Carrier Sense Multiple Access with Collision Detection) protocol [1] [7].

Structure of the Model

The top level of the model includes three computers, each consisting of the following:

- An Application block that models the consumer of data
- A MAC (medium access control) controller that governs the computer's use of the shared channel
- A T-junction that connects the computer to the network.

You can specify the packet generation rate and packet size range at the Application blocks, the transmission buffer size at the MAC Controller blocks, and the length of the cable at the Cable blocks. The Terminator, T-junction, and Cable Blocks at the bottom of the model represent physical components of the network [1] [7]

Implementing the CSMA/CD Protocol

Standard Ethernet networks use a CSMA/CD protocol to manage use of the shared channel. Each computer's MAC controller physically monitors the traffic on the channel and initiates a transmission only when the controller detects no other traffic on the channel. If transmissions from two computers compete for use of the channel, then a collision occurs. Packets that collide with other packets can make a fixed number of additional attempts after a waiting period elapses, using the binary exponential backoff algorithm.

In this model, the CSMA/CD protocol is implemented in the MAC Controller subsystems [1] [7].

Setting Bandwidth (b/s) Block

Block network bandwidth is shown in Figure 5. In this model the network bandwidth is considered 10Mbps.

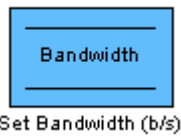


Fig 5: Set Bandwidth Block

Bandwidth can be changed if needed. In this model 1e7 is equivalent to 10Mbps.

Application Block

Application block model's application layer. Figure 6 is an application in. high-level.

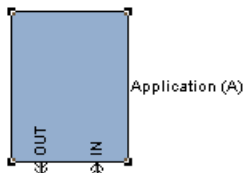


Fig 6: Application Block

Cable Block

Cable blocks are shared channels. In any network two formulas are calculated: Propagation delay and transmission delay

Transmission Delay $T = L / B$

Depending on the length L and B are the speed in Mbps.

Cable propagation delay is calculated as follows:

Channel delay $T = D / V$ where D is the same distance and V - the speed [1] [7].

To simulate the cable a time service provider is used. The delay of the cable is the time of publication.

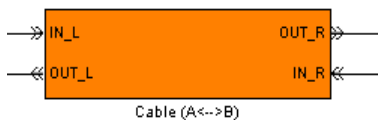


Fig 7: Cable Block

T-junction Block

T-junction connects computers to the network. Figure 8 is a T-junction Block in high-level [1] [7].

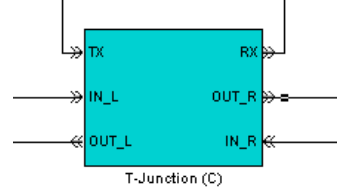


Fig 8: T-junction Block

MAC Block

In this model, the CSMA/CD protocol is implemented in the MAC Controller subsystems.

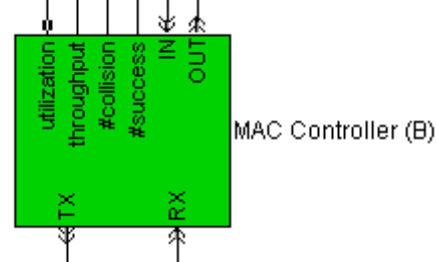


Fig 9: MAC Block

Backoff Controller Block

Backoff Controller Block is as follows. If the collision happens, the blocks runs and a random time produced to Computers to be doing again [1].

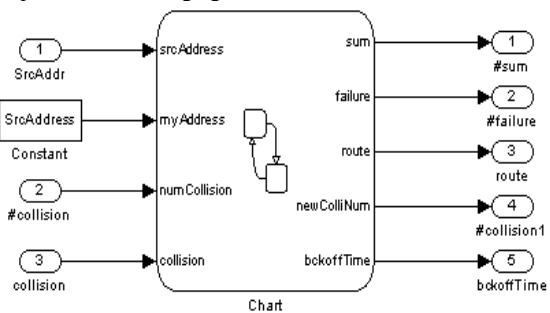


Fig 10: MAC Block

Throughput Block

Throughput Block is as follows. In the following diagram, embedded MATLAB function block in the Simulink library is used to define the functions by the user [1].

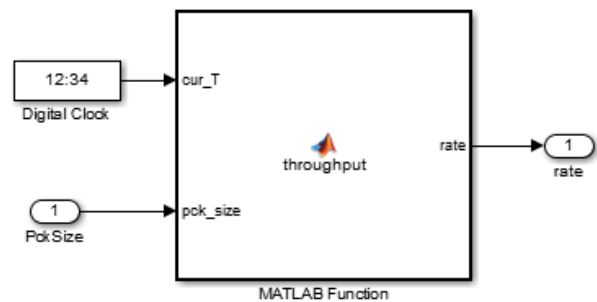


Fig 11: Throughput Block

This function is written as follows:

```

Function rate = throughput (cur_T,
pck_size)
% THROUGHPUT Compute the rate at
which the computer sends data
% INPUT:
%   cur_T      -- Current simulation
time
%   pck_size  -- Size, in bits, of the
packet that is successfully

```

```

%                               transmitted.
% OUTPUT:
%   rate      -- Current throughput
of the computer (in kbps)
Persistent acc_sum;
% Initialize persistent variables in
the first iteration.
If isempty (acc_sum)
    acc_sum = 0;
End
% Update the throughput of current
computer.
acc_sum = acc_sum + pck_size;
Rate = acc_sum / (1000*cur_T);
In the above formula, the simulation time is multiplied in
1000. Because the rate is based kbps.

```

Utilization Block

Utilization Block is as follows

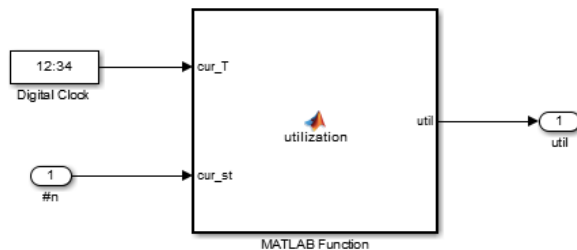


Fig 12: Utilization Block

In this part we use blocks in Simulink library. The blocks are programmable by the user. This function is defined as follows:

```

Function util = utilization (cur_T,
cur_st)
%UTILIZATION Compute fraction of time
that a signal is positive.
% UTIL = UTILIZATION (CUR_T,CUR_ST)
computes the fraction of
% time that the input CUR_ST is
positive, where CUR_T is the
% current simulation time.
Persistent acc_sum last_T last_st;
% Initialize persistent variables in the
first iteration.
If isempty(last_T)
    acc_sum = 0;
    last_T = 0;
    last_st = 0;
End
% Keep running total of the
duration that the signal is
positive.
If (last_st > 0) acc_sum + (cur_T-last_T);
End
% Update the other persistent variables.
last_T = cur_T;
last_st = cur_st;
% Compute ratio.
util = acc_sum / cur_T;
User-defined functions and create block-based input and
output is the one of the MATLAB features.

```

Terminator Block

Use the Terminator block to cap blocks whose output ports do not connect to other blocks. If you run a simulation with blocks having unconnected output ports, Simulink issues

warning messages. Using Terminator blocks to cap those blocks helps prevent warning messages. In the simulation this blocks stop the signal back to the end of the cable. The signals are absorbed by the Terminator and they cannot collide with other signals on the cable.

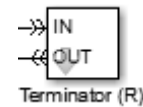


Fig 13: Terminator Block

Results and Displays

This model includes various visualization that capture the performance of this Ethernet bus:

Display blocks that show the number of packets that each computer observes as successful or unsuccessful communications. Propagation delay modeled in the Cable blocks can cause different computers to observe slightly different results.

The Throughput plot shows the transmission throughput at one of the computers.

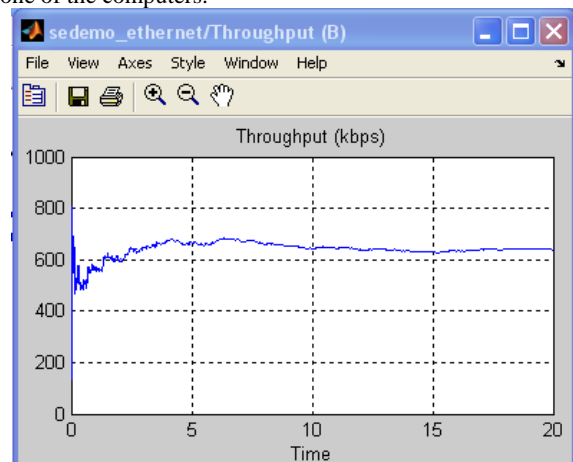


Fig 14: Throughput

The Channel Utilization plot shows the proportion of time that the channel is in use.

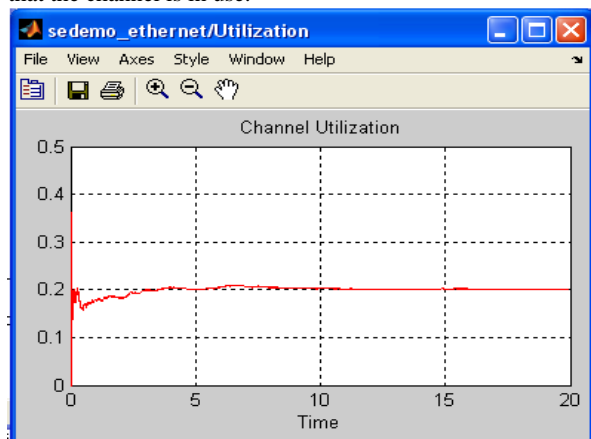


Fig 15: Channel Utilization

Given the capabilities of these tools we can use from SimEvents and Simulink as an integrated environment for modeling and evaluate computer networks. In this article the network model is as follows:

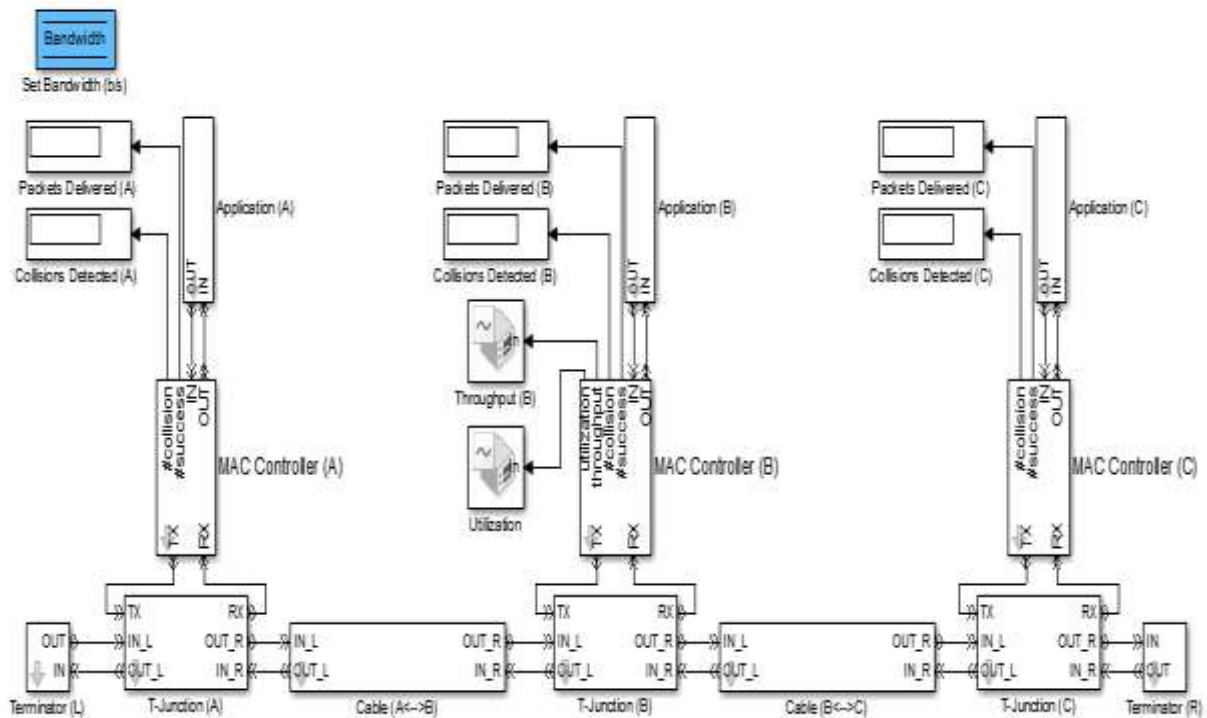


Fig 16: Communication Protocol Modeling in SimEvents

10. CONCLUSION

In this paper we introduce a comprehensive integrated environment for modeling and evaluation of computer network. SimEvents provides a discrete-event simulation engine and Component library for Simulink. You can model event-driven communication between components to analyze and optimize end-to-end latencies, throughput, packet loss, and other performance characteristics. Libraries of predefined blocks, such as queues, servers, and switches, enable you to accurately represent your system and customize routing, processing delays, prioritization, and other operations. With SimEvents you can design distributed control systems, hardware architectures, and sensor and communication networks for aerospace, automotive, and electronics applications. You can also simulate event-driven processes, such as the execution of a mission plan or the stages of a manufacturing process, to determine resource requirements and identify bottlenecks. SimEvents software incorporates discrete-event system modeling into the Simulink time-based framework, which is suited for modeling continuous-time and periodic discrete-time systems. Generally SimEvents covers wide range of important topics in the fields of modeling and evaluation (computer networks, operating systems, modeling and evaluation systems, distributed systems and computer architecture, etc.)

REFERENCES

- [1] Provides tutorials and comprehensive information about SimEvents
- [2] Banks, Jerry, John Carlson, and Barry Nelson, Discrete-Event System Simulation, Second Ed., Upper Saddle River, N.J., Prentice-Hall, 2007.
- [3] Cassandras, Christos G., Discrete Event Systems: Modeling and Performance Analysis, Homewood, Illinois, Irwin and Aksen Associates, 2007.
- [4] Fishman, George S., Discrete-Event Simulation: Modeling, Programming, and Analysis, New York, Springer-Verlag, 2007.

- [5] Zeigler, Bernard P., Herbert Praehofer, and Tag Gon Kim, Theory of Modeling and Simulation: Integrating Discrete Event and Continuous Complex Dynamic Systems, Second Ed., San Diego, Academic Press, 2007.
- [6] Provides tutorials and comprehensive information about SimEvents
- [7] Tanenbaum, Andrew S., Computer Networks, Fourth Ed., Upper Saddle River, N.J., Prentice Hall PTR, 2002.
- [8] Watkins, Kevin, Discrete Event Simulation in C, London, McGraw-Hill, 1993.
- [9] Moler, C., "Floating points: IEEE Standard unifies arithmetic model," Cleve's Corner, The MathWorks, Inc., 1996. You can find this article at <http://www.mathworks.com/company/newsletter/clevescorner/>
- [10] Gordon, Geoffery, System Simulation, Second Ed., Englewood Cliffs, N.J., Prentice-Hall, 1978.