

## A MODELING APPROACH FOR EVENT-BASED NETWORK DESIGN USING MATLAB-SIMEVENTS

E. Harahap<sup>1</sup>, A. Nurrahman<sup>2</sup>, D. Darmawan<sup>3</sup>

<sup>1</sup>*Department of Mathematics, Universitas Islam Bandung, Indonesia*

<sup>2</sup>*Department of Industrial Engineering, Universitas Islam Bandung, Indonesia*

<sup>3</sup>*Department of Information and Communication Technology, Universitas Pendidikan Indonesia*

erwin2h@unisba.ac.id<sup>1</sup>, arif.nurrahman@unisba.ac.id<sup>2</sup>, ddarmawan@upi.edu<sup>3</sup>

---

### Abstract

*The present development of technology revealed that practices in the fields of science and engineering associated with a complex event-based system design and its implementation requiring an accurate modeling method. The reliability of the method takes a very important role in the system's verification, modification, and implementation to allow a scientist or administrator to predict the result from changes to the system. Modeling and simulation generally have been used to understand the engineering and operation of a system which represented as the chronological sequence of events as a basic method of a Discrete-Event Simulation (DES). A complex system of queuing models, flow control, communications networks, and others can also be developed and analyzed using DES. This paper describes an application tools of MATLAB-SimEvents as one of modeling method with event-based simulation as a form of simplification approach for the development complex systems. The demonstration of the application is applied to the Internet system, i.e. Content Delivery Network (CDN). The discussion presented in this paper is expected to support researchers and practitioners in understanding the basics of modeling and simulation in general contexts.*

**Keywords:** *SimEvents, Modeling, Simulation, Networks, CDN*

---

### INTRODUCTION

Modeling generally used to understand the engineering and operation of a system [1] utilizing a simulation tools such as ns2/ns3 [2], Opnet Modeler [3], etc., which is used to model a complex discrete system. A simulator has its own characteristics to model and simulate an expected system such as its conformity with the experimental design so that a model can be easily designed and simulated [5]. Modeling using blocks of modular and hierarchical as provided in SimEvents MATLAB, has been used in many applications of various techniques such as hardware design, communication systems, manufacturing systems, electrical engineering, and sciences.

SimEvents is a Discrete-Event Simulation (DES) tool developed by MathWorks [1]. Some common uses of SimEvents includes a simulation process and a modeling performance of the system architecture. SimEvents adds a library of graphical building blocks for modeling a queuing system to the Simulink environment [6], and an event-based simulation engine to the time-based simulation engine in Simulink. This work seeks to develop a DES process model which shows the internal behavior of process in system model.

In the general simulation process, SimEvents is used to model process flows and logistics to understand resource availability, inventory management techniques, and the effects of arbitrary events on a mission critical network design. System capacity planning and production planning can be simulated while ascertain its optimal performance. A SimEvents model can be used to estimate through-life support (TLS) requirements such as the size of buffer switches and provisioning of services. A SimEvents model can also predict characteristics such as availability, serviceability and end-to-end latencies based on a specific maintenance schedule [7].

In the modeling performance of a system architecture, SimEvents is used to build a discrete-event model of the architecture of a computer hardware system. This architecture model simulates the transactions between different components such as processors, memory, or a communication bus. Simulation of this model provides plots of performance characteristics such as throughput, latency, packet loss and utilization [8].

In this paper, SimEvents is used to model and simulate the Internet network systems that delivered contents based on the client's proximity, which is more known as a Content Delivery Network (CDN). We model and simulate the CDN in the form of links, clients, routers, and data centers. We discussed the realization of models and simulation and performance analysis system, and then finally we present our conclusions.

## Related Work

SimEvents have a more user friendly interface compared to DEVS-Suite and Ptolemy II simulators [2] for simulation model and complexity evaluation, and add to visual complexity metrics as the numbers of components that are displayed for a given model in graphical plots. Monitoring inputs and output components in DES models are much easier with SimEvents than other tools.

Mathworks [3] presents SimEvents as an extension to MATLAB Simulink with tools for discrete-event simulation of the transactions between components in a system architecture. The author in [12] discussed modeling and simulations presented a SimEvents about a reengineered data center network (DCN), while the work in [13] presented a SimEvents based transport system model (rail yard) that allows a simulation movement of three trains as discrete events. The architecture of SimEvents and its functionality was presented in [10].

## Modeling and Simulation

Modeling is defined as the process of producing a model and argues that a model is a representation of the construction and working of some system of interest [12]. The major objective of a model is to enable an expert to predict the effect of changes to the system. In this case, the model should be a close approximation to the real system and incorporate most of its salient features as well as not being unnecessarily complex such that it is impossible to understand and experiment with it.

A simulation of a system model is literally the envisioned operation of a model of such a system. Technically, simulation is a method used to evaluate the performance of a system (existing or proposed) under some different configurations of interest over long periods of time. Simulation is used before an existing system is altered or a new system built to reduce the chances of failure to meet specifications, to eliminate unforeseen bottlenecks, to prevent under or over-utilization of resources, and to optimize system performance [12]. Basically, a model can be reconfigured and experimented with, but usually, this is impossible, too expensive or impractical to do in the real system it represents refer to some obvious factors. Besides, if the operation of the model can be clearly studied, hence, the properties concerning the behavior of the actual system or its subsystem can be inferred.

## SimEvents Implementation in Modeling CDN

By leveraging on the capabilities offered by MATLAB SimEvents such as its graphical drag-and-drop interface for building a discrete-event model, in our implementation approach of modeling CDN, the key features explored includes: *Entity-based modeling* for representing packets; *Attributes* for attaching scalar, matrix, and complex data to packets; Libraries of predefined blocks, such as queues, servers, routing, and generators, for modeling system architecture and process flow diagrams; Built-in statistics such as delay, throughput, and average queue length; Library block for writing algorithms to customize operations such as routing, processing delays, and prioritization; Entity and attribute aggregation blocks for modeling data hierarchy; Hybrid simulation capabilities for models that contain both event-based and time-based components.

SimEvents integration with MATLAB allows customization of the process flow in our implementation model. In this context, a MATLAB function can be developed to represent a task-

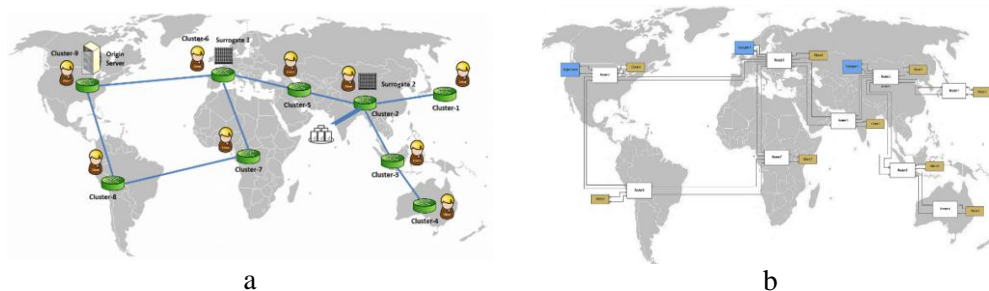
scheduling sequence, switching and routing of traffic in a process flow model. With SimEvents and MATLAB Integration within the same tools environment, it allows for a flexible generation of custom random distributions of input tasks as well as custom data statistics from a SimEvents model using a MATLAB command line interface.

CDNlink is created with two advantages. First, CDNlink creates on MATLAB-Simulink which provides a visual graphical method when addressing problems associated with designing complex control, signal processing, and communication systems. Simulink provides an efficient approach for establishing a common framework for communication throughout the design process of network system.

The modeling design using SimEvents is significantly different from traditional design methodology. Rather than using complex structures and extensive codes, by SimEvents we use Simulink design to define models with advanced functional characteristics comprising of continuous-time and discrete-time building blocks. Next, SimEvents is constructed as a model-based network simulator. A model-based design is a system that used a mathematical model of addressing problems associated with designing a network system, queueing model that has been implemented as a basic rule of network simulation. Another mathematical model can be simply implemented as part of the network system process in addition to the queueing model, or as an algorithm to improve a network system.

## Modeling CDN and Architecture

This paper develops a discrete event based process model and simulates a CDN system comprising of one origin server, one DNS server, two Data Center, 9 routers, and 9 client nodes. MATLAB SimEvents was employed in the model using full duplex traffic flow while processing queues in origin server and data centers to avoid packet losses and congestion in the CDN when scalability consideration arises. Figures 6a and 6b shows the CDN Topology and the model architecture developed in this paper.



Figs. 6a. The topology of the CDN System. 6b. The model architecture of CDN in SimEvents

The architecture of the CDN system shown by figure 6b was implemented with SimEvents. Using MATLAB SimEvents, we developed figure 6b as a simple CDN communication model that uses a UDP protocol to manage use of the shared channel for high speed traffic computation. Each CDN nodes physically monitors the traffic on the channel and initiates a transmission. With this access technique, channel contention is less likely to occur as transmissions from multiple client nodes, which are in excess full duplex mode, is transmitted to a less congested servers or Datacenters. Packets queue is managed on an application queue block.

The top-level model includes of four subsystem nodes: Clients, Routers, DNS, and Servers. Each of node consisting of the following: an application data packet block that models the packet source data with its interface to a DNS, Routers, and Servers. Links is used to connects one node to another in CDN's subnets networks.

**Client**

We design a Client node (Fig. 3) that consists of three units: packet generation, processing packet, and sink. A queuing model can be added into the input section for packet arrival rate, and in process section for packet service rate. Packet arrival is processed and then forward to a destination through a router and received replies.

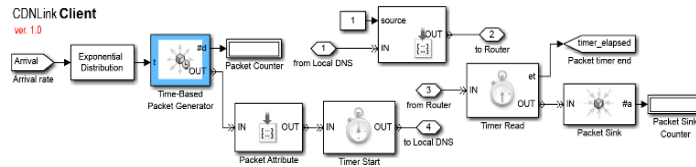


Fig. 3. The design of Client for CDNlink

**DNS and Local DNS**

Packet from a client forwards initially to a local DNS Fig. 4(a). A Switch block is functioned to evaluate the availability of a name server for the requested content from the client. If the local DNS have the record of the name server, the request packet will be send back or replied to the client with an additional information of the name server destination. Otherwise, if the record of the name server is not available, the packet request will be forwarded to an authoritative DNS server through a router.

The design of Authoritative DNS (called as DNS) is depicted on Fig. 5. Packet from local DNS is forwarded to DNS to obtain name server information per the request. DNS is aware for the condition for each server under its management, and choose an appropriate name server for each request. Load balancing algorithm is also possible to be included into this module. After the appropriate name server is added, the “updated” request packet is forwarded to the source through a “Packet Forwarder” module.

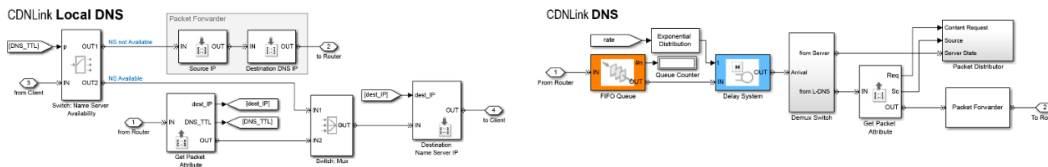


Fig. 5. The design of DNS for CDNlink.

**Router**

The configuration design of router is depicted in Fig. 6. The “Path Combiner” module receives packets from 4 different network nodes: client, local DNS/DNS, other routers, and servers. “System Delay” module is attached to simulate a system delay when packet is managed in the router. Next, after router evaluate the destination of the packet, the system forward the packet to the next hop by “Packet Forwarder” module. The output possibility from the router is similar with the packet received by the router. The packet is possible to be forwarder to 4 destinations: client, local DNS, other routers, or servers.

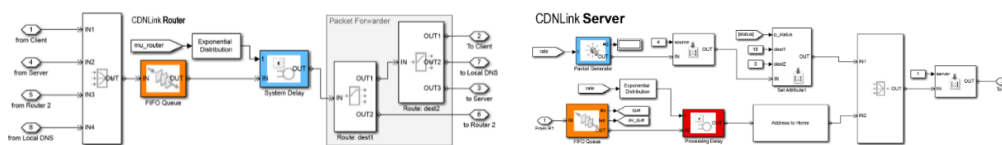


Fig. 6. The design of Router for CDNlink

**Server**

The configuration design of server is depicted in Fig. 7. Arrival packets are stored in “FIFO Queue” block before get served. The “Processing Delay” block simulate the amount time of the server processing the requests and do replies.

## Simulation Scenario and Performance Evaluation

We design the CDN system using the M/M/1 queueing model where packet arrivals from each of packet generator block in each of client are determined by a Poisson process with rate  $\lambda$ . The packet service times in origin servers and data centers have an exponential distribution with rate  $\mu$ . A single server processes the packet data one at a time with first-come first-served discipline.

Packet request was generated from each client to the origin server as destination request. When the origin-server congested, indicated by the origin's average queue-length exceeded 80% of buffer capacity, requests are redirected to the nearest surrogate server.

Display blocks that show the number of packets that each CDN node observes as a latency communication. Propagation delay modeled in the links subsystem can cause different nodes to observe slightly different results. The throughput plot shows the transmission throughput at the DCN server. The server utilization plot shows the proportion of time that the server is in use allocating resources.

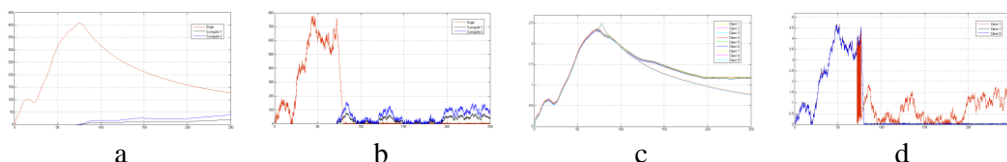


Fig. 10. (a) Average QueueLength; (b) Number of packet in queue from Origin and Surrogate servers; (c) Average RTT of clients (d) RTT of packet elapsed of clients.

## CONCLUSION

We have presented a network modeling by SimEvents approach for developing the model of Internet specifically in CDN system that scales well in today's high speed Internet application. The developed models provide an end to end reliable packet delivery in a simple UDP protocol that regulates losses and enhances a fair loads distribution in between datacenters. The simulation runs for Poisson process packet arrival with exponential distribution in servers

SimEvents provides procedures and functions that needed for simulations in the form of blocks and links made it effective and efficient to design a model specifically for network modeling. The work confirms that the packet transmission from sources to destinations have effective throughput, latency, and efficient server utilization. Figure 9 shows our simulation runs for Poisson process packet arrival with exponential distribution in origin server and the two data centers. Before the simulation runs, link consistency and other block set configurations were verified.

## Acknowledgement

This work is partially supported by funds of the Academic Affairs of Universitas Islam Bandung and the Research Department of Universitas Islam Bandung (LPPM-UNISBA) 2016.