INSIST Vol. 1 No. 1, October 2016 (30 - 33)
http://insist.unila.ac.id
DOI:10.23960/ins.v1i1.13

eISSN: 2502-8588

Received: 20/08/2016
Accepted: 19/09/2016
Published online: 01/10/2016

A Model-Based Simulator for Content Delivery Network using SimEvents MATLAB-Simulink

Erwin Harahap[1*], Icih Sukarsih[1], Gani Gunawan[1], M. Yusuf Fajar[1],
Deni Darmawan[2], Hiroaki Nishi[3]

Abstract—Simulation methodology has become popular among network researchers due to the availability of various sophisticated and the flexibility of simulation packages in model construction and the result's validation. The objective of our research is to introduce a discrete simulator tools known as SimEvents toolbox that runs on MATLAB-Simulink with the advantages of graphical user interface (GUI) system and a total customized simulator design. In this paper, we design a model-based network simulator named as "CDNlink". CDNlink is designed based on a queuing model and functioned to study the content delivery networks (CDN). As a result, we have designed the simulator for Client, Router, DNS, and Server. Moreover, we simulate the CDN system for packet round trip time (RTT) and server's queue length.

Keywords**—**simulink, content delivery network, model-based simulation, simevents, cdnlink.

I. Introduction

Network simulation is generally used to verify models, analyses, generalize the results, evaluate the performance or to compare protocols. However, there is a possibility of inaccurate results when using simulation. In order to overcome this situation, it is important to use a credible simulation tool which is easy to use, more flexible in model development, modification and validation, and incorporates appropriate analysis of simulation output data, pseudo-random number generators, and statistical accuracy of the simulation results [1].

Simulation is one of an important method to simulate a model or mechanism that it can be studied for further research [2, 13]. A network simulation is a method of implementing the network with its parameters and properties. Through a simulation, the behavior of the network is studied either by network entities interconnection using mathematical formulas, or by capturing and playing back observations from a network data. Network simulator allows researchers to test the scenarios that are difficult or expensive to simulate in the real world. Particularly, the simulator is useful to evaluate new networking protocols or to test some changes to the existing protocols in a controlled and reproducible environment.

In this paper we design a network simulator named CDNlink as a content delivery network (CDN) simulator [3, 4]. CDNlink is a model based simulator that designed based on a queuing principle, created by SimEvents toolbox and running on MATLAB Simulink platform [5].

SimEvents provides a discrete-event simulation engine and component library for analyzing event-driven system models and optimizing performance characteristics such as latency, throughput, and packet loss. SimEvents is a GUI system that provides queues, servers, switches, and other predefined blocks to model routing, processing delays, and prioritization for scheduling and communication. Using SimEvents we can study the effects of task timing and resource usage on the performance of distributed control systems, software and hardware architectures, and communication networks. Furthermore, it can also be conducted on operational research for making decisions related to forecasting, capacity planning, and supply-chain management.

The rest of the paper is organized as follows. Section 2 discuss the related works highlighting previous works and types of network simulators. In Section 3, we describe a queuing model for the simulation based on M/M/1 principle. Section 4 provides a proposed CDNlink design created on MATLAB-Simulink with the details network variables design. Section 5 discuss the implementation of the CDNlink on a simple mesh network with simulation results. Section 6 provides a conclusion with prospective future works.

II. Related Work

Network simulator is a piece of software that predicts the behavior of a network, without an actual network being present. Network simulation is a method where a program models the behavior of a network either by calculating the interaction between the different network entities such as hosts, packets, signal, etc., using mathematical formula or actually capturing and playing back observations data from a particular network.

The client sent requests to the server through the Internet which consist many types of nodes, routers, redirectors, bridge, etc., based on a rule how the request come through that nodes. In order to more understanding on how the actual network works, a simulation becomes an important rule to represent the work of the actual network by implementing some of the mathematical methods i.e. queuing theory or calculus network. There are many software simulators that offered with its pros and cons. Based on our experiences, SimEvents [5] is one of simulator that based on queueing principle which relatively easy to use when we design a network system. In addition, and we can customize the network system by implementing some algorithms based on a mathematical model.

Previous research has investigated regarding the existing network simulators. NS2 is one of powerful network simulator and has been demonstrated for CDN simulation [6] where the simulator currently updated to NS3. Another simulator that possible to design as a CDN simulation are A Modeler which

[1*]E. Harahap, I. Sukarsih, Gani Gunawan, and M. Yusuf Fajar, are with the Department of Mathematics, Universitas Islam Bandung, Bandung 40116 INDONESIA (correspondence to First Author to provide phone: +62-22-4203368; e-mail: erwin2h@unisba.ac.id).

[2]D. Darmawan is with the Department of Information and Communication Technology, Universitas Pendidikan Indonesia, Bandung, INDONESIA (e-mail: ddarmawan@upi.edu).

[3]H. Nishi is with the Department of System Design, Faculty of Science and Technology, Keio University. Kohoku-ku, Yokohama. JAPAN. (e-mail: west@sd.keio.ac.jp).

previously known as Opnet [7], CDNsim [8] that created based on OMNET++, or PlanetLab [9]. Sarkar et.al. [1] highlighted that a researcher should choose a credible simulator that offers more flexibility in model development, modification and validation, and incorporates appropriate analysis of simulation output data, pseudo-random number generators, and statistical accuracy of the simulation results. This paper is not intended to compare among network simulators. Basically, a network simulator is best when it fits with the simulated tasks.

III. Queuing Model

Queuing theory is the mathematical study of waiting for lines or queues. This approach is applied to different types of problems, in particular to design a network system [5]. There is some queuing model that can be implemented into a network system which generally divided into a deterministic and probabilistic model. In this paper we focus our network design with M/M/1 model [10, 11, 15].

M/M/1 is a queuing system where packet arrivals are determined by a Poisson process with rate λ and the packet service times have an exponential distribution with rate μ. A single server process the packet data one at a time with the first-come-first-served discipline. When service is complete, the packet leaves the queue and the number of the packet in the system is reduces by one. The buffer is of infinite size, so there is no limit on the number of the packet it can contain.

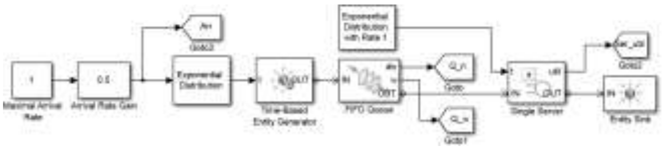


Fig. 1. A simulation design for M/M/1 queuing model which provides theoretical and simulation results.

In an M/M/1 queuing model, a number of packets (N) in the system in steady state situation is formulated by the following equation

$$N = \frac{\rho}{1-\rho} \tag{1}$$

Where $\rho = \lambda/\mu$ is a system utilization [10]. Moreover, according to Little's theorem, the average number of the packet in the queue (N_q) can be formulated by $N_q = \lambda W$ [10]. Alternatively, N_q can be generated by the following argument:

$$N_q = N - \rho = \frac{\rho}{1-\rho} - \frac{\rho(1-\rho)}{1-\rho} = \frac{\rho^2}{1-\rho} \tag{2}$$

Consequently, the packet waiting time (W) can be found based (2) as follow:

$$W = \frac{N_q}{\lambda} = \frac{\rho^2}{\lambda(1-\rho)} = \frac{\rho}{\mu-\lambda} \tag{3}$$

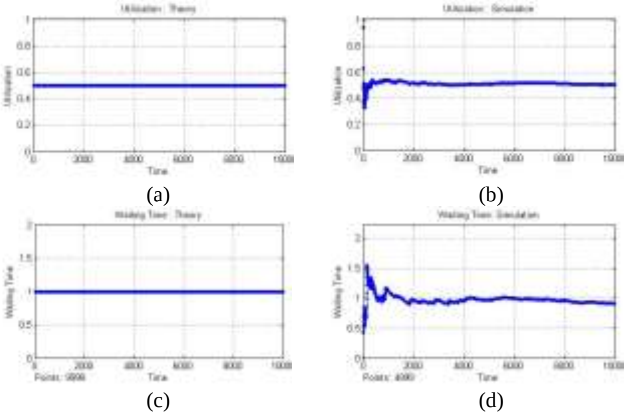
A simulation design for M/M/1 model is depicted in Fig. 1 [5]. Poisson arrival rate represents the rate signal gain with multiplier factor 1. The output signal is an input for time-based packet generator. An exponential distribution of service rate represents by rate signal with multiplier factor 1. The comparison results of system utilization and system-queue of waiting time between theoretical and simulation are depicted in Fig. 2 using λ=0.5 and μ=0.5.

IV. CDNlink Design

A. CDNlink State of The Art

CDNlink is designed to run under MATLAB-Simulink which constructed as a block diagram environment for multi-domain simulation and model-based design [12, 14]. Simulink supports the system-level design, automatic code generation, and continuous test and verification of embedded systems, provides a graphical editor, customizable block libraries, and solvers for modeling and simulating dynamic systems.

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 the network system.



(a) (b) (c) (d)

Fig. 2. A comparison result of M/M/1 by SimEvents. (a) Theoretical result of system utilization; (b) Simulation result of system utilization; (c) Theoretical result of system-queue waiting time; (d) Simulation result of system-queue waiting time.

The Simulink design paradigm is significantly different from traditional design methodology. Rather than using complex structures and extensive codes, network engineers can use Simulink design to define models with advanced functional characteristics using continuous-time and discrete-time building blocks. Second, CDNlink 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 network systems, in particular queueing model that has been implemented as a basic rule of network simulation. Other mathematical models 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.

B. Client

In this section, we design a Client node of the CDNlink. Other nodes such as local DNS, DNS, Router, and Server are provided in the next section. The design of client node is given in Fig. 3. The configuration design consists of three units: packet generation, processing packet, and sink. A queuing model can be added to 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. Packet reply will come through a router and processed before entering client's sink.

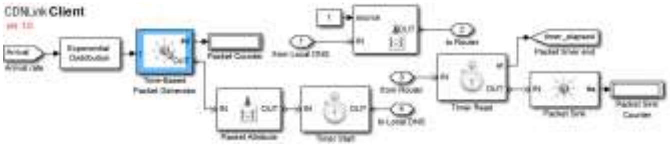


Fig. 3. The design of Client for CDNlink

C. Local DNS

The packet from a client forwards initially to a local DNS. The design of local DNS for CDNlink is depicted in Fig. 4(a). A Switch "Name Server Availability" module 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 sent 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.

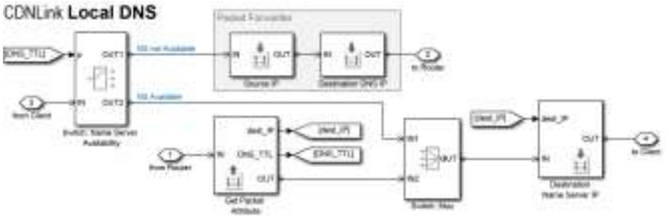


Fig. 4. The design of local DNS for CDNlink

D. DNS Nodes (Authoritative)

The design of DNS for CDNlink is depicted in Fig. 5. The packet from local DNS is forwarded to authoritative DNS server to obtain name server information according to the request. *Delay System* module is a system add a delay time when a packet is processed in the DNS system. In our design, DNS receives two type of packet: request packet from local DNS, and server state information (SSI) packet from servers. In this case, the authoritative DNS server is assumed having information of name servers as well as the state condition of those servers according to the information update by SSI packets. Therefore, the authoritative DNS is aware of the information and the condition for each server under its management, and choose an appropriate name server for each request. A mechanism for selecting the appropriate name server based on the request and state information is calculated in the "Packet Distributor" module, by implementing a specific request distribution algorithm. Load balancing algorithm is also possible to be included in this module. Next, after the appropriate name server is added, the "updated" request packet is forwarded to the source through a "Packet Forwarder" module.

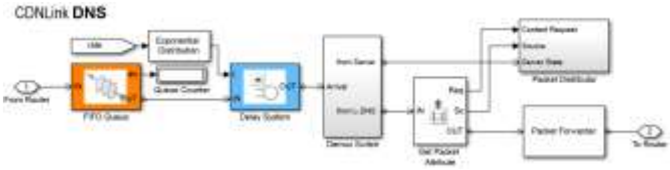


Fig. 5. The design of DNS for CDNlink.

E. Router

The configuration design of router for CDNlink 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 a packet is managed by the router. Next, after router evaluates 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 to the packet received by the router. The packet is possible to be forwarded to 4 destinations: client, local DNS (DNS), other routers, or servers.

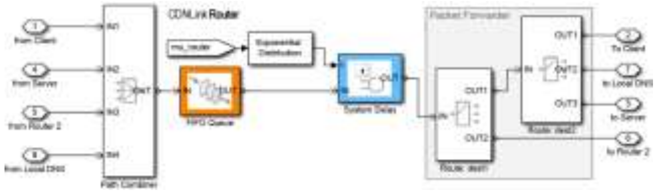


Fig. 6. The design of Router for CDNlink

F. Server

The configuration design of server for CDNlink is depicted in Fig. 7. Arrival packets are stored in "FIFO Queue" block before getting served. The "Processing Delay" block simulate the amount time of the server processing the requests and make replies. In addition, the server generates SSI packet to inform the authoritative DNS the state condition of a load of the server.

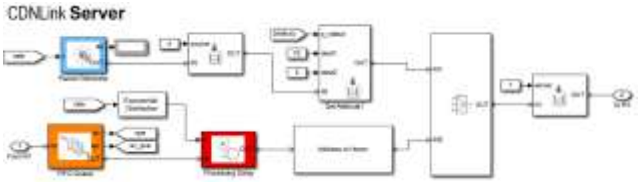


Fig. 6. The design of DNS for CDNlink.

V. CDNLINK IMPLEMENTATION

A. CDN Topology

A network topology is created to apply the CDNlink as depicted in Fig. 8. The CDNlink design on MATLAB-Simulink using SimEvents is depicted in Fig. 9. The network topology consists of 9 routers, 9 clients, 9 local DNS, 1 origin server, 1 Authoritative DNS, and 2 surrogate servers.

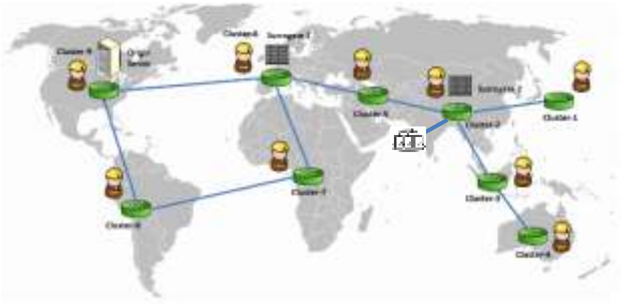


Fig. 7. The design of DNS for CDNlink.

We ran the CDNlink using MATLAB 2015a, Simulink 7.14, SimEvents 4.0, with a computer equipped with Windows 7, Core i7 processor, and 16GB of RAM. We provide performance result of average origin's queue length and a number of the packet, and average and elapsed packet RTT shown in figure 9.

B. Simulation Scenario

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 is redirected to the nearest surrogate server.

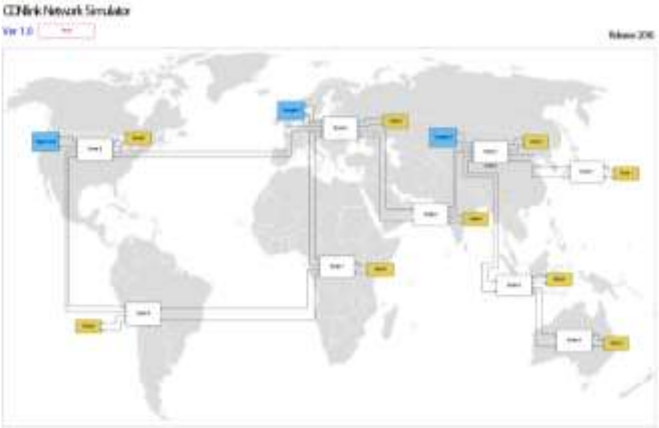
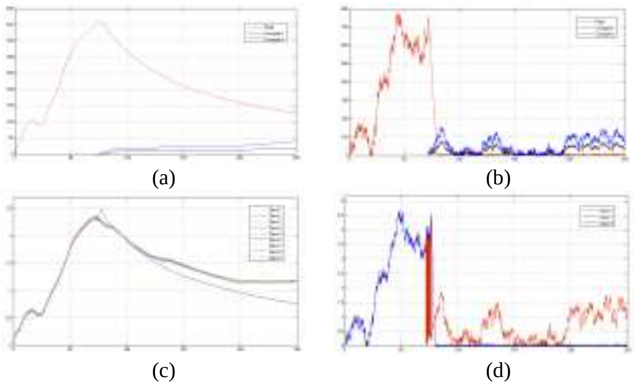


Fig. 9. The Design of CDNlink on MATLAB-Simulink using SimEvents.

C. Simulation Results

Performance result in Figs. 10.(a) and 10.(b) shows that all requests from all clients to the origin are redirected to the nearest surrogate server at time 75 seconds. The average queue length is decreased and at the same time, the queue-length of both surrogate 1 and 2 is increased.

According to the Fig. 10(c), RTT was increasing from starting of the simulation, and decrease at the time when the queue length at origin server has reached the threshold, because, all request were redirected to the nearest server. In Fig. 10(d) shows the elapsed RTT from client 1, 7 and 9. These clients represent another client that redirected to nearest server, where client 1, 2, 3, 4 were redirected to surrogate 2, client 5, 6, 7 redirected to surrogate 1, and client 8, 9 to the origin server. Performance result shows the RTT is stable after redirection process.



(a) (b) (c) (d)

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

VI. CONCLUSION

We introduced CDNlink, a model-based design network simulator as a full customized system and user-friendliness in both of model embedded and graphical interface. CDNlink is designed based on MATLAB Simulink with SimEvents Toolbox. We provide the design of Client, Router, and Server and demonstrate the implementation of the CDNlink for a CDN. In future work, we will implement a new method and design it in the CDNlink to compare and improve the quality of CDN with smart load balancing mechanism.

VII. ACKNOWLEDGEMENT

This work was partially supported by Research Grant from Mathematics Department of Universitas Islam Bandung and LPPM UNISBA (Extern-0812, 2015/2016). Special gratitude is addressed to Hiroaki Nishi Laboratory, Keio University, Japan.

REFERENCES

[1] Nurul I. Sarkar, Syafnidar A. Halim, "A review of simulation of telecommunication networks: simulators, classification, comparison, methodologies, and recommendations," Journal of Selected Areas in Telecommunications (JSAT), pp. 10-17, 2011.

[2] Jon Starkweather, "Simulation as an important method for learning and a necessary step of good research practice," 2012. [Online]. Available: http://it.unt.edu/benchmarks/issues/2012/04/rss-matters. [Accessed 2016].

[3] George Pallis, Athena Vakali., "Insight and Perspectives for Content Delivery Networks," Journal Communication of the ACM, vol. 49, pp. 101-106, 2006.

[4] Rajkumar Buyya, Mukaddim Pathan, Athena Vakali, Content Delivery Networks, vol. 9, Springer Publishing Company, Incorporated., 2008.

[5] MatLab, "Discrete Event Simulation Software - SimEvents - Simulink," [Accessed 2016], http://www.mathworks.com/products/simevents/.

[6] Sabato Manfredi, Francesco Oliviero, and Simon Pietro Romano., "A Distributed Control Law for Load Balancing in Content Delivery Networks," IEEE/ACM Trans, on Networking, vol. 21, no. 1, pp. 55-68, February 2013.

[7] Riverbed, "Riverbed Modeler," 2016. www.riverbed.com.

[8] CDNsim, "CDNsim, A Content Distribution Network simulator", http://oswinds.csd.auth.gr/CDNsim/, [Accessed 2016].

[9] PlanetLab, "PlanetLab, An open platform for developing, deploying, and accessing planetary-scale services," [2016]. www.planet-lab.org.

[10] Ng, Chee-Hock and Boon-Hee, Soong, "Queueing Modelling Fundamentals: With Applications in Communication Networks", 2 ed., England: John Wiley & Sons Ltd Publishing Ltd., 2008.

[11] Gunter Bolch, et.al., "Queuing Networks and Markov Chains, Modeling and Performance Evaluation with Computer Science Applications", vol. 2, Hoboken: John Willey & Sons Inc., 2006.

[12] Mathworks, "Simulink: Simulation and Model-Based Design," http://www.mathworks.com/products/simulink/, [Accessed 2016].

[13] Janaka L Wijekoon, Erwin Harahap, Rajitha Tennekoon, Hiroaki Nishi., "How can a service-oriented router merge with a CDN?", IEEJ Transactions on Electronics, Information and Systems, The Institute of Electrical Engineers of Japan, vol. 136, no. 8, pp. 1172-1179, August, 2016.

[14] Rajitha Tennekoon, Janaka Wijekoon, Erwin Harahap, and Hiroaki Nishi, "Prototype implementation of fast and secure traceability service over public networks", IEEJ Transactions on Electrical and Electronic Engineering, Vo. 11. No. S1., pp. 122-133., Wiley Subscription Services, Inc., A Wiley Company, June 2016.

[15] Erwin Harahap, et. al. Janaka Wijekoon, Rajitha Tennekoon, Fumito Yamaguchi, Shinichi Ishida, and Hiroaki Nishi., "Modeling of router-based request redirection for content distribution network", International Journal of Computer Applications, vol. 76., no. 13., pp.37-46., August 2013.