IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 17, Issue 1, Ver. IV (Jan – Feb. 2015), PP 06-12
www.iosrjournals.org

Performance Evaluation of a Distributed System Based Upon Fault Tree Analysis

Shipra Singh and M. L. Garg
Department of Computer Science, DIT University, Dehradun, India

Abstract: *Distributed Systems is the study of geographically distant processors, connected to one another through intermediate devices such as routers and/or switches. Simulation provides an insight into the behavior of these distant apart systems. The internal details like designing, configuring and maintaining a reliable distributed system through protocols that include routing, switching, etc is well understood by this paper. The paper describes how the design of a distributed system be built on a Network Designer while using leased line, serial ports and Ethernet technologies. Further, these designs have been simulated on a Network Simulator with the use of routing protocols. For this scheme to be implemented successfully different equipment like routers, switches, Ethernet and serial connection, PCs, etc are used and techniques like IP addressing schemes and RIP are implemented. Further, the developed system is decomposed to form a Fault tree and its reliability is evaluated with the help of Fault tree Analysis.*

Keywords: *Router, Switches, Ethernet, Serial, IP Address, Routing Information Protocol, Fault Tree Analysis.*

I. Introduction

Distributed computing is a method of computer processing in which different parts of a program are run simultaneously on two or more computers that are communicating with each other. Distributed computing is a kind of segmented or parallel computing, but the final term is mainly commonly used to refer to processing in which different parts of a program run simultaneously on two or more processors that are part of the same computer [1], [2]. While both types of processing require that a program be segmented—divided into sections that can run at the same time, distributed computing also requires that the division of the program take into account the different environments on which the different sections of the program will be executing. For example, two computers are probable to have different file systems and different hardware components.

Distributed systems provide cost effective means for resource sharing and extensibility, and obtain potential increases in performance, reliability, and fault tolerance [3], [4], [5], [6]. Such systems have gained popularity due to the low-cost hardware support in the recent years. A common Distributed System is made up of several nodes connected by a network where computing functions are shared among the nodes [7]. Further, these networks are mutually connected with one another with the help of routers and/or switches. Installing and configuring the router/switch is very cumbersome and highly technical task [8], [9], a very small mistake in the configuration process can lead to blunders in the network and can affect the reliability as well as the availability of distributed processors.

Efficient methods to optimize the system reliability have been reported by Raghavendra et. al. [10] which describes reliability of the distributed computing system as dependent not only on reliability of a communication network but also on the reliability of the processing nodes and distribution of the resources in the network whereas according to Shatz et al [11], when the system hardware configuration is fixed the system reliability mainly depends on the allocation of resources. [11] discussed the task allocation oriented reliability analysis of distributed systems with homogeneous hardware configuration. The authors considered the redundancy of both processors and communication links. A quantitative task assignment model is derived and is used to present and discuss models and algorithms for systems with level-2 or level-3 redundancy.

Simulation of a real distributed system helps in understanding the behaviors of it. It helps the learner to enhance analytical and design skills by providing safe and reliable environment for making practice [12]. The present piece of research is an attempt to design and establish a reliable Distributed System with multiple PC's, routers and switches, and to make them communicate with maximum reliability and minimum data or packet loss.

The rest of the paper is organized as follows. In section II, we discuss in brief the problem statement. Section III is the computational algorithm for how the system will be simulated to make the data transfer reliable. Section IV is a sample problem instance of a real distributed system. Section V is fault tree analysis of proposed problem and finally we conclude the research work in the Section V with the future prospects of the concept.

II. Problem Statement

Let the given system consists of a set of distant nodes interconnected with one another by the means of hardware devices termed as routers and switches. Individual processors can be connected to one another (using a switch) in any random manner and in any form of network topology, and these small-distributed networks can further be connected to a different network (using a router) and so on. Authors have to suggest an optimal algorithm to design and simulate the proposed distributed system to make the communication between each communicating node reliable.

III. Proposed Method

Once the distributed environment is established according to the choice of the user, its design and simulation part begins. The reliability of the system exists in safe assignment of IP addresses, synchronization of clock rate and definition of default gateways. Authors explain the complete design and simulation process with the help of a three – phase – protocol.

Phase 1: Configuration of routers.
Phase 2: Applying the routing information protocol
Phase 3: Assignment of IP address and default gateway for the distributed processors.

The computational algorithm for the design and simulation of a distributed system with 'n' processors connected with the help of 'm' routers is as follows:

Algorithm:
Step 1: Input i, j, m, n.
Step 2: Enable the network.
Step 3: Configure the connections.
Step 4: For i =1 to m do
Begin

4.1 Assign IP address for s0 port.
4.2 Synchronize the clock rate.
4.3 Assign IP address for e0 port.
4.4 Assign IP address for s1 port.
4.5 Synchronize the clock rate.
End

Step 5: Run the routing protocol between the networks (say RIP/ OSPF/ BGP etc.).
Step 6: For j =1 to n do
Begin

6.1 Assign IP address to the PCj.
6.2 Assign Subnet mask.
6.3 Define default gateway.
End

Step 7: ping a PC from any other PC in the network.
Step 8: If (Loss = 0%) OR (Packets sent = Packets received)
Print result and exit.
Else
Go to step 4.

IV. Discussion Of The Algorithm With Problem Instance

Consider the following distributed system with multiple processors connected with one another with the help of switches and routers. In the presented problem (Figure. 1), we have used 2501 series router that contains an Ethernet port and two serial ports and a 3550 series switch that contains twelve fast Ethernet ports and two Gigabit Ethernet ports. All the processors are connected with the help of these routers and switches to form this distributed environment. As per the algorithm, the design and simulation part is done with three – phase – protocol. Here is the detailed implementation of the algorithm.

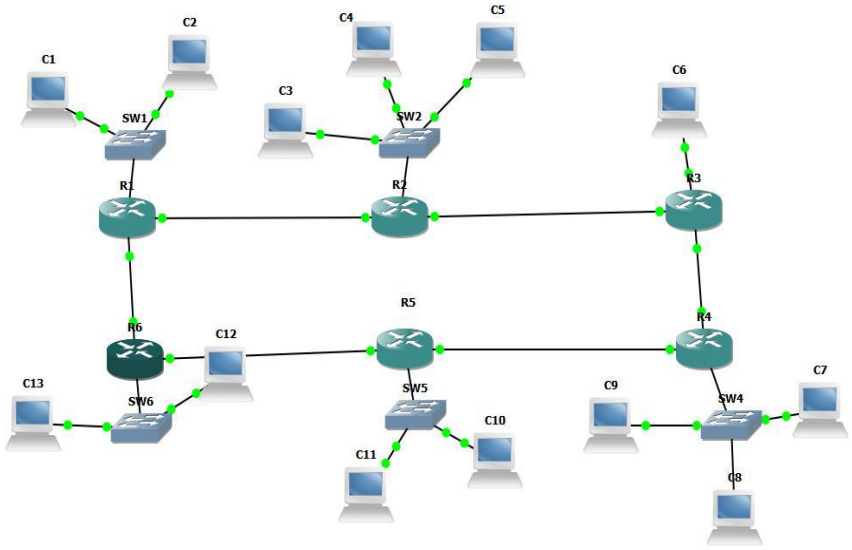

Fig.1: A sample distributed system

Phase 1: Configuring the routers

In the first phase, IP addresses are assigned to each of the router for its distinct serial and Ethernet ports Table 1.

Table 1: Router Configuration

Router Name	Port	IP Address	Subnet Mask
Router 1	e0	10.0.0.1	255.0.0.0
	s0	11.0.0.1	255.0.0.0
	s1	28.0.0.1	255.0.0.0
Router 2	e0	12.0.0.1	255.0.0.0
	s0	11.0.0.2	255.0.0.0
	s1	13.0.0.1	255.0.0.0
Router 3	s0	15.0.0.2	255.0.0.0
	s1	16.0.0.1	255.0.0.0
Router 4	e0	14.0.0.1	255.0.0.0
	s0	13.0.0.2	255.0.0.0
	s1	15.0.0.1	255.0.0.0
Router 5	e0	17.0.0.1	255.0.0.0
	s0	16.0.0.2	255.0.0.0
	s1	18.0.0.1	255.0.0.0
Router 6	e0	19.0.0.1	255.0.0.0
	s0	18.0.0.2	255.0.0.0
	s1	20.0.0.1	255.0.0.0

Phase 2: Followed by the phase 1 of the algorithm, the routing information protocol is fired in between the routers. As soon as the IP addresses have been assigned within two consecutive routers, the RIP is fired. The process repeats itself till all the routers have been configured.

Phase 3: Configuring the distributed processors

All the distributed processors are assigned IP Address, Subnet mask and default gateway as depicted in Table 2.

Table 2: Processor IP Address Assignment

Processor	IP Address	Subnet Mask	Default Gateway
PC 1	10.0.0.2	255.0.0.0	10.0.0.1
PC 2	10.0.0.3	255.0.0.0	10.0.0.1
PC 3	10.0.0.4	255.0.0.0	10.0.0.1
PC 4	10.0.0.5	255.0.0.0	10.0.0.1
PC 5	12.0.0.2	255.0.0.0	12.0.0.1
PC 6	14.0.0.2	255.0.0.0	14.0.0.1
PC 7	14.0.0.3	255.0.0.0	14.0.0.1
PC 8	14.0.0.4	255.0.0.0	14.0.0.1
PC 9	17.0.0.2	255.0.0.0	17.0.0.1
PC 10	19.0.0.2	255.0.0.0	19.0.0.1
PC 11	19.0.0.3	255.0.0.0	19.0.0.1
PC 12	19.0.0.4	255.0.0.0	19.0.0.1
PC 13	21.0.0.2	255.0.0.0	21.0.0.1

After successful completion of the three phases, the output that we have is a reliable distributed system. To verify the deploy ability of the algorithm, one can ping any processor from any other one in the distributed system. To check the reliability of the designed DS, processor PC11 has been pinged from PC4 and the communication result is depicted in Figure. 2.

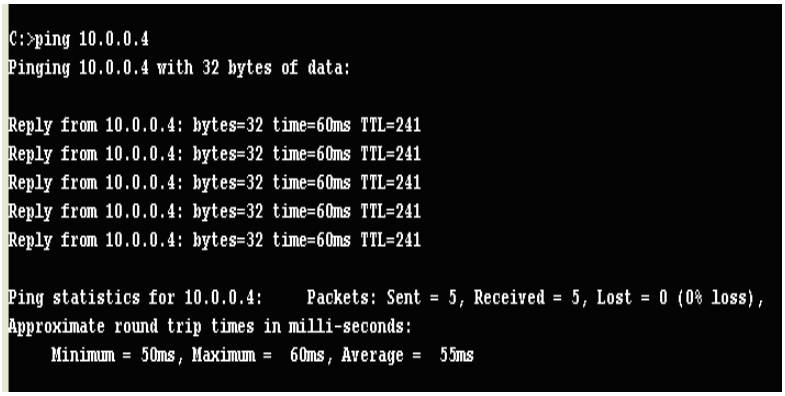

Fig. 2 Reply from PC11 to PC4

V. Reliability Analysis Of The System Using Fault Tree

Risk can be analyzed in one of two basic ways: inductively or deductively, that is either bottom-up or top-down. In a deductive analysis a system failure is postulated. The analyst then works backwards to deduce what combinations of events could have occurred for the failure to have taken place (a detective solving a crime is thinking deductively). Fault tree analysis is deductive. An inductive analysis works in the other direction. A single failure, such as a pump stopping or a valve closing at the wrong time, is postulated. The inductive analysis then determines what impact the item failure could have on the overall system performance. Event tree analysis is inductive [13].

When a system is formed from elements and units connected in parallel, series or mixed configurations, a suitable method of calculating its reliability becomes necessary [14]. Fault tree analysis is unique ways to identify how a system may fail to function a specified task and how will it reflect to the reliability of a system.

The fault tree can easily be constructed of the above-distributed system as all the nodes are connected in series. (Figure 3).

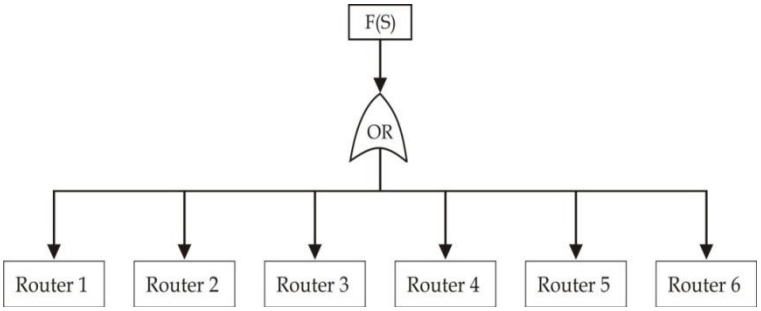

Fig. 3 Fault Tree of Distributed Systems

To understand failure rate of the routers, different nodes were 'pinged' from one another and their packet loss have been noted. After repeating the same process multiple times with different packet numbers, we get a tabular format of packet loss (Table 3).

Table 3: Reply from nth Processor to $(n+i)^{th}$ Processor

Source Node	Destination Node	No. of packets sent	% of Packet Loss
PC9	PC5	59	0
		403	1
		826	0
	PC12	83	1
		356	0
		614	2
	PC1	148	0
		376	1
		1003	2

PC2	PC5	57	0
		245	1
		509	0
	PC1	50	1
		311	2
		490	0
	PC9	165	1
		478	0
		895	2
PC3	PC11	36	1
		90	0
		410	1
	PC8	705	2
		357	3
		110	2
	PC6	1100	4
		121	2
		429	0

Figure 3(a) - Figure 3(c) represent the packet loss percentage between different processors of the network.

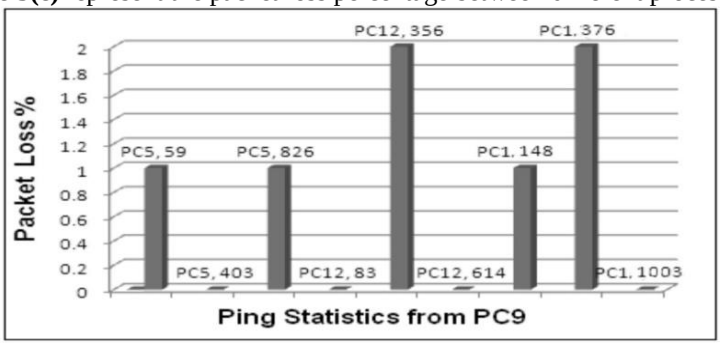

Fig 3(a) Run-time Packet Loss percentage of PC9 with PC5, PC12 and PC1

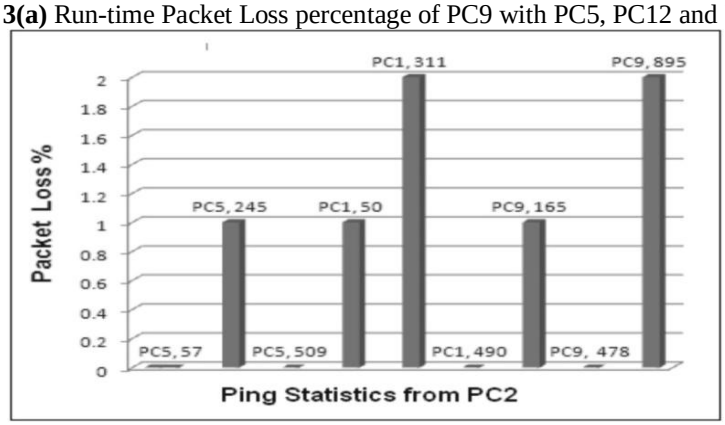

Fig 3(b) Run-time Packet Loss percentage of PC2 with PC5, PC1 and PC9

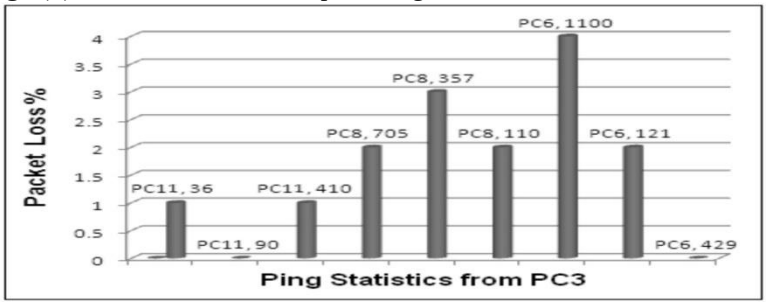

Fig 3(a) Run-time Packet Loss percentage of PC3 with PC11, PC8 and PC6

During this communication run on Boson NetSim [12], a total of 10,426 packets have been transferred from one processor to another to evaluate the reliability of designed distributed system. Simulation results show that during the entire process of communication, a total of 112 data packets have lost out of 10,426 packets that

comes out to be packet loss of 1.07%. This packet loss indicates probability of failure in the designed DS and hence affects the reliability of the system. This packet loss may be due to several reasons viz. network congestion, link failure etc. Since this packet loss cannot be deterministic on every router and may vary in different runs due reasons stated above, hence, we have to obtain the statistical probability of failure of each router. Since the packet loss of the designed system comes out to be 1.07%, we may use random events method [15] [16] [17] to generate probability of failure of each router. Accordingly, Table 4 depicts failure rate of routers.

Table 4: Failure Rate of Routers

Router No.	Probability of Failure
F(P1)	0.003
F(P2)	0.002
F(P3)	0.001
F(P4)	0.005
F(P5)	0.001
F(P6)	0.007

Now, we may obtain failure of the designed system F(S) as follows:
F(S) =P [(F1) OR (F2) OR F(3) OR F(4) OR F(5) OR F(6)]
= P[(F1 OR F2)] OR P[F(3) OR F(4) OR F(5) OR F(6)]
= [P(F1) + P(F2) – P(F1F2)] OR P[F(3) OR F(4) OR F(5) OR F(6)]
= [P(F1) + P(F2) – P(F1)P(F2)] OR P[F(3) OR F(4) OR F(5) OR F(6)]
Continuing in this way, we get
F(S) = [P(F1) + P(F2) – P(F1)(PF2)] OR [P(F3) + P(F4) – P(F3)P(F4)] OR [P(F5) + P(F6) - P(F5)(PF6)]]
= [0.003 + 0.002 – (0.003 * 0.002)] OR [0.001 + 0.005 – (0.001 * 0.005)] OR [0.001 + 0.007 – (0.001 * 0.007)]]
Continuing in this way, we get
F(S) = (0.019)
Reliability of the distributed System R(S) can be evaluated as:

$$R(S) = 1 – F(S) = 1 – 0.019$$

$$R(S) = 0.981$$

Hence, the reliability of the configured Distributed System comes out to be 0.981.

VI. Conclusion

A distributed computing environment has become necessary for all sort of distant communications, but from experience, we learned that it is not so easy to create, setup and manage even a small area network. It requires a lot of efforts to keep a distributed system up and running in reliable manner. Design and implement an actual distributed system is a complicated task Simulations provide us with the capability to understand and get a good feel about the things without actual risk involved in the given situation. Network simulation is a technique where a program models the behavior of a network either by calculating the interaction between the different network entities (hosts/packets, etc.) using mathematical formulas, or actually capturing and playing back observations from a production network. The behavior of the network and the various applications and services it supports can then be observed in a test lab; various attributes of the environment can also be modified in a controlled manner to assess how the network would behave under different conditions.

In this paper, an attempt has been made to discuss all the details regarding how to establish a connection amongst distributed processors and further to evaluated the reliability of the system, how to build a block diagram of Fault Tree and how the reliability of the system is evaluated through FTA. The satisfactory part of our research is that designed system's reliability comes out to be 0.981 i.e. the complex distributed system we designed for our simulation is 98.1% reliable.

References

[1]. Casavant T. L. & M. Singhal, "Readings in Distributed Computing Systems," Llos Alamitos, CA: IEEE Computer Society Press. 1994.
[2]. Loy D. , Dietrich D. & H. J. Schweinzer , "Open Control Networks, Boston," MA: Kluwer Academic Publishers, 2001.
[3]. N. Lopez-Benitez, Dependability Modeling and Analysis of Distributed Programs, IEEE Trans. Software Engineering, Vol. 20 No 5, pp. 345- 352, May 1994.
[4]. A. Kumar, S. Rai and D. P. Agrawal, On Computer Communication Network Reliability Under Program Execution Constraints, IEEE Trans. On selected areas in Communications, Vol. 6, No 8, pp. 1393-1400, October 1988.
[5]. V. K. P. Kumar, S. Hariri, and C. S. Raghavendra, Distributed Program Reliability Analysis, IEEE Trans. On Software Engineering, Vol. SE-12, No 1, pp. 42-50, Jan 1986.
[6]. M. S Lin and D. J. Chen, General Reduction Methods for the Reliability Analysis of Distributed Computing Systems, The Computer Journal, Vol. 36, No 7, 1993.
[7]. Min X., Yuan-shun Dai, & Kim-leng Poh, "Computer System Reliability, Models and Analysis," Kluwer Academic Publishers, New York, 2004.

[8]. M. F. Nawaz, F. Hadi, S. U. Shah, "RouterSim: A New Router Simulator for BGP and IS-IS Protocol", International Conference on Future Computer and Communication, by ieee computer society, pp. 107 – 111, 2009.
[9]. M. A. Qadeer, P. Varshney, N. H. Khan, "Design and Simulation of Interconnected Autonomous Systems", International Conference on Computer Engineering and Technology, conducted by ieee computer society, pp. 270- 275, 2009.
[10]. Raghavendra, C. S., and Hariri, S., "Reliability Optimization in the Design of Distributed Systems", IEEE Transactions on Software Engineering, Vol. SE-11, pp. 1184-1193, 1985.
[11]. Shatz, S. M., and Wang, Jai-Ping, "Models and Algorithms for Reliability-Oriented Task, Allocation in Redundant Distributed Computer Systems", IEEE Transactions on Reliability, Vol. 38, No. 1, pp. 16-27, 1989.
[12]. Boson Netsim 6.0: http://www.boson.com/Product/CIS-NS-CCNP- 02.html
[13]. Dan Goldin, NASA Administrator, Fault Tree Analysis, Clifton A. Ericson II, 2000.
[14]. Sutton, Ian S., Fault Tree Analysis, Sutton Technical Books, Houston, Texas, 2011.
[15]. L. S. Srinath, Reliability Engineering, Fourth Edition, East west press, 2006.
[16]. P. Jalote, An integrated approach to Software Engineering, Narosa Publishing House, Second Edition, 2003.
[17]. Aggarwal, K., K., and Yogesh Singh, Software Engineering, New Age International Publishers, Revised Second Edition, 2005.