

Latency Modeling for Testing & Evaluation

Eric D. Smith, Manish Khadtare, David Delgado
Systems Engineering Program
Research Institute for Manufacturing & Engineering Systems
University of Texas at El Paso

Abstract

Latency testing and evaluation of new and legacy systems is crucial. Although legacy systems, government furnished equipment (GFE), and commercial off the shelf (COTS) systems may be reliable as standalone systems, their integration into modern system may be problematic because of unanticipated latency issues.

This paper reviews general latency modeling and simulation, including a review of system functionality determination and mapping to a model. The case study shows a modeling and simulation example with a readily available Excel Add-In, and presents the results.

Introduction

Latency in Ethernet communications can be described generally in the following way. Transference time is in general defined as the sum of the processing time of communications stack of the message sender device, the transmission time in the network, and the processing time of the communications stack in the receiving device. See Figure 1.

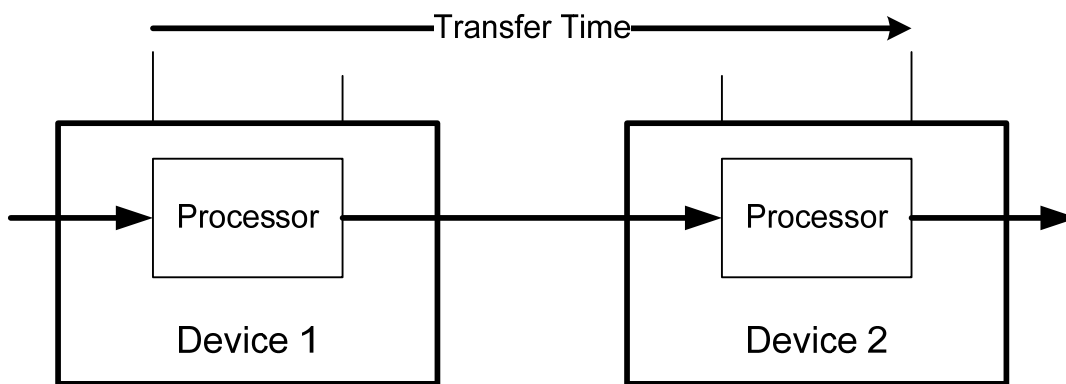


Figure 1: Transfer time

The total Transfer Time is therefore the addition of the time taken by the frame to get processed between two devices.

Usually, latency times are in the tens or hundreds of milliseconds. Sometimes, messages are differentiated according to priority levels. There are two major reasons that affect the latency of an Ethernet frame: The first is the

physical propagation of the signals through a transmission medium, and the second is the time needed to pass through switches and other electronic equipment. Physical propagation of the signals is carried out through an electrical medium or an optical medium. Propagation time is proportional to distance and inversely proportional to propagation speed. The common CAT-6 type cable has a propagation lag of 5 nanoseconds per meter.

Present switches work on a “store & forward” mode, that is, they receive the whole frame, they analyze it, they direct it to the corresponding port and they put it in the outgoing cable. The accumulated lag depends on the features of the switch, the speed of the involved ports, and the amount of traffic at the switch at that moment. Typical delays for a frame passing through a switch vary from a few microseconds to hundreds of milliseconds.

For a frame propagating through physical media, we can calculate the minimum time of the frame will spend in the cable. A 64-byte frame takes the following times to pass a reference point, depending on different transmission rates given in nano-seconds per bit:

- 10Mbps: $672 \text{ bits} \times 100\text{ns/bit} = 67.2\mu\text{s}$.
- 100Mbps: $672 \text{ bits} \times 10\text{ns/bit} = 6.72\mu\text{s}$.
- 1Gbps: $672 \text{ bits} \times 1\text{ns/bit} = 672\text{ns}$.

The latency introduced by the switch must be added to the time and for that purpose it is necessary to explain the exact definition of switch latency. We can define latency in a switch as the “time interval starting when the last bit of the input frame reaches the input port and ending when the first bit of the output frame is seen on the output port.” By defining latency in this way, the value of the latency is independent of the length of the frame. To get an idea of the complexity of a current Ethernet switch, the different stages that compound the switch should be analyzed. These devices internal to a switch can be subdivided into three blocks:

- Input block
- Switching block
- Output block

Latency analysis for switches can be carried out through different tests. For example, the behavior of the latency of accessing traffic within the switch can be noted. A different approach is to note the latency for traffic going from a substation house to a central substation, as in Figure 2.

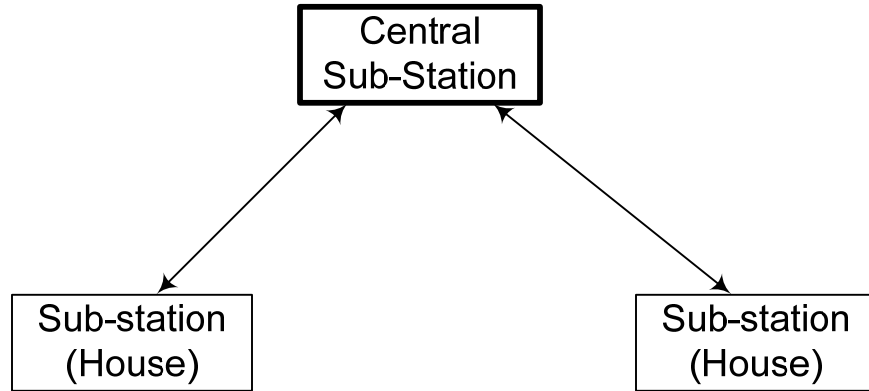


Figure 2: Sub-station network

Latency time can be used as an input parameter to determine quality or other system characteristics. Feedback and feed-forward devices can utilize latency times to respond to changing circumstances, for example, in aero-vehicle location and coordination. Traffic generated by non-latency-critical applications may have time intensive protocols for lost frame recovery.

Analytic latency modeling, where mathematical formulas model the exact intricacies of latencies, can be quite challenging and complicated [Soh and Dillon, 1996] [Laki, Matray, Haga, Csabai and Vattay] [Cho, Choi and Cho] [Wang, Krishnamurthy, Martin, Anderson and Culler] [Sikdar, Kalyanaraman and Vastola] [Cousins, 2006]. However, Analytic modeling difficulties can be largely bypassed by resorting to probabilistic Monte Carlo analysis latency-critical situations.

Case Study: Electronic Communications Model

- Probabilistic latency modeling can easily be accomplished by the following steps:
- 1, Acquiring latency specifications for the components involved,
 - 2, Formulating a probabilistic model, and,
 - 3, Running the model, and analyzing the output.

Figure 3 shows a typical logical functional flow in electronic communications.

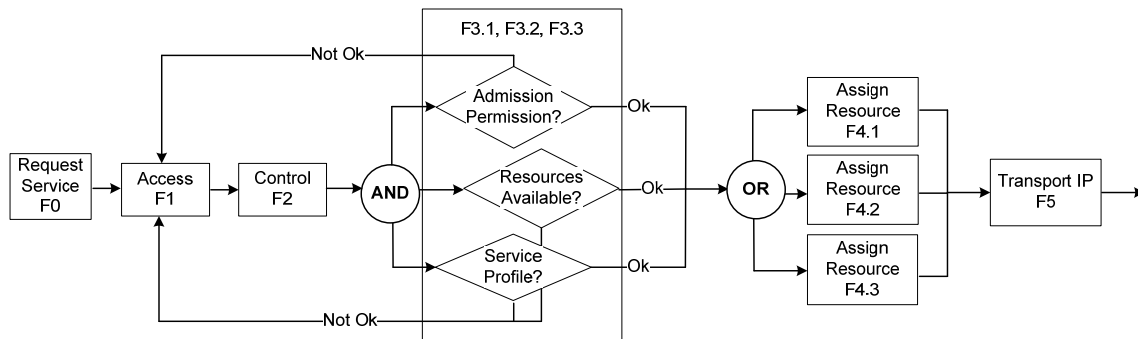


Figure 3: Communication logical functional flow

The functions and operations are labeled as FO, F1, etc., and proceed logically from left to right. Logically, the functions must occur serially, except in the case where an AND proceeds F3.1, F3.2, and F3.3; in this case, all of the F3 sub-functions must be completed in order for the F4 functional group to begin working. The F4 functional group is composed of three (3) functional options, where the assignment of any one of them allows the functional flow to proceed to F5. An additional important feature in the diagram is that the F3 sub-functions must all be completed satisfactorily, or else the functional flow reverts back to F1. The F3 sub-functions therefore each have two (2) important parameters associated with them: latency of performance, and a check of satisfactory completion.

Modeling of this probabilistic scenario is simplified by the logical functions available within Excel. The following formula was implemented in Excel.

$$\begin{aligned} & \text{LatencyF0} + \text{LatencyF1} + \text{LatencyF2} \\ & + \text{IF}(\text{F3.1Ok and F3.2Ok and F3.3Ok})\{\text{MaxLatency}(\text{F3.1, F3.2, F3.3})\} \\ & \quad \text{ELSE}(\text{Return to F1}) \\ & + \text{MinLatency}(\text{F4.1, F4.2, F4.3}) + \text{LatencyF5} \end{aligned}$$

Implementing such a formula is made easy in Excel, because the individual latencies can be programmed into, and be made to appear, within a single cell in a spreadsheet. Obtaining the full latency formulation is then a matter of implementing the logic in a few additional cells, and adding the total latency in an output cell.

The probability distributions chosen for the functions F0-F5 were:

F0: Poisson distribution

The Poisson distribution gives the probability that a different number of exponentially distributed events will occur in a fixed period of time, and is often used in modeling service arrival times.

F1: Truncated Normal distribution

This distribution is a Normal distribution that can be arbitrarily truncated both with a minimum value and a maximum value, and can be useful in modeling naturalistic situations where there is a definite minimum time necessary for accomplishing a task, and a maximum time.

F2: Integer distribution

This distribution produces an integer value between a bottom and top integer value, and can model functions that only terminate on integer values.

F3: Exponential distribution

Exponential distributions are useful for modeling events that occur continuously and independently at a constant average rate; exponential distributions are memory-less, in that the time necessary for an event to

occur is independent of how much time has already passed in waiting for the event.

F4: Triangular distribution

Triangular distributions are useful approximations to unknown distributions. Triangular distributions are easily constructed if there exist reasonable estimates for a minimum time, most likely time, and maximum time.

F5: Discrete distribution

Discrete distributions are useful for modeling situation where a clock or other timing mechanism that is internal to a sub-system will only allow the sub-system to produce an output a known and discrete times. Each discrete time is accompanied by a probability that indicates how often it occurs.

RiskSim Add-In for Excel

For this model, RiskSim, a risk simulation Add-In for Excel, was employed.

RiskSim is available for trial from www.TreePlan.com.

The benefits of RiskSim are listed as:

- “* Save time by using RiskSim to automate Monte Carlo simulation.
- * Make informed decisions explicitly accounting for uncertainty.
- * Use RiskSim's charts to explain your analysis to colleagues.
- * Use Excel's formatting commands to customize RiskSim's charts.
- * Easily use the single-file RiskSim add-in with no complicated installation; simply open RiskSim.xla whenever and wherever you need it.”

To use RiskSim, you

- (1) create a spreadsheet model
 - (2) optionally use SensIt to identify critical inputs
 - (3) enter one of RiskSim's fourteen random number generator functions in each input cell of your model
 - (4) choose Tools | Risk Simulation from Excel's menu
 - (5) specify the model output cell and the number of what-if trials (maximum 32,000)
 - (6) interpret RiskSim's histogram and cumulative distribution charts.”
- (www.TreePlan.com)

RiskSim random number generator functions include the following:

Integer, Discrete, Cumulative, Triangular, Uniform, Exponential, Poisson, Binormal, Normal, Truncated Normal, Bi-Variate Normal, Truncated Bi-Variate Normal, Log Normal, Cumulative, and Sample.

Case Study Formulation

The simulation model in this paper was constructed in Excel with the support of RiskSim, and the spreadsheet that was formulated is shown in Table 1.

F0	F1	F2	F3	F4	F5		
Poisson	TrucNorm	Integer	Exp	Triangle	Discrete		
RANDPOISSON(mean)							
2468							
Mean	RANDTRUNCNORMAL(Mean,StDev,MinValue,MaxValue)						
2500	5534						
	Mean	RANDINTEGER(bottom,top)					
	5800	3166					
	StDev	Top	RANDEXPONENTIAL(lambda)				
	2100	3500	125				
	MinValue	Bottom	Lamda	RANDTRIANGULAR(minimum,most_likely,maximum)			
	1500	2750	0.002	2024			
	MaxValue		OK?	Minimum	RANDDISCRETE(value_discrete_table)		
	7800		RAND()	1780	4000		
			0.51	Most Likely	Latency	Probability	
			Threshold	1990	3000	0.3	
			0.01	Maximum	4000	0.6	
		Ok? IF	1	2840	5000	0.1	
			RANDEXPONENTIAL				
			104				
			Lamda	RANDTRIANGULAR(minimum,most_likely,maximum)			
			0.003	2056			
			OK?	Minimum			
			RAND()	1780			
			0.05	Most Likely			
			Threshold	1990			
			0.01	Maximum			
		Ok? IF	1	2840			
			RANDEXPONENTIAL				
			456				
			Lamda	RANDTRIANGULAR(minimum,most_likely,maximum)			
			0.004	2238			
			OK?	Minimum			
			RAND()	1780			
			0.61	Most Likely			
			Threshold	1990			
			0.01	Maximum			
		Ok? IF	1	2840			
LatencyF0	LatencyF1	LatencyF2	LatencyF3	LatencyF4	LatencyF5	LATENCY	
2468	5534	3166	456	2024	4000	17648	
			IF All Ok	MINIMUM			
			456				
			MAXIMUM				
			9156				
			Sum(F1+F2+MaxF3)				

Table 1: Latency simulation model as constructed in Excel with RiskSim

Simplifying assumptions are often necessary in constructing latency models. In this case study, it was necessary to simplify the possible endless recursion between functional set F3 and function F1 with the assumption that, if any failure occurred in functional set F3, there would be only one recursion to F1. In the spreadsheet, an additional assumption was made, and this was that the latency times for the recursive call to F1, F2 and F3 could be taken to be the same latency times as occurred before the failure in functional set F3. This is a reasonable assumption, since the latency times themselves are original within the simulation instance.

Common errors that occur during construction of simulation models include: Incorrect specifications, improper assumptions for the probability distributions, improper simplifications of the real architecture, and incorrect formulation of the intended model within Excel. Additionally, the output of the simulation must be interpreted correctly in order to obtain useful results. In critical applications, it may not be reasonable to assume that the underlying simulation package is without errors.

Case Study Results

RiskSim's output distribution for the LATENCY, after 10,000 iterations, is a histogram of output values, as shown in Figure 4.

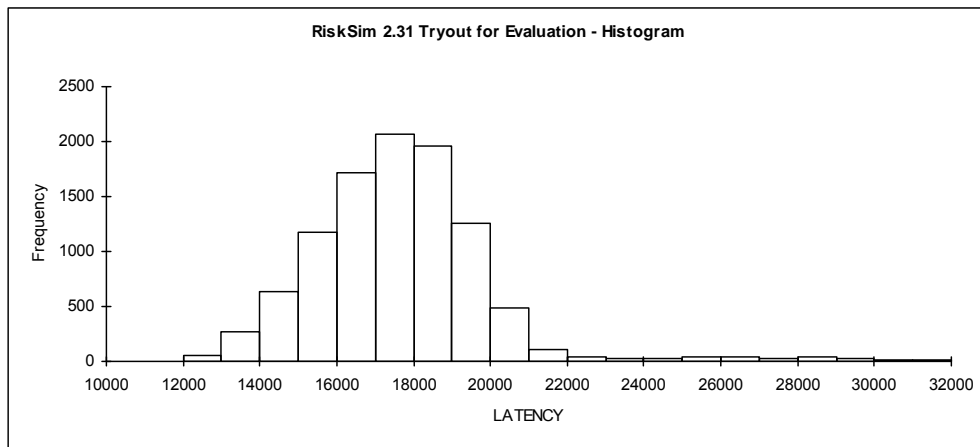


Figure 4: Histogram of output values

RiskSim also produces numerical summary output, which describes the histogram, as shown in Table 2.

RiskSim 2.31		Mean	17664
Date	9-Nov-09	St. Dev.	2351
Time	8:17 PM	Mean St. Error	24
Workbook	risk231e.xls	Minimum	11609
Worksheet	ITEA	First Quartile	16230
Output Cell	\$H\$47	Median	17576
Output			
Label	LATENCY	Third Quartile	18791
Seed	3308851	Maximum	31714
Trials	10000	Skewness	1.6746

Table 2: Example RiskSim tabular, numerical summary output

From this information, it is easy to determine the output mean, and to predict the deviation from the output mean to whatever confidence level is needed. The outputted histogram is also used to construct a cumulative chart, as shown in Figure 5.

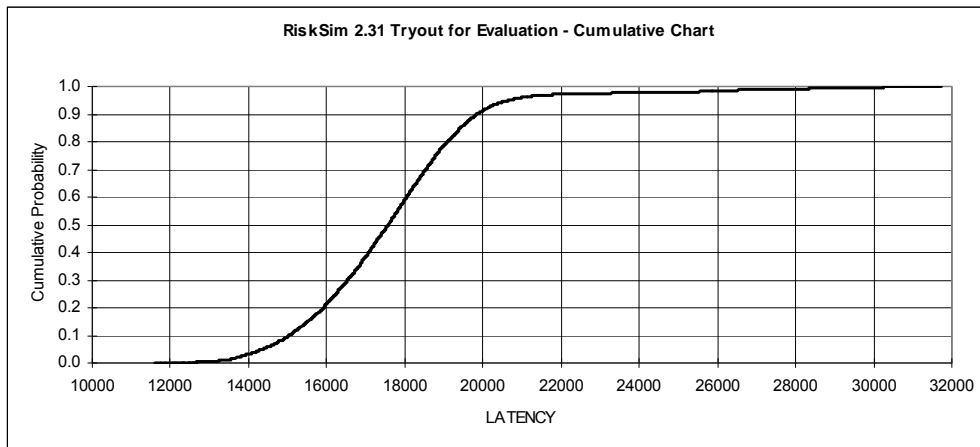


Figure 5: Cumulative chart of output histogram

Conclusion

Latency modeling can be vastly simplified by substituting exact analytic formulations with probabilistic models that can be easily formulated and run in Excel with the aid of a simulation add-in such as RiskSim. The results of a probabilistic simulation are robust and easy to examine.

References

- Y.-S. Cho, E.-J. Choi, and K.-R. Cho, "Modeling and analysis of the system bus latency on the SoC platform," Korea.
- M. Cousins, Dealing with latency, MusicTech Magazine(May) (2006), 43-46.
- S. Laki, P. Matray, P. Haga, I. Csabai, and G. Vattay, "A detailed path-latency model for router geolocation," Budapest, Hungary.
- B. Sikdar, S. Kalyanaraman, and K. S. Vastola, "An integrated model for the latency and steady-state throughput of TCP connections," Rensselaer Polytechnic Institute, Troy, NY.
- B. C. Soh, and T. S. Dillon, Hardware fault latency: Model validation, Microelectronics Reliability 36(9) (1996), 1231-1235.
- R. Y. Wang, A. Krishnamurthy, R. P. Martin, T. E. Anderson, and D. E. Culler, "Modeling communication pipeline latency."