Availability Analysis of Computer Network using Petri-Nets

노철우,
신라대학교 컴퓨터정보공학부

Cheul Woo Ro, Artem Pak
Division of Computer and Information Engineering
Silla University, Busan, KOREA

Abstract

This paper reviews methods used to perform reliability and availability analysis of the network system composed by nodes and links. The combination of nodes and links forms virtual connections (VC). The failure of several VCs cause failure of whole network system. Petri Net models are used to analyze the reliability and availability. Stochastic reward nets (SRN) is an extension of stochastic Petri nets provides modelling facilities for network system analysis.

1. INTRODUCTION

Users of network systems want them to be reliable which refers to the overall quality of the data communication system. Although bug-free operation is the goal of any client-oriented system, it cannot be achieved in practice. Once again, cost, time to market, and the impossibility of testing all possible scenarios preclude it. The question, therefore, is how many bugs are acceptable. In telephony systems, a typical goal is to mishandle no more than one call in 10,000, which equates to four-nines reliability, or 99.99%. In other systems, such as those that handle financial transactions, the requirements are even more stringent. If a telephone call is mishandled, the user can simply try again. However, if a financial transaction is mishandled, correcting the problem is much more tedious and expensive, because someone will probably have to correct the error manually.

Hardware failure, data corruption, and physical site destruction all pose threats to a network system that must be available close to 100 percent of the time. We can enhance the availability of a network by identifying network components that must be available, then identifying the points at which those components can fail. Increasing availability also means reducing the probability of failure. Network availability directly depends on the hardware and software used in building the network system, and the effectiveness of the operating procedures.

The *reliability* of a system is its ability to maintain operation over a period of time t. Formally, the reliability, $R(t)$, of a system is

$$R(t) = P_r(\textit{the system is operational} \in [0, t]).$$

If we define X to be a random variable representing the lifetime of the system and also letting F be the cumulative distribution function (CDF) of X, then the reliability of the system at time t is

$$R(t) = P_r(X > t) = 1 - F(t).$$

It is assumed that a system is working properly at $t = 0$; therefore, $R(0) = 1$.

When modeling a system, it is often but not always assumed that the failure rate is constant; however, this assumption only holds for the normal lifetime of a system and is not true during *burn-in* or *end-of-life*. The importance of this assumption is when the failure rate, λ, is constant, the resulting CDF of the lifetime of the components is exponential. That is

$$F(t) = 1 - e^{-\lambda t}$$

and the reliability is

$$R(t) = e^{-\lambda t}$$

Another measure often used for the analysis of systems is *availability*. The availability of a system is often expressed as the *instantaneous availability*, $A(t)$, and/or the *steady-state availability*. The instantaneous availability, $A(t)$, is defined as the probability that a system is operational at time t. It allows for one or more failures to have occurred during the interval (0; t). If a system is not repairable (e.g., a deep space exploring spacecraft), the definition of $A(t)$ is equivalent to $R(t)$. *Dependability* is used as a catch-all phrase for various measures such as reliability, availability etc.

Another measure used to describe a system is its expected life or *mean time to failure* (MTTF). Formally,

$$MTTF = \int_0^\infty R(t)\,dt.$$

If we continue our assumption of a constant failure rate, λ, then the MTTF of a system is simply $1/\lambda$.

The purpose of this paper is to illustrate methods used for determining the reliability and availability of a network system consisting of nodes and links. Using the nodes and links we form several virtual connections (VC) from source to destination. We consider two cases: failure of network components without repair and with independent repair. We will illustrate SRN modelling techniques for analysis of the network system using VCs.

The paper organized as follows: Section 2 deals with introduction and extension of Petri Nets (PN), Section 3 describes network system consisted of nodes and links, Section 4 presents the system modelling using PN, Section 5 gives the numeric results.

2. STOCHASTIC PETRI NETS

2.1 Stochastic Petri Nets (SPNs)

A Petri Net (PN) is a bipartite directed graph with two disjoint sets called places and transitions [2,3]. Directed arcs in the graph connect places to transitions (called input arcs) and transitions to places (called output arcs). Places may contain an integer number of entities called tokens. The state or condition of the system is associated with the presence or absence of tokens in various places in the net. The condition of the net may enable some transitions to fire. This firing of a transition is the removal of tokens from one or more places in the net and/or the arrival of tokens in one or more places in the net. The tokens are removed from places connected to the transition by an input arc; the tokens arrive in places connected to the transition by an output arc. A marked PN is obtained by associating tokens with places. The marking of a PN is the distribution of tokens in the places of the PN. A marking is represented by a vector

$M = (\#(P_1), \#(P_2), \#(P_3), ..., \#(P_n))$

where $\#(P_i)$ is the number of tokens in place i and n is the number of places in the net.

If exponentially distributed firing times correspond with the transitions, the result is a SPN [1,2,3]. Allowing transitions to have either zero firing times (immediate transitions) or exponentially distributed firing times (timed transitions) gives rise to the GSPN [3]. The transitions with exponentially distributed firing time are drawn as unfilled rectangles; immediate transitions are drawn as filled rectangles.

For a given GSPN, an extended reachability graph (ERG) is generated with the markings of the reachability set as the nodes and some stochastic information attached to the arcs, thus connecting the markings to each other. Under the condition that only a finite number of transitions can fire in finite time with non-zero probability, it can be shown that a given ERG can be reduced to a homogeneous continuous time Markov chain (CTMC) [7,9].

2.3 Stochastic Reward Nets (SRNs)

In order to make more compact models of complex systems, several extensions are made to GSPN, leading to the SRN. One of the most important features of SRN is its ability to allow extensive marking dependency. In an SRN, each tangible marking can be assigned with one or more reward rate(s). Parameters such as the firing rate of the timed transitions, the multiplicities of input/output arcs and the reward rate in a marking can be specified as functions of the number of tokens in any place in the SRN.

Another important characteristic of SRN is the ability to express complex enabling/disabling conditions through guard functions. This can greatly simplify the graphical representations of complex systems. For an SRN, all the output measures are expressed in terms of the expected values of the reward rate functions. To get the performance and reliability/availability measures of a system, appropriate reward rates are assigned to its SRN. As SRN is automatically transformed into a Markov Reward Model (MRM) [10,11], steady state and/or transient analysis of the MRM produces the required measures of the original SRN.

2.4 Markov Reward Models (MRM)

SRNs provide the same modeling capability as Markov reward models (MRMs). A Markov reward model is a Markov chain with reward rates (real numbers) assigned to each state. A state of an SRN is actually a marking (labeled $(\#(P_1), \#(P_2), \#(P_3), ..., \#(P_n))$ if there are n places in the net). We label the set of all possible markings that can be reached in the net as Ω. These markings are subdivided into tangible markings Ω_T and vanishing markings Ω_V. For each tangible marking i in Ω_T a reward rate r_i is assigned. This reward is determined by examining the overall measures to be obtained. Several measures are obtained using Markov reward models. These include the expected reward rate both in steady state and at a given time, the expected accumulated reward until either absorption or a given time, and the distribution of accumulated reward either until absorption or a given time.

The expected reward rate in steady state can be computed using the steady state probability of being in each marking i for all $i \in \Omega_T$. For steady state distribution π_i, the expected reward rate is given by

$$E[R] = \sum_{i \in \Omega_T} r_i \pi_i$$

The SRN model can be solved by using SPNP, a software package for the automated generation and solution of Markovian stochastic systems developed by researchers in Duke University [12].

3. NETWORK SYSTEM DESCRIPTION

Virtual Private Network (VPN) technology, which is widely used by many organizations, is based on the idea of tunneling. VPN tunneling involves establishing and maintaining a virtual network connection (that may contain intermediate hops).

A possible scenario of VPN using virtual connections is shown in Figure 1. The switching nodes (*a, b, c, d* and *f*) are available to provide a VC to a desired destination through the links (*ab, bd, df, ac, ce, ef, cd, de*).

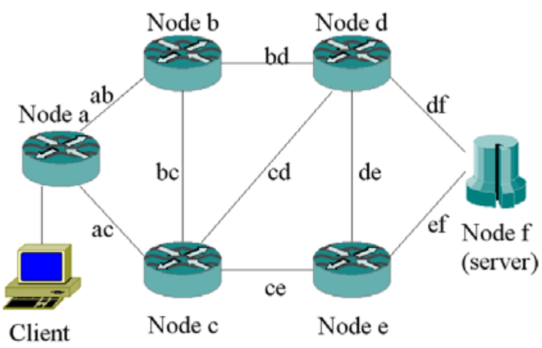

Figure 1. Network Configuration

As we see there are a lot of logical paths that could be obtained using this network structure but we will be focusing only on the paths shown in Table 1.

Path	Route
1	a-ab-b-bd-d-df
2	a-ac-c-ce-e-ef
3	a-ab-b-bd-d-dc-c-ce-e-ef
4	a-ac-c-cd-d-de-e-ef

Table 1. Path descriptions

We consider the network is said to be "up" if at least one of the four possible paths is available for communication. For a particular path to be available, all the nodes and links in corresponding route must be available. Note that failure of a particular link or node may result in unavailability of more than one path. For example if node *c* fails, paths 2, 3, 4 become unavailable. In the next sections, we show how all these aspects are captured in various reliability and availability models.

4. SYSTEM MODELLING

4.1 SRN Reliability Model

The SRN reliability model for our network example is shown in Figure 2. The net specification now only contains places and transitions which represent failure behavior for each individual component. The failure of any node or link may cause failure of one or more VC. System fault tree is "encoded" as a set of boolean functions. A simple halting condition represents failure of the whole network system. Table 2 shows the list of boolean functions which indicate failures of some sub-paths of the network, failures of VCs and halting condition of the whole network system. When the halting condition is executed we consider that the system goes down. The reward rate *r* for unreliability is assigned in terms of the boolean function for the halting condition as:

if (VC1 ∧ VC2 ∧ VC3 ∧ VC4)
    return 1; (system is unreliable)
else
    return 0; (system is reliable)

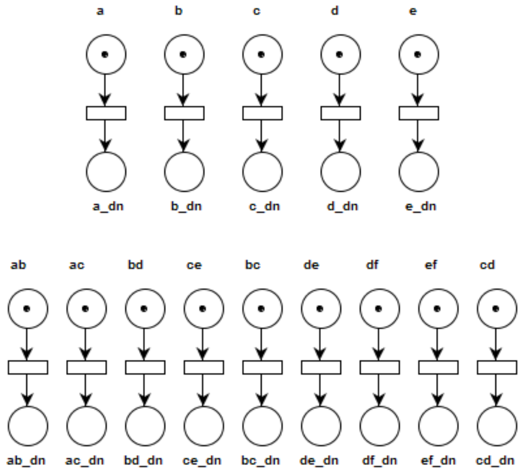

Figure 2. SRN Reliability Model

Name	Function
G1	#(a_dn) == 1
G2	#(ab_dn) == 1 ∨ #(b_dn) == 1 ∨ #(bd_dn) == 1
G3	#(c_dn) == 1 ∨ #(cd_dn) == 1 ∨ #(d_dn) == 1
G4	#(ce_dn) == 1 ∨ #(e_dn) == 1 ∨ #(ef_dn) == 1
G5	#(d_dn) == 1 ∨ #(df_dn) == 1
G6	#(d_dn) == 1 ∨ #(de_dn) == 1 ∨ #(e_dn) == 1 ∨ #(ef_dn) == 1
G7	#(ac_dn) == 1 ∨ #(c_dn) == 1 ∨ #(ce_dn) == 1
Virtual Connections	
VC1	G1 ∨ G2 ∨ G5
VC2	G1 ∨ G7 ∨ G4
VC3	G1 ∨ G2 ∨ G3 ∨ G4
VC4	G1 ∨ #(ac_dn==1) ∨ G3 ∨ G6
Halting Condition	
if (VC1 ∧ VC2 ∧ VC3 ∧ VC4) then network goes down	

Table 2. List of boolean functions

4.2 SRN Availability Model

In this case we consider the network system with repairs allowed. Each component has its own repair facility and is independent of failure and other component. The SRN simply consists of the failure and repair behavior of the system (Figure 3). The system being up or down is still encoded by exactly the same functions as listed in Table 2 with the exception that there is no halting condition. Instead, a reward function, which when evaluated by solving the underlying Markov reward model, gives the instantaneous and steady state availability of the system. The reward rate r for availability of this system is assigned in terms of the boolean functions defined earlier as:

if (VC1 ∧ VC2 ∧ VC3 ∧ VC4)
    return 0; (system is unavailable)
else
    return 1; (system is available)

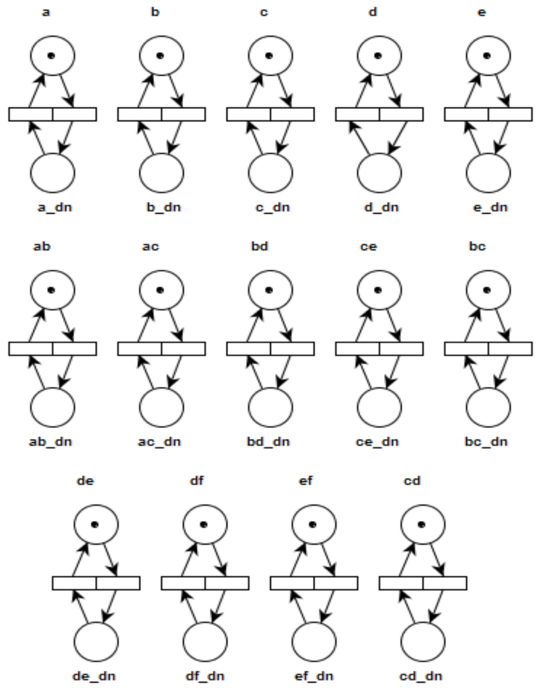

Figure 3. SRN model with independent repair

5. NUMERIC RESULTS

The SRN models were solved using the SPNP (Stochastic Petri Net Package). SPNP [12] provides support for specifying the SRN using a "C" like language and allows for the modeler to do steady state, transient, cumulative transient and sensitivity analysis. We use transient analysis because our goal is to perform network system analysis during the appropriate amout of time.

In order to assign values for the failure rates we chose a MTTF for nodes and links equal to 30 and 15 days correspondingly. For the repair rates for nodes and links we assign values equal to 6 and 4 items per day correspondingly.

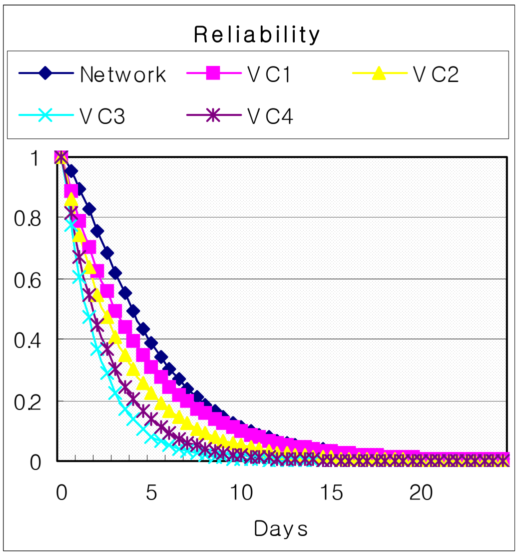

Figure 4. Reliability of the network

The results of reliability analysis obtained by solving SRN model shown in Figure 4. The network system works over certain amount of time without repair facility and after some time goes down. In our example the system is considered as non-operational after 15 days after start.

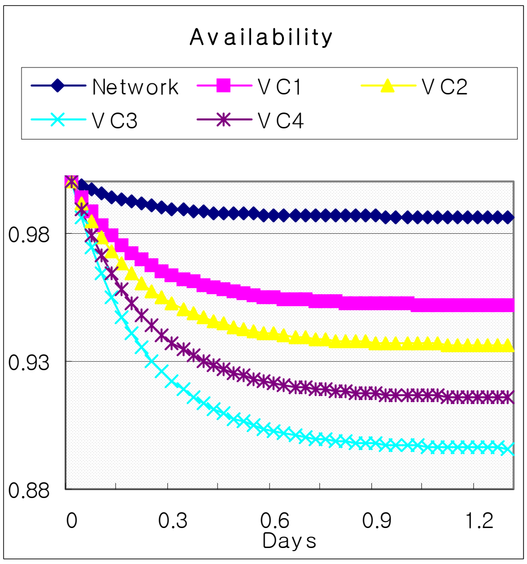

Figure 5. Availability of the network

The results from solving SRN availability model shown in Figure 5. In this case we can say that the availability of the network system goes down during some time and then it does not decrease. In our example we can say that the network system availability is higher than 98%.

6. CONCLUSION

In this paper we reviewed reliability and availability modelling techniques by applying them to the network system consisting of nodes and links. We developed SRN models for both cases and got the numerical results using SPNP package.

▮ REFERENCES ▮

[1] K. S. Trivedi, R. Sahner, A. Puliafito, "Performance and Reliability Analysis of Computer Systems" , 1996

[2] J. L. Peterson, "Petri Net Theory and the Modeling of Systems", Prentice Hall, Inc., Englewood Cliffs, 1981.

[3] M. Ajmone-Marsan, D. Kartson, G. Conte, and S. Donatelli, Modelling with Generalized Stochastic Petri Nets, John Wiley & Sons, Inc., New York, NY, 1995.

[4] G. Bolch, S. Greiner, H. de Meer, and K. S. Trivedi, Queueing Networks and Markov Chains, Modeling and Performance Evaluation with Computer Science Application. John Wiley & Sons, New York, NY, 1998.

[5] Balakrishnan, Meera and Trivedi, K. S., Componentwise Decomposition for an Efficient Reliability Computation of Systems with Repairable Components, Proc. Twenty-fifth International Symposium on Fault-Tolerant Computing, Pasadena, CA, July 1995.

[6] J. L. Peterson, Petri Net Theory and the Modeling of Systems. Englewood Cliffs, NJ: Prentice-Hall, 1981.

[7] German, R., Logothetis, D., and Trivedi, K., Transient Analysis of Markov Regenerative Stochastic Petri Nets: a Comparison of Approaches, in Petri Net Performance Models, PNPM'95, 1995.

[8] Stochastic Petri Nets and Their Applications to Performance Analysis of Computer Networks, K.S.Trivedi, and H. Sun, Invited paper at International Conference on "Operational Research For a Better Tomorrow", New Delhi, Dec. 24, 1998

[9] Numerical Transient Solution of Finite Markovian Queueing Systems, J. Muppala and K. S. Trivedi, in: Queueing and Related Models, U. N. Bhat and I. V. Basawa (ed.), pp. 262-284, Oxford University Press, 1992.

[10] Transient Analysis of Markov and Markov Reward Models, K. S. Trivedi, A. Reibman, and Roger Smith, in: Computer Performance and Reliability, G. Iazeolla , P. J. Courtois and O. J. Boxma (eds.), Elsevier Science Publishers B.V. (North-Holland), 1988, pp. 535-545.

[11] Markov and Markov Reward Models: A Survey of Numerical Approaches, A. Reibman and R. M. Smith, and K. S. Trivedi, European Journal of Operations Research, Vol. 40, pp. 257-267, 1989.

[12] G. Giardo, K. S. Trivedi: SPNP Users Manual Version 6.0. Technical report, Duke Univ., 1999.