The influence of application processes on performance reliability

Yue Zhang, Ning Huang[a], Weiqiang Wu and Shuo Zhang

School of Reliability and Systems Engineering, Beihang University, Beijing, China

Abstract. Application process has significant influence on network performance reliability, however, previous studies mainly focus on the impact of network characteristics (such as routing and traffic) on network performance reliability and ignore the influence of application process. In this paper, a multi-tiers model is proposed to capture the process of application. Furthermore, an application paths matrix is proposed to describe all the transmission paths of the application process in the network. Based on the application paths matrix, the boundary of the critical point of the traffic generation rate is derived theoretically, below the lower limit of the boundary the network stays in free-flow state and is reliable, while above the upper limit of the boundary the congestion occurs and the network performance reliability decreases. Furthermore, the effect of the boundary is verified in numerical simulations. The simulation results also show that congestions occurred in middle-tier host nodes could bring more influence on the network performance reliability.

1 Introduction

Network performance reliability is determined by both the network infrastructures and applications [1]. The decrease of network performance reliability is usually caused by congestion, which is reasoned by the heavy use of network applications [2]. Numerous works related to the performance of part nodes in the network are usually based on the queueing theory or network calculus theory, which is difficult to evaluate the performance of the total network due to the computational complexity [3,4]. By the end of the 20th century, the researches in complex network have proved that the network collapses above a certain traffic threshold and a traffic dynamics analysis method based on the mean-field theory is proposed to capture the performance of the whole network [5,6]. Since then numerous works focused on the performance analysis of whole network.

In particular, Ohira and Sawatari proposed the initial model to infer the network congestion in 1998 [6], in which the traffic is generated according to the packets generation rate λ and congestion occurs when λ is above the certain packet generation critical point λ_c. Since then numerous works follow this basic idea and concentrate on the shifting of critical points that the network turns from a free-flow state to congestion state [7]. Researchers firstly find that the network structure is important to the congestion of networks [8, 9], such as Toroczkai et.al have found that the scale-free networks are less prone to congestion than the random networks [10]. Since improving the routing strategy is a more practical and effective way to improve network performance than changing the physical

[a] Corresponding author : hn@buaa.edu.cn

structure of the network, so some researchers have focused on the influence of routing on the congestion [11,12], such as Wang et al. have proposed an efficient local routing strategy to improve the network performance [13]. Furthermore, in a specific network structure with particular routing strategy, more researches about the mitigating and controlling traffic congestion have been related to the resource allocation methods, especially, the node delivery capacity allocation has attracted much interest [14-16]. Recently, some researchers realized that dependence relations between networks is important factor to the network performance and some works have analyzed the influence of dependence relations on the congestion [17-19].

The above works have analyzed the influence of traffic or the network characteristics (such as routing or structure) on performance. However, modern Internet applications actually shows process feature, which means that the application is consisted by multiple tiers and each application tier provides certain function to its preceding tier and uses the functionality provided by its successor to carry out its part of the overall application process [20,21], but the above works ignored the multi-tiers application process. Although some researchers analyzed the performance of multi-tiers application [22,23], they are basically based on queuing theory and difficult to evaluate the performance of whole network.

In this paper, we focus on the influence of application process on the network performance reliability. The multi-tiers application model and the application paths matrix are proposed to help to analyze the influence of application process on network performance reliability. The boundary of the critical point of packets generation rate is estimated. We studied the influence of application process on the network performance reliability with a Lattice network topology. Some interesting results are observed in this paper and could be utilized to provide advice to network administrators and application providers.

2 MULTI-TIERS APPLICATION AND TRAFFIC

The following concepts are defined to help understand the relationship between the applications [24]:

A Service: is a function that a network provides to users;

An Application: is a procedure or service used by customers with some demanded performance requirements.

In the following, the service particularly describe the function that host node could provide, while the application describe the procedure provided to the end user, which is consisted by multiple services.

2.1 Multi-Tiers application model

In current, most Web applications are designed as multi-tier systems [21], for example, the ecommerce Websites e-bay and the web-based information systems such as Wikipedia. The multi-tier applications consist of multiple tiers though each tier hosts server(s) with identical functionality. Each tier provides a predefined service to the successor tier and receives services from its preceding tier. An 3-tiers application used in [23] is illustrated as an example in Fig.1. The 3-tiers application consist of the web-tier, application-tier and data-tier: the web-tier receives requests from end-users and provide them to access the applications; the application-tier receive requests from the Web-tier and look up information in the database; The data-tier basically stores applications data.

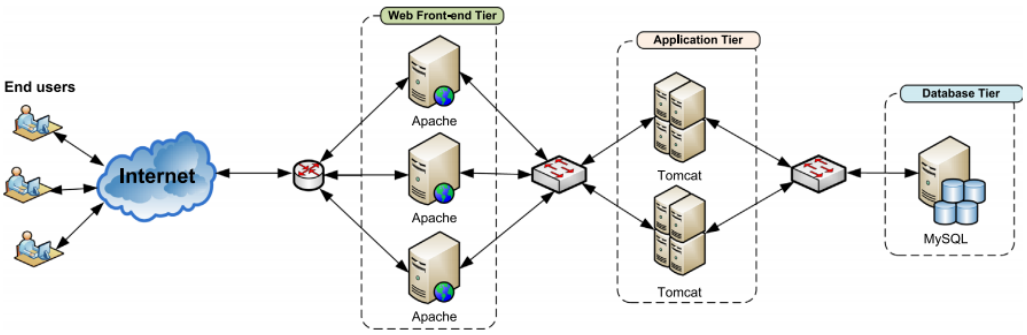

Figure 1. An example of 3-tiers application in internet [23].

For an application deployed in the network with M tiers denoted by T_1, …, T_M, the packet is processed exactly once by tier T_i and then forwarded to tier T_i+1 for further processing until it reaches T_M. In particular, T_1 is the origin tier of application packet and T_M is the terminal tier of application packet. If let $\boldsymbol{APP_M}$ denote the process of application with M tiers, then it could be describe as $\boldsymbol{APP_M}$=$\{T_1, T_2,\ldots, T_M\}$. Moreover, each tier could corresponds to multiple host nodes, let S_i denote the set of host nodes for T_i and card(S_i) denote the cardinal number of set S_i.

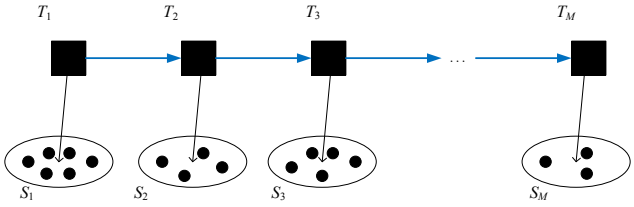

Figure 2. An example of application process of $\boldsymbol{APP_M}$ with $\boldsymbol{M}$-tiers.

Fig.2 present an illustrative example of the application process with M tiers. It shows that each tier T_i corresponds to a node set S_i which could provide identical functionality in the network, such as a set consist by 6 nodes S_1 could provide the service of T_1.

2.2 Application paths matrix

In order to better understand the influence of application process on the performance reliability, in this section, we introduce the application paths matrix **P**. Let us consider the application process $\boldsymbol{APP_M}$ with M different tiers. The corresponding host nodes set is $\{S_1, S_2,\ldots, S_M\}$. Let ***path*** denote the path of application process in the network, ***path*** could be described as:

$$path = \left[a_{1,1} \quad a_{1,2} \quad \ldots \quad a_{1,card(S_1)} \quad a_{2,1} \quad a_{2,2} \quad \ldots \quad a_{2,card(S_2)} \quad \ldots \quad a_{M,1} \quad a_{M,2} \quad \ldots \quad a_{M,card(S_M)} \right] \tag{1}$$

where $a_{k,l}(1 \le k \le M, 1 \le l \le card(S_l))$ is the host node for T_k. If this path pass through host node $a_{k,l}$, then $a_{k,l} = 1$, else $a_{k,l} = 0$. The application process matrix **P** represents for all paths of application process in the network, so **P** could be described as follows:

$$\mathrm{P} = \begin{bmatrix} path_1 \\ path_2 \\ \ldots \\ path_{card(S_1)} \end{bmatrix}_{card(S_1) \times (\sum_{i=1}^{M} card(S_i))} = \begin{bmatrix} a(1)_{1,1} & a(1)_{1,2} & \ldots & a(1)_{M,card(S_M)} \\ a(2)_{1,1} & a(2)_{1,2} & \ldots & a(2)_{M,card(S_M)} \\ \ldots & \ldots & & \ldots \\ a(card(S_1))_{1,1} & a(card(S_1))_{1,2} & \ldots & a(card(S_1))_{M,card(S_M)} \end{bmatrix}_{card(S_1) \times (\sum_{i=1}^{M} card(S_i))} \tag{2}$$

If $path_i$ pass through host node $a_{k,l}$, then $a(i)_{k,l}=1$, else $a(i)_{k,l}=0$. Let λ_i denote the packets generation rate of $path_i$. In this paper, we assumed that $\lambda_i=\lambda_j=\lambda,(1\le i,j\le M)$. If we use **Z** denote the load of host node, then **Z** could be described as:

$$Z=(z_{1,1},z_{1,2},...,z_{1,card(S_1)},z_{2,1},z_{2,2},...,z_{2,card(S_2)},z_{M,1},z_{M,2},...,z_{M,card(S_M)}) \tag{3}$$

where $z_{i,j}=\sum_{l=1}^{card(S_i)}\lambda_l a(l)_{i,j}=\sum_{l=1}^{card(S_i)}\lambda a(l)_{i,j}$.

2.3 Network traffic model

In fact, the generation and transmission of traffic is influenced by the application process in the network. So in this paper, we have improved the traffic model used in [9,13] by considering the application process. The details of the traffic model in a network with $\boldsymbol{APP_M}$ could be:

1) Traffic generation process: in a network with application process $\boldsymbol{APP_M}$=$\{T_1, T_2,\ldots, T_M\}$, the packet is originated in the host nodes set S_1 of T_1. We assume the packets are generated in the nodes set S_1 according to the packets generation rate λ, below which a packet is generated while above which no packet is generated each simulation time.

2) Traffic delivery process: the packets are delivered along with the application process. In particular, the packets generated in S_1 will be transported to S_2 and wait for the service from T_2. After the service of T_2, it will be transported to S_3 and wait for the service of T_3. Repeat the above process, until it reaches to S_M. Moreover, packets are delivered from S_i to S_{i+1} along the shortest path routing strategy. If there are more than one shortest path, then we choose one shortest path randomly. All nodes have the same delivering capacity C_d, that is, the maximal number of packets each node can deliver during one simulation time step. All services have the server capacity C_s, that is, the maximal number of packets each server can serve during one simulation time. The "first-in-first-out" rule is applied at each queue of the nodes and at each serve queue of server nodes in the network.

3) Traffic removing process: the packet will be removed from the network after it served by the terminal tier T_M.

3 Network performance reliability model

In this paper, the basic idea of the network performance reliability is that: Let T_k denote the travel time of the packet k and T^m denote the max acceptable delay that user could tolerance. If $T_k > T^m$, then this packet is past its acceptable time, which means that the packet transmission process is unreliable. The travel time T is consist by the delivery time between every two tiers and the service time at each tier. So in a network with M-tiered application process $\boldsymbol{APP_M}$, the travel time T could be measured as:

$$T=\sum_{i=1}^{M}T_s(i)+\sum_{i=1}^{M-1}T_d(i,i+1) \tag{4}$$

where $T_s(i)$ denote the service time at tier i and $T_d(i, i+1)$ denote the delivery time between tier i and $i+1$. Let $P(t)$ denote the set of the packets in the network at time t, $card(P, t)$ is cardinal number of the set $P(t)$; Let $\{S(t) \mid s\in P(t),\ T_s > T_s^m\}$ denote the set of the "unreliable" packets in the network at time t, $card(S, t)$ is the cardinal number of the set $S(t)$. Let $R(t)$ denote the network performance reliability at time t, $R(t)$ could be described as:

$$R(t)=1-\frac{card(S,t)}{card(P,t)} \tag{5}$$

Furthermore, if we assumed that the max acceptable delay T^m could be the max travel time when the network in free-flow state in this paper, then the network is reliable when it stays in free-flow state and obviously congestion have great influence on the network performance reliability.

Actually, the application process could be divided into the delivery process between every two consecutive tiers and the service process at each tier. So if congestion occurs in the delivery process or the service process, the network reliability will decrease. The order parameter η is widely used to measure the network states in complex network science. If η=0, it indicates that the network system is in the free-flow state; While if η>0, network turns into congested state 9:

$$\eta = \lim_{t\to\infty} \frac{1}{\lambda S} \frac{<\Delta Z_p>}{\Delta t} \tag{6}$$

where $Z_p(t)$ denote the number of packets in the system at time t, $\Delta Z_p = Z_p(t+\Delta t) - Z_p(t)$, <…> indicates average over time windows of width Δt, λ is the traffic generation rate and S is the number of nodes generating packets.

Related works in complex networks focus on the delivery process of packets in the network, and a critical point of packets generation rate λ_c^d is observed, below which $\eta \to 0$ and the packets are delivered freely in the network, while above which $\eta > 0$ and congestion occurs in the delivery process of packets. In the following, we focus on the relation between the packets generation rate λ and the service process in the network and discuss the critical point of packets generation rate λ_c^s from the service process view. The basic idea is that: if the average load on host node exceeds its server capacity, then congestion occurs. So the critical point of packets generation rate λ_c^s could be calculated as follows:

$$\max\{z_{i,j}\} = C_s \tag{7}$$

substituting (3) to (7), we could have:

$$\lambda_c^s \max\{\sum_{l=1}^{card(S_1)} a(l)_{i,j}\} = C_s \tag{8}$$

Considering two limiting cases: 1) if the load are uniformly distributed on host nodes, then we could calculate the maximum value of λ_c^s. The average load z_i of host node for tier T_i could be:

$$z_i = \lambda_c^s card(S_1) / card(S_i) \quad \{1 \le i \le M\} \tag{9}$$

Equation (9) could be simplified as:

$$\lambda_{c,\max}^s card(S_1) / \min\{card(S_i)\} = C_s$$
$$\lambda_{c,\max}^s = \frac{C_s \min\{card(S_i)\}}{card(S_1)} \tag{10}$$

2) for the tier T_i with minimum card(S_i), if the load are centralized in one particular host node j and other host nodes all support load generated by one host node for T_1 respectively, then we could calculate the minimum value of λ_c^s. The average load $z_{i,j}$ of host node could be:

$$z_i = \lambda_c^s (card(S_1) - \min\{card(S_i)\} + 1) \tag{11}$$

Equation (9) could be simplified as:

$$\lambda_c^s(card(S_1)-\min\{card(S_i)\}+1)=C_s$$
$$\lambda_{c,\min}^s=\frac{C_s}{(card(S_1)-\min\{card(S_i)\}+1)} \tag{12}$$

So in a network with application process ***APP*$_M$**, with host nodes set$\{S_1, S_2,\ldots, S_m\}$. Specially, if $card(S_1)=\min\{card(S_i)\}$ so we could find that $\lambda_{c,\max}^s=\lambda_{c,\min}^s$. In summary, if $\lambda>\lambda_{c,\max}^s$, the congestion occurs at the server hosts nodes. However, if $\lambda<\lambda_{c,\min}^s$, the server host nodes stay in free-flow state and the network state is determined by the delivery process of packets. Finally, the boundary of the critical point λ_c could be given by $(\lambda_{c,\min},\lambda_{c,\max})$, where $\lambda_{c,\max}$ could be $\min\{\lambda_c^d,\lambda_{c,\max}^s\}$, above which the network turns into congestion state; and $\lambda_{c,\min}$ could be $\min\{\lambda_c^d,\lambda_{c,\max}^s\}$, below which the network stays in free-flow state. In the following, we focus on the influence of application process on network reliability when packets generation rate λ approximate to λ_c.

4 Numerical results and analysis

In this section, we analyze the network reliability considering the 3-tiers application on $L\times L$ lattice network (L=32) with Periodic boundary conditions. The host nodes set for each tier could be S_1, S_2 and S_3. Moreover, we assumed that $card(S_1)\geq card(S_2)\geq card(S_3)$. The hosts nodes are distributed in the network randomly and all hosts nodes have same server ability C_s=2. According to 9, the critical point λ_c^d of the delivery process $\lambda_c^d=2/L=0.125$. In the following, we assume the max acceptable delay $T^m=(M-1)\times D$. Finally, all nodes have same delivering capacity C_d=2 and the buffer size is assumed as infinity.

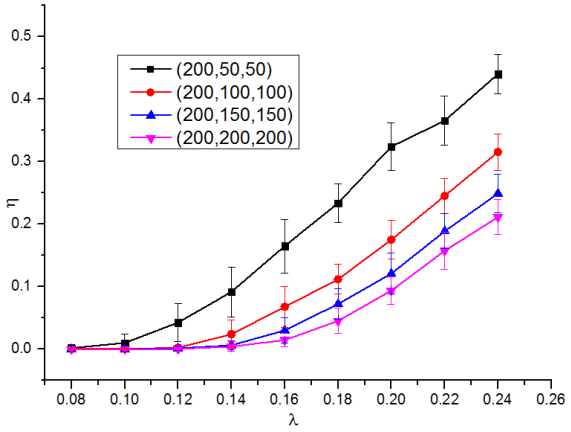

Figure 3. The relation between η and λ (each point is averaged from 30 independent simulations).

According to (9) and (12) we could find out that the critical point λ_c is related to the minimum value of $card(S_i)$. In Fig.3, we analyzed the critical point λ_c with different $card(S_1)$, $card(S_2)$ and $card(S_3)$. The curves with ($card(S_1)$, $card(S_2)$, $card(S_3)$) represent for the simulation results when the cardinal number of hosts set for each tier is $card(S_1)$, $card(S_2)$ and $card(S_3)$. So according to (9) and (12), we could calculate the maximum and minimum value of the critical point λ_c for each ($card(S_1)$, $card(S_2)$, $card(S_3)$) simulation scenario. The results are (0.013, 0.5), (0.02, 1), (0.04, 1.3) and (0.12, 2). Fig.3 shows the critical point λ_c stays in the boundary set illustrated in section 3, which confirmed the validity of boundary set. Furthermore, the results in section 2 also show that the critical point increases with the increase of $card(S_2)$ and $card(S_3)$.

In the following, we discussed the network performance reliability and the packets generation rate. Fig.4 shows network performance reliability against simulation time with different packets generation rate λ . The cardinal number of hosts set for each tier is (200,200,200) and the critical point λ_c is 0.14 according to Fig.3. Fig.4 shows that if $\lambda < \lambda_c$, the network stays in free-flow state and the performance reliability approximate to 1, while if $\lambda > \lambda_c$, the congestion occurs in the network and the network performance reliability falls quickly. Moreover, the larger λ is, the more unreliable the network is.

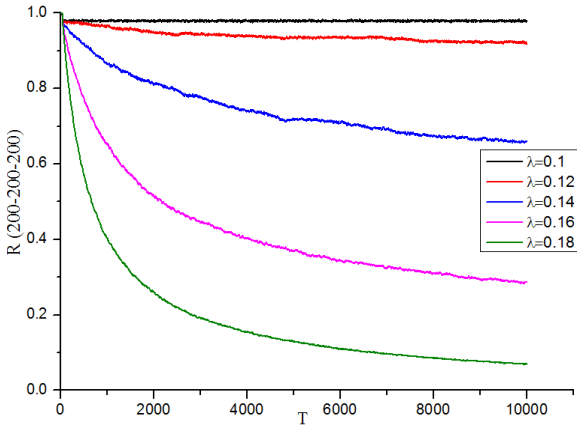

Figure 4. The relation between the *R* and λ (each point is averaged from 30 independent simulations).

Usually deploying more servers in the network could improve the network reliability. Since the network becomes unreliable when $\lambda > \lambda_c$,so in Fig.5 we discussed the relation between the reliability and the server hosts numbers with fixed server hosts numbers and terminal hosts numbers. The packets generation rate λ is 0.14. It shows that the network performance reliability increase with the increase of the server hosts numbers for T_2.

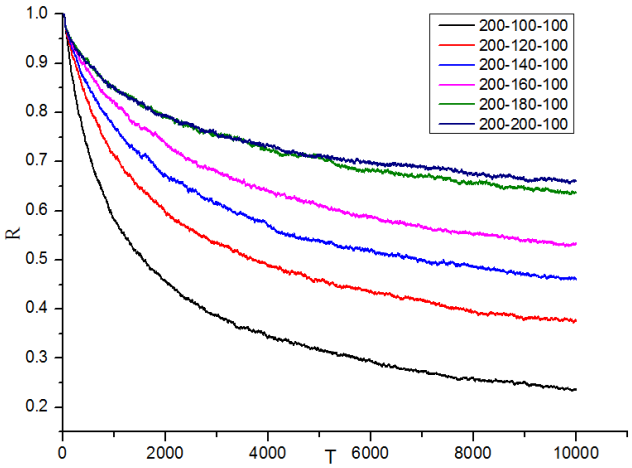

Figure 5. The relation between *R* and middle server host numbers(each point is averaged from 30 independent simulations).

A general problem that network administrators and service providers care about is that how to allocate the server hosts to each tiers when the total server hosts and the server hosts for T_1 is fixed. Since the network becomes unreliable when $\lambda > \lambda_c$, we discussed the influence of the server hosts number on the network performance reliability when $\lambda > \lambda_c$ in Fig.6. The total server hosts number for T_2 and T_3 is 250. Fig.6 shows that the network performance reliability decreases as the decrease of the

server hosts numbers for T_2. Since the less the server hosts number for T_i is, the heavier congestion in T_i is. Fig.6 shows that the congestion occurred in middle Tier could bring more influence on the reliability than the terminal tier.

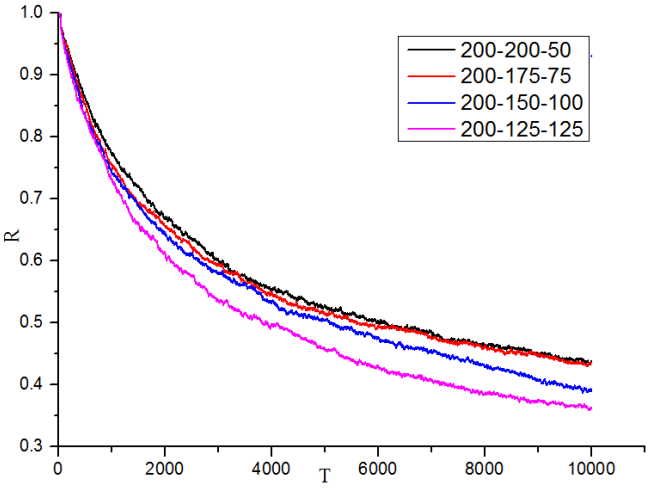

Figure 6. The relation between *R* and server hosts numbers (each point is averaged from 30 independent simulations).

5 Conclusion

In this paper, a general multi-tiers application model is proposed to capture the multi-tier process of application and an application paths matrix is proposed to capture all paths of the application process. The boundary of the critical point of the packets generation rate is estimated, below the lower limit of the boundary the network stays in free-flow state and the network is reliable, while above the upper limit of the boundary the network stays in congestion state and the network performance reliability decreases. Furthermore, the effective of the boundary is verified in numerical simulations on 32×32 lattice network with 3-tiers applications. The simulation results also shows that the congestion occurs in the middle tier could bring more influence on the network performance reliability. We believe that the results and methods in this paper are helpful to optimize the application processes in network. An open question to be solved in future work is that in this paper we considered the lattice network topology, more network structures could be studied in future, for example, the BA network or the ER network

References

1. N. Huang and Z.T. Wu, J. Syst. Eng. Electron, **35**(12), 2651-2660 (2013)
2. R. Li, N. Huang and R. Kang, *IEEE Proceedings Reliability and Maintainability Symposium* (*RAMS*) (San Jose, USA, 2010)
3. J.Y.L. Boudec and P. Thiran, *Network calculus: a theory of deterministic queuing systems for the internet* (Berlin, German, 2001)
4. T.G. Robertazzi and G. Thomas, *Computer networks and systems: queueing theory and performance evaluation* (New York, USA, 2000)
5. A. Arenas, A. Díaz-Guilera and R. Guimera, Phys. Rev. Lett., **86**(14), 3196 (2001)
6. T. Ohira and R. Sawatari, Phys. Rev. E, **58**(1), 193 (1998)
7. S. Chen, et al., Math. Probl. Eng., 256-267 (2012)
8. P. Holme, Adv. Complex Syst., **6**(02), 163-176 (2003)
9. L. Zhao, et al., Phys. Rev. E, **71**(2), 026125 (2005)

10. Z. Toroczkai and K.E. Bassler, Nature, **428**(6984), 716 (2004)
11. P. Echenique, J. Gómez-Gardeñes and Y. Moreno, Phys. Rev. E, **70**(5), 056105 (2004)
12. G. Yan, T. Zhou, B. Hu, Z.Q. Fu and B.H. Wang, Phys. Rev. E, **73**(4), 046108 (2006)
13. W.X. Wang, C.Y. Yin and G. Yan, Phys. Rev. E, **74**(1), 016101 (2006)
14. J. Ma, W. Han, Q. Guo and Z. Wang, Physica A, **456**, 281-287 (2016)
15. Y. Xia and D. Hill, EPL, **89**(5), 58004 (2010)
16. G. Q. Zhang, et al., Physica A, **390**(2), 387-391 (2011)
17. C.G. Gu, et al., Phys. Rev. E, **84**(2), 026101 (2011)
18. R. G. Morris and M. Barthelemy, Phys. Rev. Lett., **109**(12), 128703 (2012)
19. F. Tan, J. Wu, Y. X. Xia and C.K. Tse, Phys. Rev. E, **89**(6), 062813 (2014)
20. D. Huang, B. He and C. Miao, IEEE Commun. Surv. Tutor., **16**(3), 1574-1590 (2014)
21. D.K. Barry and P.J.Gannon, *Web services and service-oriented architectures: the savvy manager's guide* (San Francisco, USA ,2003)
22. B. Urgaonkar, G. Pacifici, P. Shenoy, M. Spreitzer and A. Tantawi, *ACM SIGMETRICS Performance Evaluation Review* (Banff, canada, 2005)
23. K. RahimiZadeh, M. Analoui, P. kabiri and B. Javadi, J. Netw. Comput. Appl., **56**, 166-187 (2015)
24. N. Huang, Y. Chen, D. Hou and L.D. Xing, J. Syst. Eng. Electron., **22**(6), 1030-1036 (2011)