ISSN : 0974 - 7435 Volume 8 Issue 3

BioTechnology
An Indian Journal

FULL PAPER

BTAIJ, 8(3), 2013 [411-420]

A bio-pepa based vulnerability diffusion model for WSN

Hongwu Lv*, Huiqiang Wang, Amjad Ali, Guangsheng Feng
College of computer science and technology, Harbin Engineering University, No.145 Nantong Street, Nangang District, Harbin, 150001, (CHINA)
E-mail : lvhongwu@hrbeu.edu.cn

ABSTRACT

Vulnerability is the essential reason of security and dependability, which has become a serious problem for Wireless Sensor Network (WSN) while a lot of additional vulnerabilities are brought out by interactions among nodes and even more than inherent ones. A vulnerability diffusion model for WSN is proposed in this paper. Compared to current vulnerability diffusion model, we take the characteristic of heterogeneous of vulnerability diffusion into consideration, that is, the vulnerability diffusion between clusters is not the same as in a single cluster. Inspired by epidemiological model, we use the characteristics of static hierarchy of Bio-PEPA, a process algebra for the analysis of biological systems, to describe vulnerability diffusion in different scenes, including vulnerability diffusion in single cluster, vulnerability diffusion in multi-cluster without migration or with migration. The experimental results show that the vulnerability diffusion can be reduced by enhancing the recovery capability and decreasing the rate of vulnerability migration between clusters. Our works provide an insight into the nature of vulnerability propagation, and is useful to improve the security of WSN.

KEYWORDS

WSN;
Bio-PEPA;
Vulnerability analysis;
Vulnerability diffusion.

INTRODUCTION

For Wireless Sensor Network (WSN), vulnerability is the underlying causes that harm to security and reliability[1]. Recently, a lot of studies have pointed that more and more additional vulnerabilities are brought out by connections between nodes, even more than inherent ones. Vulnerability propagation has become a serious problem.

In WSNs, nodes usually need to cooperate with each other to accomplish tasks together, such as transmitting or collecting information. If there only a single cluster, the interaction between nodes has a strong randomness and the kind of vulnerability diffusion can be seemed as homogeneous medium, which is just as current studies. For example, De[2] studied the vulnerability diffusion problems of multi-hop broadcast protocol in the WSN, and discussed the impacts from sojourn time, connectivity, recoverability with spy software.

But, the nodes of WSN are usually organized by cluster, the probability of connection between two nodes in different clusters is usually unequal with them in the

FULL PAPER

same cluster. Thus the vulnerability diffusion is generally heterogeneous on the entire system. Meanwhile, with the advances in mobile computing technology, there are more and more mobile nodes are brought into WSN and they join/ quit a cluster freely, which also play an important role to propagate vulnerability. So far, the above factors have not been considered in current vulnerability diffusion models in[2, 3]. Besides, as the scale of systems increasing, or even thousands of nodes, traditional methods such as state transition diagram, colored Petri nets facing serious problem of state space explosion[4]. For the shortcomings of existing models, we will study the vulnerability diffusion in and between clusters, as well as vulnerability migration between clusters while compromised nodes move to another clusters.

Inspired by epidemiological model, Bio-PEPA (Bio Performance evaluation algebra)[5], a formal language for biological systems is used to describe and analyze the diffusion processes of vulnerability in WSNs. The paper is organized as follows. At first, Bio-PEPA is simply introduced in section 2, and the vulnerability diffusion of WSN is modeled in section 3, and section 4 shows the experiment results of vulnerability diffusion, and we give a conclusion in section 5 at last.

Bio-PEPA

Bio-PEPA is a new formal language, which is born from biological networks and Performance Evaluation Process Algebra(PEPA). For the characteristics of static hierarchy[5], it is possible for Bio-PEPA to describe a series of separate location, and thus is suitable for the description of vulnerability diffusion between different clusters. At the same time, Bio-PEPA can be converted to Ordinary Differential Equations (ODEs), simplifying the process of solving large-scale computing systems composed by a large number of components.

The basic elements of Bio-PEPA includes two kinds, namely specie component *S* and model component *P*, the former is used to describe the behaviors of each species, and the latter is used to describe the initialization and interaction between species. The semantics of Bio-PEPA can be simplified as followings[5, 6].

$$\mathbf{S ::= (\alpha, k) \downarrow S \mid (\alpha, k) \uparrow S \mid (\alpha, (k_1, k_2)) \mid \odot S \mid S + S}$$

$$\mathbf{| A | S @ L} \quad (1)$$

$$\mathbf{P ::= P \underset{G}{\triangleright\!\triangleleft} P \mid S(x)} \quad (2)$$

Where the meaning of the each expression is defined as:

1) α represents an action type and its function rate is k.
2) There are three operation types, *op*="!|'!|™", representing the role of component *S* in a reaction. Specifically, "! is used to indicate a reactant, '! a product, ™" a modifier.
3) Because there are not chemical or biological reactions in this paper, according to the literature[6], we gives a new meaning of reaction, refers to the interaction between the nodes, or status changes. For a reaction, this paper allows the presence of non-simplest reaction, that is, A+B ®A+C or A+B ®2A. There are one basic types of k, Mass-Action. While there is a non-simplest reaction, $(\alpha,(k_1,k_2))$ ™" is used to represents changes of k, and k_1, k_2 are respectively coefficient of species before and after the reaction. In this case, $k=k_2-k_1$. A point worth noting is that, the formula $(\alpha,k)\ op\ S$ is omitted as $\alpha\ op\ S$ while k=1.
4) The operation + represents the choice between activities.
5) $A \overset{def}{=} S$ defines a constant.
6) The compartment *L* are static hierarchy, which allows us to represent most of the features of diffusion between different clusters. *C@L* represents the components *C* in position *L*. When only one location, *L* can be omitted. And in this paper, there is only one kind of compartment, which is called cluster.
7) The operation $\triangleright\!\triangleleft$ represents cooperation between activities, in which *G* is the set of activities synchronized in the process of cooperation. If $\triangleright\!\triangleleft_*$ is written, it means all the activities involved is needed to synchronize.
8) In *S*(*x*), *x*"$\mathbf{N}^+$ represents the population of species at initial time.

For the solution of Bio-PEPA, we can exploit different kinds of analysis methods, including CTMC(continuous time Markov chain), Gillespie simulation and ODEs[5]. When a system has a large scale,

the CTMC derived from Bio-PEPA will face a state space explosion problem, and Gillespie simulation takes a long time to run, so we choose ODEs method in this paper.

Due to the constraints of paper length, one can refers to[6] for more details of syntax and operational semantics.

VULNERABILITY DIFFUSION MODELS OF WSN

There are many causes that may leads to vulnerabilities of WSN, including bugs, spy-ware and malicious software. Recently, in addition to inherent vulnerabilities, more and more vulnerabilities break out for interconnected and unreasonable trust relationship[8]. Comparing to inherent vulnerabilities, vulnerability diffusion depending on the links between nodes follows with good mathematical statistical laws[4]. And instead of discussing the generating process of vulnerabilities, we mainly considers on diffusion process of weak points in this paper. Understanding of the laws of vulnerability diffusion will help to prevent the propagation of vulnerability, increasing system security at a low cost.

In order to study the problem simply, for a certain node, we only focus on whether it has vulnerability, without paying attention to the number of vulnerabilities it has. Learning from software vulnerabilities state model and the classic Epidemic Model SIR[7], the nodes in WSN can be classified into the following five types:

Healthful (H): The nodes does not contain known vulnerabilities, but they may be infected while connecting with the V_E and V_S;

Undisclosed(U): The nodes contain known vulnerabilities, but these vulnerabilities have not been detected.

Infected(I): The nodes have been. discovered to contain known vulnerabilities.

Failed(F): Vulnerabilities have not been repaired timely, leading to a failure of node.

Recovery(R): Recovered nodes. Repair methods may include simple disconnection, denial of service, as well as more effective approaches such as online upgrading, patching, reconfiguration, etc..

Denoting state set $S=\{H,U,I,F,R\}$, where vulnerable nodes constitute vulnerability state set $S_V=\{U,I,F\}$. According to the diffusion theory of vulnerability, vulnerability caused by connection spreads only between *H* and *U*, or *H* and *I*, without regard to the diffusion between *U* and *I* that not effecting on the results.

Under the above conditions, we will research the diffusion process of vulnerability at a fixed initial vulnerabilities in the time interval [0, *T*), $0<T<+\infty$.

Vulnerability diffusion in single cluster (VDS)

The model of VDS

Firstly, considering the simplest case that vulnerability diffusion happens only within a single cluster. Dropped the details, vulnerability diffusion within a single cluster can be summarized as the following nine rules.

[Interactions1]:

(1) <connection1> $H+U \rightarrow 2U$

H is infected by *U* with connections.

(2) <connection2> $H+I \rightarrow U+I$

H is infected by *I* with connections.

(3) <discovery> $U \rightarrow I$

U is discovered by header or base station and converted to *I*.

(4) <fail1> $U \rightarrow F$

U fails or has an error.

(5) <fail2> $I \rightarrow F$

I fails or has an error.

(6) <recovery1> $U \rightarrow R$

U is repaired

(7) <recovery2> $I \rightarrow R$

I is repaired

(8) <recovery3> $F \rightarrow R$

F is repaired

(9) <insecure> $R \rightarrow H$

Since there may still be other defects, *R* may transform to *H*.

For the rule of <connection1> and <connection2>, nodes become vulnerability while connecting to another vulnerability node. And for the invisibility nature of vulnerability diffusion, the node infected is thought to evolve to state *U* firstly, and *U* can be converted into *I* according to <discovery>.

For the rule of <insecure>, some vulnerabilities have not been completely repaired such as by interrupting connections; and other vulnerabilities such as security flaws of protocols can not be resolved locally. So nodes may still have some underlying defects, and some *R* will

FULL PAPER

be reconverted to *H* again.

Then we model the process of vulnerability diffusion in a single cluster by bio-PEPA. Denoting the actions set of interaction or connection as *Interactions*1={ <connection1>, <connection2>, <discovery>, <fail1>, <fail2>, <recovery1>, <recovery2>, <recovery3>, <insecure>}. For each reaction, the function rate is denoted as $r_\alpha, \alpha \in Interactions1$. In this section, we dose not considered nodes joined or deleted, that is, the total number of nodes is fixed in a certain period of time, and is denoted by n_{max}. At time t, the number of H,U,I,F,R are separately denoted by n_H, n_U, n_I, n_F, n_R According to the Mass-Action rule of stoichiometric coefficient, the functional rates of reactions are defined as,

$$\mathbf{f_{connection1}} = \mathbf{r_{connection1}} * \mathbf{n_H} * \mathbf{n_U}$$

$$\mathbf{f_{connection2}} = \mathbf{r_{connection2}} * \mathbf{n_H} * \mathbf{n_I}$$

$$\mathbf{f_{discovery}} = \mathbf{r_{discovery}} * \mathbf{n_U}$$

$$\mathbf{f_{fail1}} = \mathbf{r_{fail1}} * \mathbf{n_U} \quad \mathbf{f_{fail2}} = \mathbf{r_{fail2}} * \mathbf{n_I}$$

$$\mathbf{f_{recovery1}} = \mathbf{r_{recovery1}} * \mathbf{n_E} \quad \mathbf{f_{recovery2}} = \mathbf{r_{recovery2}} * \mathbf{n_S}$$

$$\mathbf{f_{recovery3}} = \mathbf{r_{recovery3}} * \mathbf{n_F} \quad \mathbf{f_{insecure}} = \mathbf{r_{insecure}} * \mathbf{n_R}.$$

Assuming the maximum number of components in a cluster n_{max} is N_M, and there is only one location *L*1, and without taking the correlationship between diffusion into account, let *step-size* = 1, thus various types of components can be described by bio-PEPA.

$$H \overset{def}{=} (connection1,1)\downarrow H + (connection2,1)\downarrow H + (insecure,1)\uparrow H;$$

$$U \overset{def}{=} (connection\ 1,(1,2))\odot U + (connection2,1)\uparrow U + (discovery,1)\downarrow U + (fail1,1)\downarrow U + (recovery1,1)\downarrow U;$$

$$I \overset{def}{=} (connection2,(1,1))\odot I + (discovery,1)\uparrow I + (fail2,1)\downarrow I + (recovery2,1)\downarrow I;$$

$$F \overset{def}{=} (fail1,1)\uparrow F + (fail2,1)\uparrow F + (recovery3,1)\downarrow F;$$

$$R \overset{def}{=} (recovery1,1)\uparrow R + (recovery2,1)\uparrow R + (recovery3,1)\uparrow R + (insecure,1)\downarrow R;$$

Set X_0=($N_H^0, N_U^0, N_I^0, N_F^0, N_R^0$) to be an initial vector at the beginning time then the process of VDS is modeled as,

$$H[N_H^0] \triangleright\!\triangleleft_* U[N_U^0] \triangleright\!\triangleleft_* I[N_I^0] F[] R[]$$

A case study

In this sub-section, a simple case is chosen to analyze the VDS model. We set the number of nodes to 300 in the experiment, that is N_{max}=300, the initial vector of each components to X_0=(270, 20, 10, 0, 0).

For the value of parameters, consider the most common scenario of random unicast, the connection probability between any two nodes are 1/N[4], but the nodes compromised may be infected with spywares or worms, so the components of *I* infected within a cluster may initiate to scan or connect, which means a higher connection probability. According to the test of vulnerability diffusion in paper[4], the probability of *I* connecting with other species to increase by 10 times, $r_{connection2}$ =10*1/N. In following experiments, we will consider the tested system with the lower and higher level of recoverability separately. Given that while there a week recovery capability, $r_{recovery1}$= $r_{recovery2}$= $1*10^{-3}$; and while having a strong capability of recover, let $r_{recovery1}$= 1.0, but not all component of *U* can be recovered for its implicit characteristic, let $r_{recovery2}$= 0.7. All other parameters' values are shown in TABLE 1. To improve the efficiency of solution, we use a tool called Eclipse Bio-PEPA Plug-in developed by the University of Edinburgh to help to solve ODEs[8].

TABLE 1 : The initial values of Parameters

Parameters	value	Parameters	value
$r_{discovery}$	0.5	$r_{recovery3}$	1.0
r_{fail1}	$5*10^{-2}$	$r_{insecure}$	0.9
r_{fail2}	$1*10^{-2}$		

While the system has a week recoverability and other parameters are set as in TABLE 1, the population of components changes with time just as in Figure 1. Seen from Figure1, at first, the components of *H* begin to become *U* and *I*. And after an interval of time, *U* are discovered or exploited to evolve to the components of *I*. At last, for a weak recoverability of WSN, only a small part of *I* be recovered and most of the nodes become vulnerability.

Next, we consider a strong recoverability of WSN while other parameters are set as TABLE 1 too. The population of components in this scene is shown in Fig-

FULL PAPER

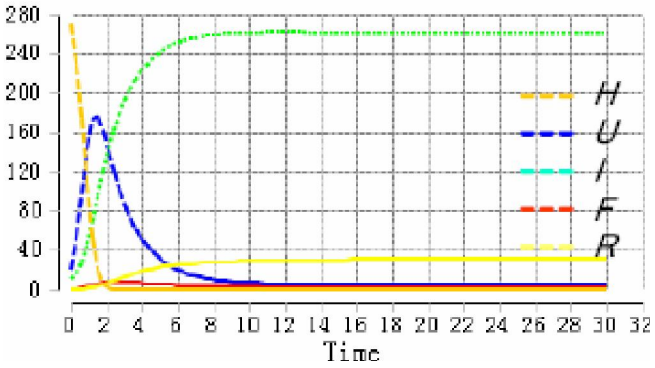

Figure 1 : The population of components while having week recoverability

ure 2. At the beginning, the number of *U* and *I* increases, and then most of them are detected and finally recovered to the component of *R* due to system upgraded, bug fixed or other repair methods. At the same time, a part of *H* is not infected for the existence of strong recoverability. Overall, with the growth of recoverability, vulnerability has been effectively limited to a smaller range.

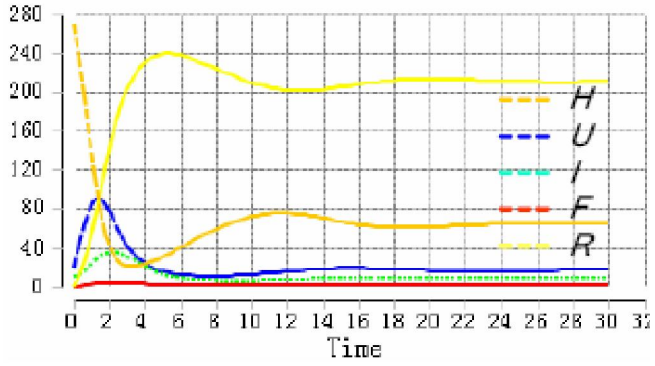

Figure 2 : The population of components while having strong recoverability

Base on the hypothesis of strong recoverability, assuming connection probability of both *U* and *I* with other components increase to 10*1/N, and other parameters stay at the same. At this scene, the population of components is shown in Figure3. At first the number of *U* and *I* significantly increases, and maximum number of components of *U* and *I* becomes larger than that of Figure 2, which means expanding the range of vulnerability diffusion. Since the existence of strong recoverability to repair the components of *U* and *I*, the number of *U* and *I* returns to a lower level again. Thus, frequent interactions and cooperations among nodes in a cluster can increase the diffusion of vulnerability.

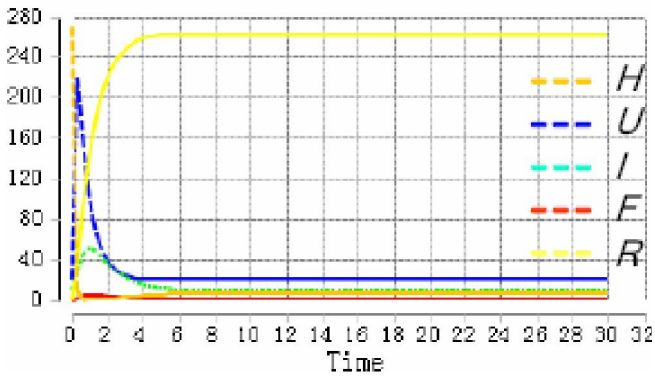

Figure 3 : The population of components while $r_{connection2}$ is reset to be $3.3*10^{-2}$

In order to test our method, the above three scenes is simulated by the algorithm of Gillespie too, and the population of each component shows that this two kinds of method is generally the same. But due to the constraints of paper length, we omit these pictures.

Vulnerability diffusion in multi-cluster

Vulnerability diffusion in multi-cluster without migration (VDM)

When a cluster composed of fixed nodes entirely or nodes do not switch between multiple clusters, it is thought to be not existence of node migration. In generally, we choose an example of three clusters to explain the process of VDM for simplicity, called *L*1, *L2* and *L*3. In this paper, we primarily focus on the impact from multi-cluster on vulnerability diffusion and keep the other parameters the same for all clusters, Thus all kinds of rate of activity in all the three clusters is set to be same, which is helpful to simplify the problem. The vulnerability diffusion between clusters also happens between *H* and *U* or between *U* and *I*. Except for the same regulars as VDS, there are also two other extra rules for VDM.

[Interactions2]: p, q" $\{L1, L2, L3\}, q$‘‘‘p.

<link1$_{pq}$>

$H@\ p+U@\ q \rightarrow U@\ p+\ U@q$;

<link2$_{pq}$>

$H@p+I@q \rightarrow U@p+\ I@q$;

As the assumption that each cluster is the same, the coefficients of diffusion reaction between clusters are also same, which can be computed by the rule of mass-action,

$f_{link1\,pq} = r_{link1\,pq} * n_{H@p} * n_{U@q}$ ÿ

FULL PAPER

$f_{link2\,pq} = r_{link2\,pq} * n_{H@p} * n_{I@q}$

The connection probability between clusters is a ratio of $r_{connection1}$, and is recorded as *link_ratio*" [0, 1]ÿthen the coefficients of diffusion reaction between clusters are defined as $r_{link1\,pq}$ =*link_ratio** ÿ= *link_ratio**ÿ*p, q*" {*L*1, *L*2, *L*3}, *q*‘‘*p*.

Vulnerability diffusion in multi-cluster with migration (VDMM)

If there are mobile nodes in a cluster, the vulnerability may be spreaded by migration of node. At this scene, Vulnerability Diffusion in Multi-Cluster with Migration is called as VDMM.

In this part, we also divide the system into 3 clusters, separately *L*1, *L*2 and *L*3, and the migration rate of nodes in all the three clusters is treated as the same to simplify the problem. Based on the model in subsection 3.2.1, relationship of components in model VDMM can be describe in Figure 4.

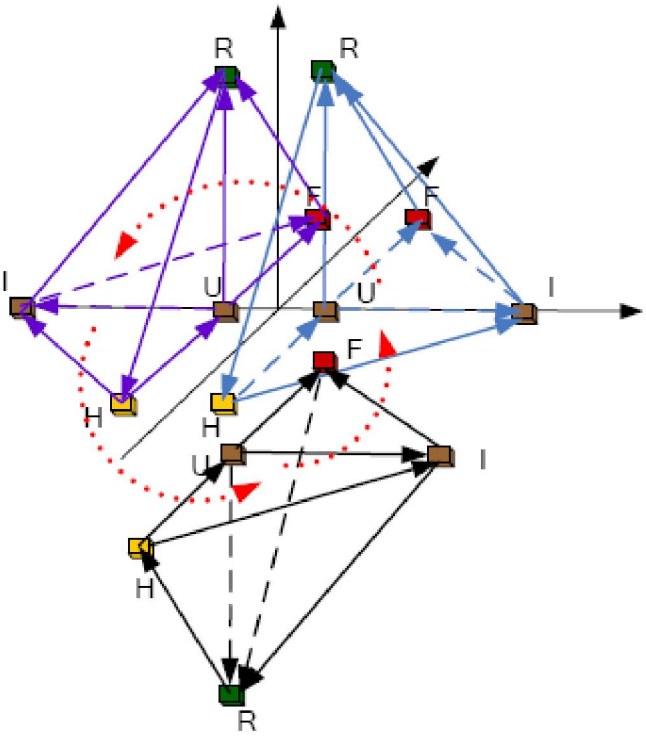

Figure 4 : The relationship of components in the model of VDMM

Assuming all the nodes can move to other clusters at a certain ratio, and the proportion of migrated nodes with the total is *migration_rate*" [0, 1]. The migration can be described by bio-PEPA as $(m_{ij}, specie[location\_i \to location\_j], (1,1))$. Then we add a new rule to vulnerability diffusion for VDMM.

[Interactions3]:

Let *p, q*" {*L*1, *L*2, *L*3}, *p*‘‘*q, specie*

<migration>

specie@p →*specie@q*;

The coefficient of migration between clusters are defined as,

=*migration_rate* *, where *i*" {*L*1, *L*2, *L*3}.

Then the model of VDMM is given.

Setting the maximum number of components in cluster *k* to be , in the time interval [0, *T*), , the process of vulnerability diffusion is modeled by Bio-PEPA as followings.

$$H@p \overset{def}{=} (link1_p,1)\downarrow H@p + (link2_p,1)\downarrow H@p + (insecure_p,1)\uparrow H@p + \sum_q (link1_{pq},1)\downarrow H@p + \sum_q (link2_{pq},1)\downarrow H@p + \sum_q (m1_{pq,W[p\to q]},(1,1))\downarrow H@p + \sum_q (m1_{qp,W[q\to p]},(1,1))\uparrow H@p + \sum_q (m2_{qp,R[q\to p]},(1,1))\uparrow H@p;$$

$$U@p \overset{def}{=} (link1_p,(1,2))\odot U@p + (link2_p,1)\uparrow U@p + (discovery_p,1)\downarrow U@p + (fail1_p,1)\downarrow U@p + (recovery1,1)\downarrow U@p + \sum_q (link1_{qp},1)\uparrow U@p + \sum_q (link2_{qp},1)\uparrow U@p;$$

$$I@p \overset{def}{=} (link2,(1,1))\odot I@p + (discovery_p,1)\uparrow I@p + (fail2_p,1)\downarrow I@p + (recovery2_p,1)\downarrow I@p;$$

$$F@p \overset{def}{=} (fail1_p,1)\uparrow F@p + (fail2_p,1)\uparrow F@p + (recovery3_p,1)\downarrow F@p;$$

$$R@p \overset{def}{=} (recovery1_p,1)\uparrow R@p + (recovery2_p,1)\uparrow R@p + (recovery3_p,1)\uparrow R@p + (insecure_p,1)\downarrow R@p + \sum_q (m2_{pq,R[p\to q]},(1,1))\downarrow H@p + \sum_q (m2_{qp,R[q\to p]},(1,1))\uparrow H@p;$$

In the model, while there are no connection or migration between two clusters, let the corresponding coefficient to be zero. Then in location *k*, the initial number of species A is <, vulnerability diffusion driven by a certain initial vulnerabilities can be modeled as

$$H[N^{v}_{W1}] \triangleright\!\!\triangleleft_{*} U[N^{v}_{E1}] \triangleright\!\!\triangleleft_{*} I[N^{v}_{S1}] \triangleright\!\!\triangleleft_{*} F[N^{v}_{F1}] \triangleright\!\!\triangleleft_{*} R[N^{0}_{R1}] \triangleright\!\!\triangleleft_{*} H[N^{0}_{W2}] \triangleright\!\!\triangleleft_{*} U[N^{0}_{E2}] \triangleright\!\!\triangleleft_{*} I[N^{0}_{S2}] \triangleright\!\!\triangleleft_{*} F[N^{0}_{F2}] \triangleright\!\!\triangleleft_{*} R[N^{0}_{R2}] \triangleright\!\!\triangleleft_{*} \ldots \triangleright\!\!\triangleleft_{*} H[N^{0}_{WN}] \triangleright\!\!\triangleleft_{*} U[N^{0}_{EN}] \triangleright\!\!\triangleleft_{*} I[N^{0}_{SN}] \triangleright\!\!\triangleleft F[N^{0}_{FN}] \triangleright\!\!\triangleleft R[N^{0}_{RN}]$$

A case study

For comparison with VDS, the total number of

nodes is still set to be 300, each cluster has 100 components, and the connection probability of connection within cluster is also same. And assuming in the initial time $L1$ has the component of U and I, and other locations only have the component of H, which means that only $L1$ is vulnerability. The initial number of component is =20, =10, =70, = =100ÿand the remaining value is 0.

(1) Analysis of VDM

Firstly, for comparison with VDS, considering a weak recoverability of WSN, the rest parameters' values ars as the same as that of Figure 1. The population of component U and I is shown in Figure 5(a) and (b) separately while *link_rate* = 1/100 and *link_rate* = 1/10. Seen from the figures, the time used to reach the maximum number of both U and I at location $L1$ is always earlier that that of at $L2$. Likewise, when we enlarge *link_rate*, the time needed to reach a peak become shorter in Figure 5(b), and because of the limit of paper length, we don't give extra figures. This phenomenon indicates that enhancing the *link_rate* will accelerate the diffusion, in other words, reducing the connection probability will decrease the vulnerability diffusion.

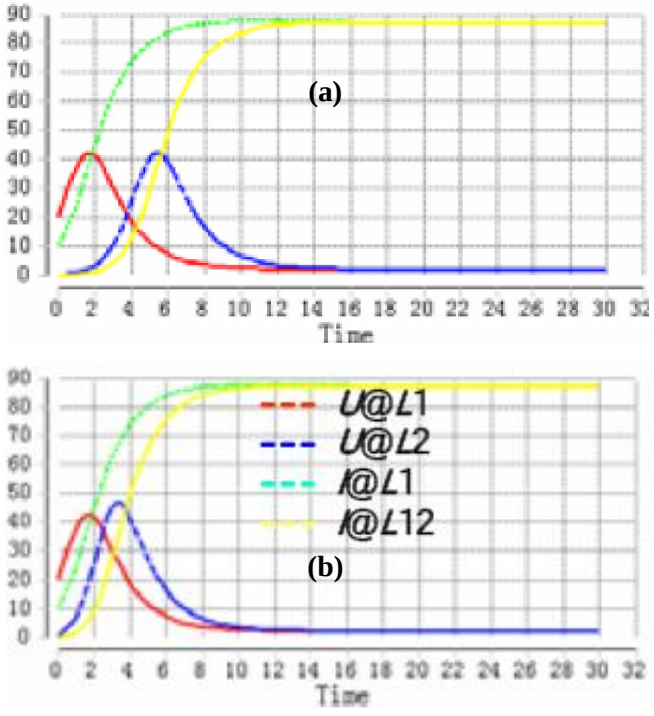

Figure 5 : The population of U and I in a cluster while (a) *link_rate*=1/100 and (b) *link_rate*=1/10

Figure 6 shows the population of U^* and I^* which includes all the components of U and I in all three clusters, while *link_rate*=1/10. Seen from the Figure 6, VDM needs more time to arrive at the same level than VDS. But when reach a steady state, the population of each species in multi-cluster is almost as same as that of a single cluster. So while the recoverability is weak, there is not much difference in diffusion scale between VDM and VDS.

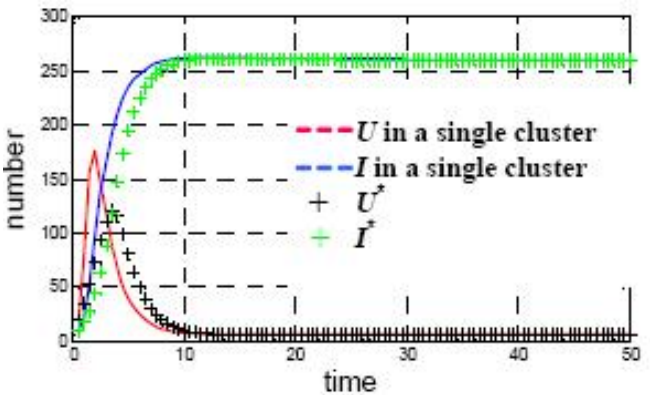

Figure 6 : The comparison between VDS (fixed line) and VDM ('+' line) while having a weak recoverability

Secondly, while every cluster has a strong recoverability, we still set *link_rate*=1/10. At this scene, the population of U and I in $L2$ is shown in Figure 7. Comparing with Figure 5(b), the population of both U and I are significantly decrease. Furthermore, the results of Figure 8 show the comparison of the population of U^* (or I^*) in multi-cluster with a single cluster, and we can find that the number of U and I in VDM are significantly less than that of VDS when reach a steady state at last. The phenomenon tells us a fact that dividing WSN system into multi-cluster is helpful to limit the diffusion of vulnerability while the clusters have strong recover-

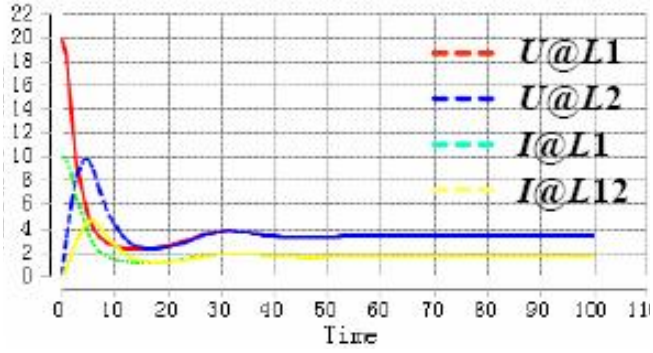

Figure 7 : The population of U and I in $L2$ while having a strong recoverability

FULL PAPER

ability. The law behind the phenomenon is also easy to understand, that is, the probability of repairing weak points in clusters become larger with the increase of recoverability. It means that vulnerability spread to other clusters with a lower probability, slowing down vulnerability diffusion indirectly.

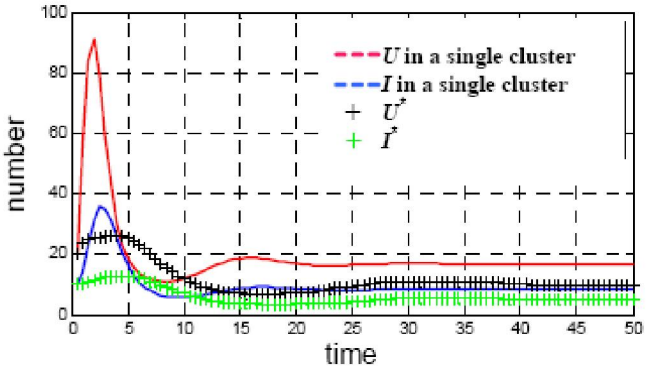

Figure 8 : The comparison between VDS (fixed line) and VDM ('+' line) while having a strong recoverability

For the broadly existence of recovery methods such as software upgrading, patching and antivirus, the approach of suppressing vulnerability diffusion in multi-cluster is easily exploited, which has the same effect with dividing into logic sub-network used in practice.

(2) Analysis of VDMM

At first, the clusters are set to be of a weak recoverability, and other parameters are as the same as that in Figure 5(b). In Figure 9(a) and (b), the population of *U* and *I* is shown separately while *migration_rate*= 1/100 and 1/2. Seen from the figure, with the increase of *migration_rate*, the diffusion of vulnerability accelerates. And when *migration_rate* increase to 1/2, the population of *U* (or *I*) in *L2* are essentially coincident with that of *L1*, which indicates that migration of nodes accelerates the diffusion of vulnerability.

Next, while the system has a strong recoverability, the population of *U* and *I* is shown in Figure 10(a) and (b) separately while *migration_rate*=1/100 and 1/2 to compare with Figure 9. The trends also tell us the same conclusion that the increase of *migration_rate* accelerates the diffusion of vulnerability. For example, as the population of *U* and *I* arrive at the same level in all the three clusters, the cost of time is 12 units (seconds) while *migration_rate* =1/100, but less than 5 units (seconds) while *migration_rate* =1/2

Figure 11 shows the impacts from nodes migration

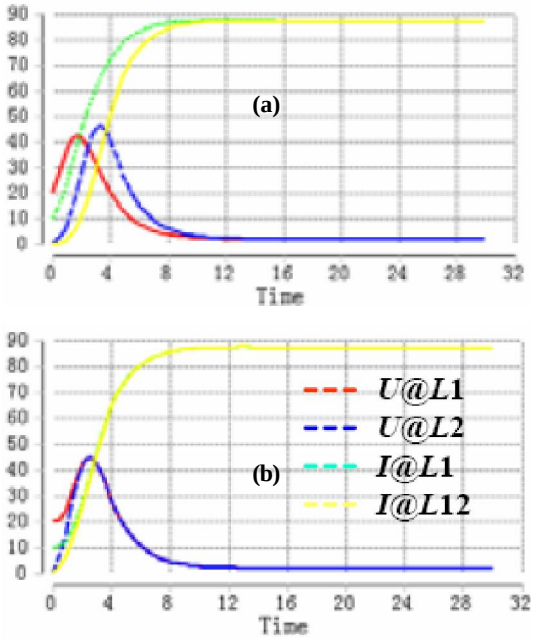

Figure 9 : The population of *U* and *I* VDMM has a weak recoverability with (a) *migration_rate*=1/100 and (b)

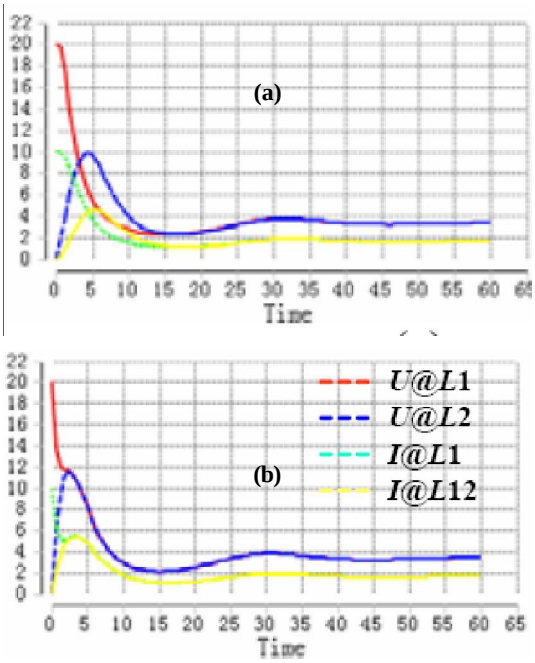

Figure 10 : The population of *U* and *I* while VDMM has a strong recoverability with (a) *migration_rate*=1/100 and (b) *migration_rate*=1/2 seperately

on the vulnerability diffusion when WSN is capable of strong recoverability. In Figure 11(a), the population of *U* and *I* in *L2* is shown, and the extreme values of curves tell us that the maximum number of *U* and *I* is increased for VDMM. In other words, the migration of nodes expands the maximum scale of vulnerability. And in Fig-

ure 11 (b), the population of U^* and I^* in the whole system indicate a same result. But we also need to note that the final scale of vulnerability diffusion is not different between VDM and VDMM after arriving a steady state at last, while the strong recoverability works.

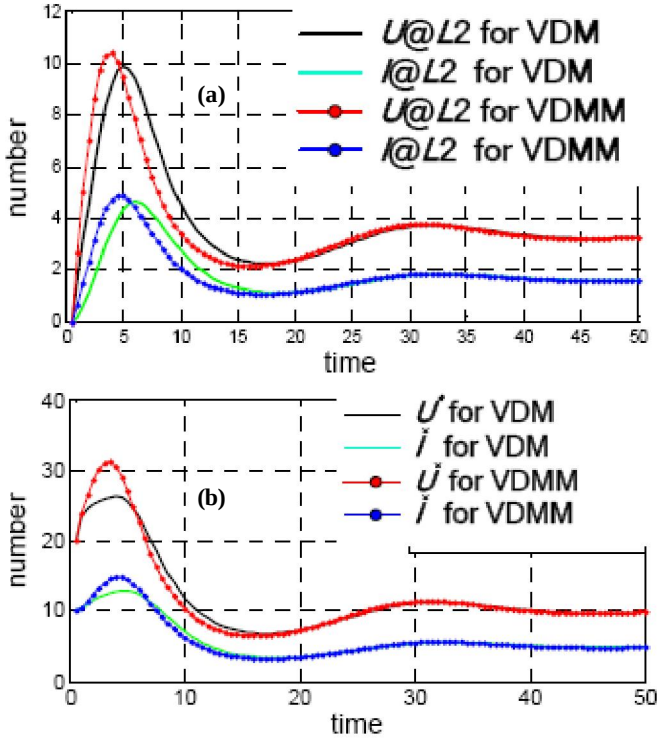

Figure 11 : (a) the population of *U* and *I* in a cluster, and (b) the population of U^* and I^* in the whole system for both VDM and VDMM

It must be noted that for a certain parameter, in the most scenes only two or three values are given to explain the impacts from it on vulnerability diffusion, but in fact we use more different values to verify the trend, and we don't give extra figures just because of the limit of paper length.

Through the analysis of the above three cases, it is not difficult to find that our model can simulate the trend of vulnerability diffusion for WSN with multi-cluster reasonably, generally consistent with reality. And the following conclusions can be drawn.

1) Dividing WSN system into multi-cluster is helpful to limit the diffusion of vulnerability while the clusters have strong recoverability.
2) Reducing the connection between clusters will decrease the vulnerability diffusion, and the less connections between clusters, the more apparent this phenomenon.
3) The speed of vulnerability diffusion will be accelerated as nodes migrating between clusters freely, and they are a positive relationship.

RELATED WORKS

And many researches has been done for analysis of vulnerability. At present, more and more studies show that vulnerabilities brought out by connections between nodes are even more than inherent ones. And the kind of vulnerabilities caused by connections grows explosively and becomes the majority.

Many scholars have devoted their attentions to research the law of vulnerability evolution and vulnerability propagation to predict vulnerability diffusion and provide the basis for recovery. A. Ozment[9] from University of Cambridge analyzed the history data of OpenBSD in the past 8 years to study the evolution of vulnerability. S. Neuhaus[10] summarized and classified the vulnerabilities of Mozilla to predict outbreak of security events. For diffusion of data error, Hiller[3] discussed diffusion behaviors between associated software modules, and proposed a method of setting checkpoint and recovery point. In the field of WSN, De[2] studied the vulnerability diffusion problems of multi-hop broadcast protocol, and discussed the impacts from sojourn time, connectivity, recoverability with spy software, but he don't take notice of the characteristic of heterogeneous in the process of vulnerability diffusion.

But the vulnerability diffusion between clusters in WSN is not homogeneous, because the probability of connection between nodes within or without a cluster is usually different, and the mobile nodes also accelerate the spread of vulnerability diffusion. So that we build a vulnerability diffusion model to simulate the characteristic of heterogeneous for vulnerability diffusion. Beside, when the system contains thousands of nodes, traditional CTMC or DTMC methods is hard to solve because of a very big state space. To overcome the shortcoming of state space explosion, we use Bio-PEPA to convert into ODEs in this paper.

CONCLUSION

A vulnerability diffusion model for WSN by Bio-PEPA is proposed inspired by epidemiological model

in this paper. With respect to the characteristic of heterogeneous in the process of vulnerability diffusion, we take the multi-cluster, recoverability and migration of nodes into count to model vulnerability diffusion in three steps. At first, a VDS model is built to describe vulnerability diffusion in a single cluster just as traditional model, which is a basis for comparison. Then the model of VDM and VDMM are separately built to analyze vulnerability diffusion with or without migration of nodes, compared with VDS. The experiment results shown that enhancing recoverability of cluster, decreasing the connection rate between cluster and preventing the migration of node is useful to reduce vulnerability of WSN. It is needed to point that though there are only three clusters in the cases, the model proposed is also suitable for a finite number of clusters. In addition, our method can be used in large-scale systems without a problem of state space explosion.

In further study, we hope to build a vulnerability diffusion model with nodes added randomly.

ACKNOWLEDGEMENTS

This work is supported by the National Natural Science Foundation of China under Grant Nos. 61370212; Special Fund for Basic Scientific Research of Central Colleges under Grant Nos. HEUCFZ1213, HEUCF100601; the Research Fund for the Doctoral Program of Higher Education of China 20122304130002. the Natural Science Foundation of Heilongjang Province under Grant No. ZD 201102, F201217.

REFERENCES

[1] T.Kavitha, D.Sridharan; Security Vulnerabilities In Wireless Sensor Networks: A Survey. Journal of Information Assurance and Security, **5**:, 31-44 **(2010)**.

[2] P.De, Y.Liu, S.Das; An Epidemic Theoretic Framework for Vulnerability Analysis of Broadcast Protocols in Wireless Sensor Networks. IEEE Transactions on Mobile Computing, **8(3)**:, 413-425 **(2009)**.

[3] M.Hiller, A.Jhumka, N.Suri; EPIC: Profiling the Propagation and Effect of Data Errors in Software. IEEE Transactions on Computers, **53(5)**:, 1-19 **(2004)**.

[4] Orhan Gemikonakli, Enver Ever, Altan Kocyigit; Approximate solution for two stage open networks with Markov-modulated queues minimizing the state space explosion problem. Journal of Computational and Applied Mathematics, **223(1)**:, 519–533 **(2009)**.

[5] F.Ciocchett, J.Hillston; Bio-PEPA: a framework for the modelling and analysis of biological systems. Theoretical Computer Science, **410(33, 34)**:, 3065-3084 **(2009)**.

[6] Federica Ciocchetta, Jane Hillston; Bio-PEPA: An Extension of the Process Algebra PEPA for Biochemical Networks. Electronic Notes in Theoretical Computer Science, **194(3)**:, 103–117 **(2008)**.

[7] Federica Ciocchettaa, Jane Hillston; Bio-PEPA for Epidemiological Models. Electronic Notes in Theoretical Computer Science, **261**, 43–69 **(2010)**.

[8] F Ciocchetta, A Duguid, S.Gilmore; The Bio-PEPA Tool Suite. The Sixth International Conference on the Quantitative Evaluation of Systems. Eger, Hungary, Sept. 13-16, 309-310 **(2009)**.

[9] A.Ozment; Vulnerability Discovery & Software Security. London:University of Cambridge[D], **(2007)**.

[10] S Neuhaus, T Zimmermann, C.Holler, et al.; Predicting Vulnerable Software Components. Conference on Computer and Communications Security (CCS'07), Alexandria, USA, October 29-November 2, 529-540 **(2007)**.