计算机研究与发展
Journal of Computer Research and Development
DOI:10.7544/issn1000-1239.2018.20170305
55(11):2467-2481,2018

目标成本值最优的物联网 WSS 蠕虫抑制算法

黄一才　周伟伟　郁　滨
(解放军信息工程大学　郑州　450001)
(huangyicai3698@163.com)

Optimal Suppression Algorithm Against Worm Propagation in Wireless Service System for IoT Based on Target Cost Function

Huang Yicai, Zhou Weiwei, and Yu Bin
(*PLA Information Engineering University*, *Zhengzhou* 450001)

Abstract　With the adoption of the general standard of communication protocol, wireless service system (WSS) in IoT is proposed to achieve the real-time connection between person and things or things and things. According to the characteristics of IEEE 802.15.4, wireless sensor nodes are embedded in the devices for data collection and command broadcasting. However, the isomorphism of the sensor nodes makes the worm propagation an increasingly serious problem. Firstly, based on the classification of epidemiological models related to worm propagation and the analysis of the characteristics of various models, an epidemiological model is constructed, which specially introduces sleep state and quarantine state into state transition. The transition relationship of nodes is defined simultaneously. Secondly, according to the radio frequency, the number and range of infected nodes with actual transmission ability are determined. Thirdly, we introduce the target cost function between worm and wireless service system, and put forward a dynamic differential game with complete information based on the overall damage. Then, the existence of saddle-point solution is proved, which is solved by combining state parameters, cooperative state variables and Hamiltonian functions. The optimal defense algorithm is proposed to minimize target cost function. Finally, different algorithms are implemented and the performance evaluation is carried out by comparing the characteristics of nodes in each state and the corresponding overall damage. The experimental results show that the optimal defense algorithm based on improved epidemic model can suppress worm propagation in wireless service system effectively and efficiently.

Key words　Internet of thing (IoT); wireless service system (WSS); worm propagation; differential game; Hamilton function

摘　要　物联网无线服务系统(wireless service system, WSS)是以通用的协议标准实现人与物、物与物相连的实时网络交互系统.该系统在设备中嵌入无线传感器节点以实现数据上传和决策下发,但传感器节点的同构性特点使得蠕虫传播问题日益严重.为此,在对现有蠕虫传播的流行病模型进行分类并总结

收稿日期:2017-05-03;**修回日期**:2018-01-18
基金项目:河南省科技攻关项目(132102210003);信息保障技术重点实验室开放基金项目(KJ-15-104)
This work was supported by the Project of Henan Provincial Key Scientific and Technological Research (132102210003) and the Open Fund of Key Laboratory for Information Security (KJ-15-104).
通信作者:周伟伟(zww15238060801@163.com)

各类模型特点的基础上,首先提出了具有睡眠状态和隔离状态的流行病模型,定义了系统中节点的状态转换关系;其次,依据节点的射频通信距离,确定了具有实际传染能力的感染节点数量及范围;再次,引入蠕虫与无线服务系统的目标成本函数,给出了基于目标成本值的完全信息动态微分博弈模型;然后,证明了该博弈存在鞍点策略,利用状态变量、协状态变量和汉密尔顿函数求解鞍点策略并设计了保证目标成本值最优的防御策略算法;最后,仿真实现本算法与 2 种蠕虫防御策略算法,通过各状态节点的变化特点及目标成本值的对比实验进行性能评估. 实验结果表明:基于改进流行病模型的最优防御算法在抑制无线服务系统蠕虫传播方面有明显优势.

关键词 物联网;无线服务系统;蠕虫传播;微分博弈;汉密尔顿函数

中图法分类号 TP309.1; TP309.5

作为互联网的发展与延伸,物联网技术将广泛应用于社会的方方面面[1-3]. 依据数据采集及传输方式,物联网可分为有线服务系统和无线服务系统(wireless service system, WSS). 物联网 WSS 的实质是将最新的信息处理与通信技术进行有效的整合,利用各种感知设备和智能物体采集语音、图像、位置等数据信息,并通过无线通信方式将采集到的信息传递到资源平台进行云计算、数据挖掘等操作后,将处理与控制结果呈现给各用户终端[4-6]. 然而,由于 WSS 需要将大量感知设备和物体连接到应用系统中,而且这些设备和物品信息大多采用资源受限的无线传感器网络节点进行传输,这在增加系统灵活性和方便性的同时,为蠕虫在网络中的大规模传播提供了便利条件[7]. 蠕虫可以窃取被感染节点的隐私数据和敏感信息,同时伪造身份向邻居节点发送包含恶意代码的无线再编程指令,使感染蠕虫的节点数量不断增加. 被感染节点可能导致系统大面积数据阻塞甚至瘫痪. 感知设备中嵌入的大多数无线传感器节点具有资源有限的特性,如何最大限度地防御蠕虫在系统中的传播成为物联网快速发展的关键. 目前,针对无线网络中蠕虫的研究主要是利用传统的流行病模型构建传播体系[8],在此基础上,分析不同网络参数对传播速率的影响[9].

许多分布式网络中的蠕虫传播及抑制方案这些年被相继提出以保护采集数据的安全性和隐私性. 有线网络中经典的 Kermack-Mckendrick 参考模型将节点划分为 3 种不同的状态,利用控制参数和动力学微分方程描述蠕虫的传播速率,但仅将节点状态假设为易感状态,感染状态和恢复状态与 WSS 中 IEEE 802.15.4 协议不符[10]. 基于 Markov 链的传播模型通过引入节点半径与信道扫描的速率等参量求解各状态变量的微分博弈方程,但该方法未考虑 WSS 中基站控制节点休眠和唤醒转换概率的情况[11]. 基于细胞自动机的传播模型对多跳广播协议中的蠕虫病毒传播特点进行了模拟,但并未给出具体的防御和抑制方案[12]. Bradley 等人[13]提出了一种基于进程代数的复合主动蠕虫传播模型,通过 PEPA 流近似方法求解各状态的常微分方程. 但该模型需要消耗大量系统资源,并不适用于节点资源受限的 WSS. 基于分层结构的异构网络蠕虫传播模型 Enhanced-AAWP 利用本地优先扫描蠕虫和随机扫描蠕虫等方法研究蠕虫的渗透传播过程[14]. 由于传统蠕虫传播模型中未考虑设备节点失效和休眠状态,因此,现有传播机制的分类并不适用于 WSS.

高效的蠕虫分类机制是提高蠕虫传播模型准确性和防御策略效率的重要因素. 为解决现有蠕虫分类准确性差的问题,朱克楠等人[15]基于朴素贝叶斯机器学习提出了利用行为序列报告和操作相似度窗口对蠕虫进行自动分类的方法. 该方法为 WSS 蠕虫传播及抑制模型的进一步研究提供了有力支撑.

上述方案仅利用各种传播模型对蠕虫在不同网络环境下的传播特点和分类方法进行了分析,依据动力学原理获取不同节点的数据集,并未研究如何基于传播规律建立动态防御策略集抑制蠕虫在 WSS 中的传播.

快速准确的蠕虫检测方法是实现 WSS 动态最优策略防御的基础. 近年来,一些学者通过降低噪声和网络随机事件的干扰,提出了许多基于行为序列的检测算法. 毛蔚轩等人[16]利用节点休眠状态和基于人工经验的自适应学习方法,在分析网络拓扑中各层节点数据流依赖关系的基础上,给出了一种进程调用行为相似性与异常度结合的蠕虫抑制算法. 该算法保证在已知少量蠕虫样本的条件下得到理想的检测率. 文献[17]提出基于平均场法建立免疫模型,但该模型的抑制策略中没有休眠模式. 针对混淆蠕虫攻击难以检测的问题,马洪亮等人[18]提出利用

遍历抽象语法树(abstract syntax tree, AST)和单等级支持向量机(one class support vector machine, OC-SVM)检测混淆并跟踪节点相关的变量终值检测反混淆. Das 等人[19]用频繁项离群确定同源度,并对同源判定阈值进行预定义,进而实现蠕虫的同源判定. 郭洋河等人[20]将数据和行为依赖关系相结合,提出了一种基于造价序列库的蠕虫识别机制. 文献[21-22]利用挂钩 API 的方法追踪外部节点的访问行为,改善了动态检测方法执行路径单一的问题. 这些检测方法对蠕虫的检测成功率较高,但缺少与 WSS 匹配的蠕虫最优防御算法.

在现有的蠕虫检测策略条件下,如何设计最优防御算法是 WSS 在高安全领域应用的关键. 文献[23]指出可以将蠕虫防御问题转化为动态博弈纳什均衡的求解问题. Mishra 等人[24]引入攻击方的类型概率和收益矩阵,通过判断贝叶斯博弈中收益函数值选取最优动作集合,但该方案不能实时校正外部节点的后验概率. Pandey 等人[25]将蠕虫和系统的攻击防御转化为二者的最优化博弈问题,提出基于僵尸网络环境的流行病模型以及微分博弈机制,在考虑防御策略及恶意软件参数限制的基础上给出了动态博弈的闭环纳什均衡解. 该方案虽然得到了微分博弈最优解,但未讨论隔离状态对其他网络状态的影响. 这些方案虽然对蠕虫在无线网络中的传播及抑制进行了一定程度的研究,但并未考虑 WSS 中基站能够使易感节点免疫,同时可以广播补丁程序治愈已感染节点.

结合 WSS 中的网络参数及节点状态特点,仍需重新建立传播模型及基于时间特性的动态微分博弈模型,在满足网络参数约束条件的情况下,有效抑制蠕虫在网络中的传播并使系统的安全成本最低.

本文的贡献是构建一个描述物联网 WSS 蠕虫传播规律的流行病模型和蠕虫防御动态微分博弈模型,求解最优策略控制集合的取值并给出了抑制蠕虫传播的防御算法. 具体地,本文主要得到 3 个方面的结果:

1) 依据物联网中 WSS 的网络拓扑结构,对蠕虫攻击下设备节点之间的状态转换过程进行分析,给出了 WSS 状态转换图和各状态微分方程,利用通信半径确定具有感染能力的节点区域和数量并构建改进的流行病模型.

2) 建立蠕虫和 WSS 间的动态微分博弈模型,通过目标成本函数量化攻击方和防御方不同决策所得到的成本支付.

3) 利用汉密尔顿函数、瞬时成本函数以及终期成本函数的线性条件证明了动态微分博弈鞍点的存在性. 结合成本函数与汉密尔顿函数的最小化极值特点,求解蠕虫传播的最优防御策略并给出了具体算法.

1 基础知识

1.1 WSS 系统结构

随着物联网在各个领域的快速发展,其安全问题越来越受到重视,尤其是对信息安全性和隐私性要求较高的 WSS 中. WSS 大多采用无线传感器网络技术实现各节点之间的数据与图像传输、数据处理、实时控制等. 无线信道的开放性和节点的同构性使恶意软件很容易在网络中大面积传播.

物联网 WSS 的系统结构如图 1 所示. 该系统的目标是保证无论何时、何地、任何媒体设备的实时连通性和可控性. 其结构主要包括无线传输系统、数据采集系统和终端用户系统. 由于 WSS 设备的移动性和采集数据分布范围广的特点,无线传感器网络技术在 WSS 数据采集和传输中的应用较为普遍. 但嵌入无线传感器网络节点的设备一旦被蠕虫感染,极易在网络中传播. 这些被感染的节点对系统数据的机密性、隐私性和完整性构成了巨大威胁.

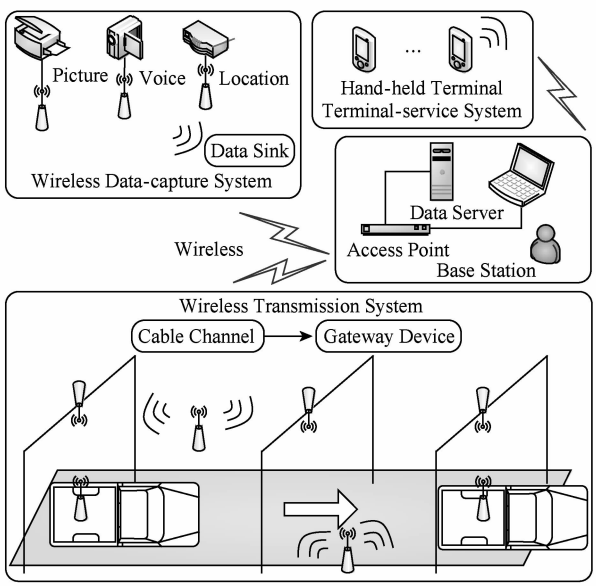


Fig. 1 System structure of WSS in IoT

图 1 物联网 WSS 的系统结构

1.2 网络链路模型

WSS 中各节点链路可模型化为由 N 个节点组成的无向图. 设 $p(d_{ab})$ 为节点 a 和 b 之间存在一条

无向边的概率,则网络中各 WSS 节点之间存在的边的总数为 $E=\sum_{a=1}^{N}\sum_{b=a+1}^{N}p(d_{ab})$. 令 ξ 为节点的平均度数,即每个节点持有的平均链接数为

$$\xi=\frac{2E}{N}. \tag{1}$$

节点间存在链接的概率 $p(d_{ab})$ 由 WSS 中的无线电通信机制以及节点间的通信半径决定. 信号发射端与信号接收端通过阈值判定模型确定节点间是否存在通信链路[26].

1.3 传统流行病模型

流行病模型最早是用于预测和描述具有传染性因素特征的部分人员集合随时间实时变化的态势. 这些因素特征包括网络节点的发送率、接收率、传播速率等. 依据不同网络的特点,现有流行病模型分为 SI[27],SIS[28],SIR[29] 模型.

通常情况下,网络中感染节点注入安全补丁可以转变为恢复状态. 因此,SIR 模型更适用于具有安全补丁的网络流行病传播特点. $S(t)$,$I(t)$,$R(t)$ 分别代表在时刻 t 易感节点、感染节点和免疫节点的数量. 假设网络中各节点之间存在通信链路,令 α 为蠕虫感染速率,β 为免疫速率,则描述各状态节点变化速度的微分方程为

$$\frac{\mathrm{d}S(t)}{\mathrm{d}t}=-\alpha S(t)I(t)-\beta S(t),$$

$$\frac{\mathrm{d}I(t)}{\mathrm{d}t}=\alpha S(t)I(t)-\beta I(t),$$

$$\frac{\mathrm{d}R(t)}{\mathrm{d}t}=\beta(S(t)+I(t)).$$

分析其他网络模型时,可以依据经典 SIR 模型和具体的网络协议进行分析和求解.

1.4 蠕虫攻击模型

WSS 中传播的蠕虫主要指被动型蠕虫. 假设 WSS 中源节点的安全认证机制失效,蠕虫控制该节点并获取认证密钥. 感染节点利用签名密钥对嵌入蠕虫字段的再编程命令帧签名并发送到邻居节点. 其他邻居节点收到伪装的命令帧后验证认证机制,更新配置并执行相应操作. 蠕虫在 WSS 节点间通过多跳的形式向外传播. 感染节点采集和传输的数据遭受窃听、篡改和破坏.

2 WSS 系统模型

2.1 状态转换

WSS 中的蠕虫传播可以用包含易感节点、感染节点和免疫节点的区域来表示. 不同状态节点的分布如图 2 所示. 由于免疫节点在全网均匀分布,因此,未在系统模型中标出.

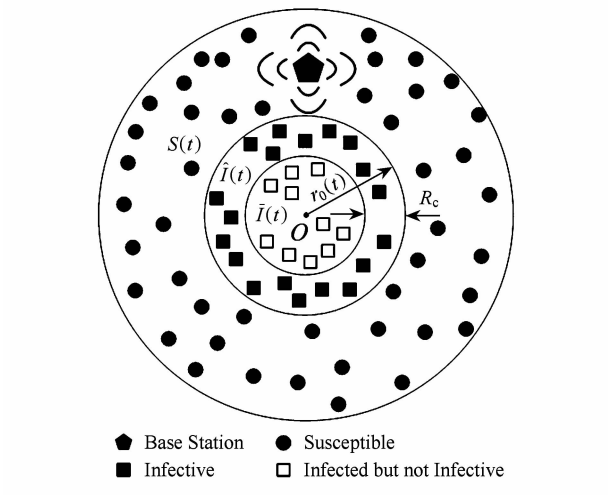


Fig. 2 System model of WSS

图 2 WSS 系统模型

假设 WSS 中的节点个数为 N,各区域节点分布密度为 σ. 若整个网络节点分布不均匀时通过转换可等效为统一的分布密度. 在时刻 t,易感节点的数量为 $S(t)$. 同样地,$I(t)$,$R(t)$ 分别表示感染节点和免疫节点的数量. $r_0(t)$ 为时刻 t 感染节点的区域半径,各节点的通信半径是 R_c. 在感染节点区域内,由于 $\bar{I}(t)$ 中节点与易感节点的通信距离超过节点的通信半径 R_c,这部分感染节点无法感染外侧的易感节点. 与易感节点的距离小于半径 R_c 的感染节点数量定义为 $\hat{I}(t)$. 基站通过向网络中发布安全补丁使部分易感节点和感染节点恢复. 节点数量满足关系:

$$I(t)=\bar{I}(t)+\hat{I}(t).$$

依据 WSS 的特点,节点可分为 8 种不同的状态,各状态之间的转换关系如图 3 所示:

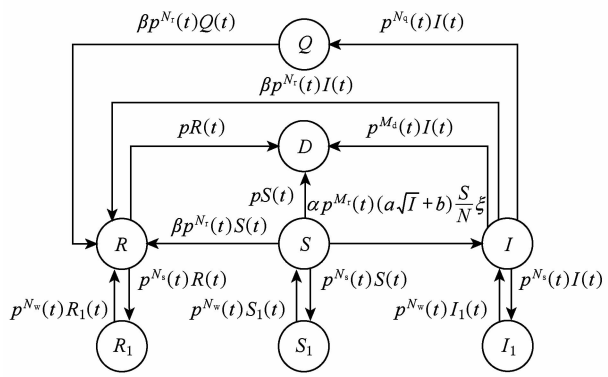


Fig. 3 State transition diagram of nodes in WSS

图 3 WSS 节点状态转换图

1) S. 处于状态 S 的节点为易感节点,即易遭受蠕虫感染,但目前仍处于正常工作状态.

2) S_1. 处于状态 S_1 的节点是状态 S 的休眠节点,由于节点处于休眠状态时停止收发数据,该状态的节点不会被蠕虫感染.

3) I. 处于状态 I 的节点为感染节点,蠕虫可以窃取节点存储、处理、传输的数据,同时能通过该节点感染其他易感节点.

4) I_1. 处于状态 I_1 的节点为感染节点 I 的休眠节点,该状态的节点失去通信功能,无法感染其他易感节点.

5) Q. 处于状态 Q 的节点为系统通过安全规则检测并隔离的节点,该状态的节点无法感染其他易感节点.

6) R. 处于状态 R 的节点为植入安全补丁后获得对蠕虫免疫状态的节点.

7) R_1. 处于状态 R_1 的节点为状态 R 的休眠节点.

8) D. 处于状态 D 的节点为失效节点,节点失去通信和数据处理功能.

假设 α 为节点由状态 S 转化 I 的相关系数,且与每个感染节点发送数据帧的最大速率有关,时刻 t 感染节点实际发送速率与最大速率的比值为 $p^{M_r}(t)$. 节点由状态 I 转化为 D 的概率为 $p^{M_d}(t)$. 节点由易感状态 S 转变为休眠状态 S_1 的概率为 $p^{N_s}(t)$,休眠状态 S_1 被唤醒的概率为 $p^{N_w}(t)$. 因此,状态 I 与 I_1 以及 R 与 R_1 相互转换的概率分别为 $p^{N_s}(t)$和 $p^{N_w}(t)$. 状态 I,Q,S 的节点转化为 R 是由基站发送的安全补丁更新完成的. 时刻 t 基站发送安全补丁的实际速率与最大速率的比值为 $p^{N_r}(t)$. β 为 S 和 I 转化 R 的相关系数,且与基站发送安全补丁的最大(通信)速率有关. 节点从状态 I 转换为状态 Q 的概率 $p^{N_q}(t)$是由 WSS 中的安全策略控制的. 节点由状态 R 和 S 转换为状态 D 的概率为常数 p.

在以上参数中,$p^{M_r}(t)$和 $p^{M_d}(t)$由蠕虫控制,且 $0\leqslant p^{M_r}(t)\leqslant 1$,$0\leqslant p^{M_d}(t)\leqslant 1$. $p^{N_s}(t)$,$p^{N_w}(t)$,$p^{N_r}(t)$,$p^{N_q}(t)$由 WSS 通过基站广播命令帧并配合蠕虫检测安全机制进行控制,且 $0\leqslant p^{N_s}(t)\leqslant p_{\max}^{N_s}$,$0\leqslant p^{N_w}(t)\leqslant 1$,$0\leqslant p^{N_r}(t)\leqslant 1$,$0\leqslant p^{N_q}(t)\leqslant p_{\max}^{N_q}$. 其中,$p_{\max}^{N_s}$ 为节点由工作状态转换为休眠状态的最大概率,$p_{\max}^{N_q}$ 与 WSS 协议中安全机制的性能有关.

8 种不同状态的节点满足:

$$N=S(t)+I(t)+R(t)+S_1(t)+I_1(t)+R_1(t)+D(t)+Q(t). \tag{2}$$

在图 2 中,当系统中仅有感染节点和易感节点时,则有 $I(t)=\sigma\pi r_0^2(t)$. 厚度为 R_c 的环形区域内的感染节点数量为 $\hat{I}(t)=I(t)-\sigma\pi(r_0(t)-R_c)^2$. 由式(1)可知,仅有 $\frac{N-I}{N}\xi$ 个邻居节点为易感节点. 由此可得,状态为 S 的节点转化为 I 的瞬时增量为

$$\alpha p^{M_r}(t)(2\sqrt{\sigma\pi}R_c\sqrt{I(t)}-\sigma\pi R_c^2)\frac{N-I}{N}\xi.$$

在图 3 中,假设初始条件为 $I(0)=m$,$R(0)=S_1(0)=I_1(0)=R_1(0)=D(0)=Q(0)=0$,则有 $S(0)=N-m$. 设时刻 t 状态为 S 的节点数量为 $S(t)$,则状态为 S_1 的节点瞬时增量为 $p^{N_s}(t)S(t)-p^{N_w}(t)S_1(t)$. 同理,状态为 I_1 的节点瞬时增量为 $p^{N_s}(t)I(t)-p^{N_w}(t)I_1(t)$,状态为 R_1 的节点瞬时增量为 $p^{N_s}(t)R(t)-p^{N_w}(t)R_1(t)$. 节点从状态 S 转换为 R 的瞬时增量为 $\beta p^{N_r}(t)S(t)$. 从状态 S 转换为 D 的瞬时增量为 $pS(t)$. 由假设条件可知,状态 I 转换为 R 和 D 的瞬时增量分别为 $\beta p^{N_r}(t)I(t)$和 $p^{M_d}(t)I(t)$,状态 I 转换为 Q 的瞬时增量为 $p^{N_q}(t)I(t)$,由 Q 到 R 的瞬时增量为 $\beta p^{N_r}(t)Q(t)$,状态 R 转换为 D 的瞬时增量为 $pR(t)$. 当 WSS 中同时存在这 8 种不同状态的节点时,状态为 I 的感染节点在以 $r_0(t)$为半径的圆形区域内的分布密度为 σ_1,由式(2)可得:

$$\sigma_1=\frac{1}{\pi r_0^2(t)}[N-(S(t)+R(t)+S_1(t)+I_1(t)+R_1(t)+D(t)+Q(t))]. \tag{3}$$

令 $a=2\sqrt{\sigma_1\pi}R_c$,$b=-\sigma_1\pi R_c^2$,则 S 转换为 I 的瞬时增量为 $\alpha p^{M_r}(t)(a\sqrt{I}+b)\frac{S}{N}\xi$. 利用以上分析并结合状态转换图,可得 8 个状态微分方程为

$$\begin{cases}\dfrac{dS}{dt}=p^{N_w}(t)S_1(t)-p^{N_s}(t)S(t)-\beta p^{N_r}(t)S(t)-\alpha p^{M_r}(t)(a\sqrt{I}+b)\dfrac{S}{N}\xi,\\ S(0)=N-m,\end{cases} \tag{4}$$

$$\begin{cases}\dfrac{dS_1}{dt}=p^{N_s}(t)S(t)-p^{N_w}(t)S_1(t),\\ S_1(0)=0,\end{cases} \tag{5}$$

$$\begin{cases}\dfrac{dI}{dt}=-(p^{M_d}(t)+p^{N_s}(t)+p^{N_q}(t)+\beta p^{N_r}(t))I(t)+p^{N_w}(t)I_1(t)+\alpha p^{M_r}(t)(a\sqrt{I}+b)\dfrac{S}{N}\xi,\\ I(0)=m,\end{cases} \tag{6}$$

$$\begin{cases}\dfrac{dI_1}{dt}=p^{N_s}(t)I(t)-p^{N_w}(t)I_1(t),\\ I_1(0)=0,\end{cases} \tag{7}$$

$$\begin{cases}\dfrac{\mathrm{d}R}{\mathrm{d}t}=\beta p^{N_r}(t)(S(t)+I(t)+Q(t))-\\ \quad pR(t)+p^{N_w}(t)R_1(t)-p^{N_s}(t)R(t),\\ R(0)=0,\end{cases}\tag{8}$$

$$\begin{cases}\dfrac{\mathrm{d}R_1}{\mathrm{d}t}=p^{N_s}(t)R(t)-p^{N_w}(t)R_1(t),\\ R_1(0)=0,\end{cases}\tag{9}$$

$$\begin{cases}\dfrac{\mathrm{d}D}{\mathrm{d}t}=pS(t)+pR(t)+p^{M_d}(t)I(t),\\ D(0)=0,\end{cases}\tag{10}$$

$$\begin{cases}\dfrac{\mathrm{d}Q}{\mathrm{d}t}=p^{N_q}(t)I(t)-\beta p^{N_r}(t)Q(t),\\ Q(0)=0.\end{cases}\tag{11}$$

2.2 博弈模型

假设蠕虫和 WSS 都是理性的,蠕虫与 WSS 构成连续时间上的微分博弈,则 WSS 中防御蠕虫传播的最优策略问题转化为求解微分博弈的最优解.其中,蠕虫通过参数 $p^{M_r}(t)$ 和 $p^{M_d}(t)$ 实现对传播特性的控制,WSS 通过参数 $p^{N_s}(t)$, $p^{N_w}(t)$, $p^{N_r}(t)$, $p^{N_q}(t)$ 实现对蠕虫的防御.

定义 1. WSS 蠕虫传播防御微分博弈. 给定一个博弈时间段 T,每个博弈回合所用的时间为 T_0, T_0 大小与不同通信协议的性能参数有关,则时间段 T 内经历的博弈阶段数量为 $k=\lceil T/T_0\rceil$. WSS 蠕虫传播防御微分博弈 K 是一个四元组 $Z(N,B,F,J)$,其中:

1) $N=\{$蠕虫 A, WSS $D\}$是一个包含 2 个参与者的集合;

2) $B=M\times S$,其中,$M=\{m(t)\mid m(t)=(p^{M_r}(t),p^{M_d}(t)),0\leqslant p^{M_r}(t)\leqslant 1,0\leqslant p^{M_d}(t)\leqslant 1\}$是蠕虫的策略集, $S=\{s(t)\mid s(t)=(p^{N_s}(t),p^{N_w}(t),p^{N_r}(t),p^{N_q}(t)),0\leqslant p^{N_s}(t)\leqslant p_{\max}^{N_s},0\leqslant p^{N_r}(t)\leqslant 1,0\leqslant p^{N_q}(t)\leqslant p_{\max}^{N_q}\}$是 WSS 的策略集合,第 l 个博弈阶段攻防双方的策略为$(m(lT_0),s(lT_0))$;

3) $F=\{f(t,r(t),m(t),s(t))\mid f_S,f_I,f_R,f_{S_1},f_{I_1},f_{R_1},f_Q,f_D\}$,其中 $f(t,r(t),m(t),s(t))$是一个随时间连续变化的状态函数,$r(t)=(S(t),I(t),R(t),S_1(t),I_1(t),R_1(t),Q(t),D(t))$是一个 8 维的状态变量,其中,

$$f_S=\frac{\mathrm{d}S}{\mathrm{d}t},f_I=\frac{\mathrm{d}I}{\mathrm{d}t},f_R=\frac{\mathrm{d}R}{\mathrm{d}t},f_{S_1}=\frac{\mathrm{d}S_1}{\mathrm{d}t},$$

$$f_{I_1}=\frac{\mathrm{d}I_1}{\mathrm{d}t},f_{R_1}=\frac{\mathrm{d}R_1}{\mathrm{d}t},f_Q=\frac{\mathrm{d}Q}{\mathrm{d}t},f_D=\frac{\mathrm{d}D}{\mathrm{d}t};$$

4) $J(m(t),s(t))=\int_{(l-1)T_0}^{lT_0}w(t,r(t),m(t),s(t))\mathrm{d}t+u(r(lT_0))-u(r((l-1)T_0))$是第 l 个博弈阶段与 WSS 的 QoS 相关的成本函数,其中,$w(t,r(t),m(t),s(t))\mathrm{d}t$ 表示在很小的一段时间 Δt 内各状态影响网络 QoS 所造成的瞬时成本,$u(r(lT_0))-u(r((l-1)T_0))$为第 l 个博弈阶段结束时重新部署节点替代失效节点所付出的终期成本.

在定义 1 中,博弈双方分别为蠕虫 A 和 WSS D,WSS 通过调整防御策略 S 使目标成本函数 J 的值达到最小,而蠕虫通过攻击策略 M 保证 J 的值达到最大,该博弈模型为动态完全信息零和微分博弈,即攻击方和防御方均知道对方采取每个策略的支付.

在蠕虫的控制策略集合$(p^{M_r}(t),p^{M_d}(t))$中,$p^{M_r}(t)$可以控制时刻 t 感染节点发送包含蠕虫字段的命令帧的速率,该值的大小决定了易感节点被感染的速率,攻击方在获得收益的同时需要支付发送蠕虫命令帧的成本以及由此动作所引起的防御策略变化;感染节点通过参数调整概率 $p^{M_d}(t)$选择耗尽节点资源并使易感节点转变为失效节点,攻击方通过使失效节点无法在网络中发挥功能来增加蠕虫在博弈中的收益,但同时感染其他易感节点的 $\hat{I}(t)$的数量减少,$I(t)$增加的速度放缓,该部分收益减小.在 WSS 的防御策略集合$(p^{N_s}(t),p^{N_w}(t),p^{N_r}(t),p^{N_q}(t))$中,当节点处于休眠状态时,节点的通信功能关闭,$p^{N_s}(t)$通过使易感节点处于休眠状态减小被感染的速率,与此同时,感染节点处于休眠状态也会降低安全补丁增加恢复节点的速率,休眠状态时节点失去通信功能也会使防御方的支付成本增加;$p^{N_w}(t)$控制节点由休眠状态转变为工作节点的速率,其效果与 $p^{N_s}(t)$相反;$p^{N_r}(t)$控制其他状态节点转变为恢复节点的速率,当该参数增加时,感染节点转变为恢复节点的速率提高使防御方收益增加,同时在其他状态节点转变为恢复节点的过程中,基站发送大量安全补丁占用通信资源使防御方收益减小;$p^{N_q}(t)$调节防御方安全机制隔离感染节点的速率,隔离感染节点可以减小蠕虫的传播速率,同时会增加系统运行安全机制的成本,感染节点失去原有的数据转发等功能也会增加防御方的支付成本.

设集合$(m^*(t),s^*(t))=[(p^{M_r^*}(t),p^{M_d^*}(t)),(p^{N_s^*}(t),p^{N_w^*}(t),p^{N_r^*}(t),p^{N_q^*}(t))]$为动态微分博弈的最优解,对于蠕虫攻击方,最大化最小成本函数 J 的值可表示为 $V_0=\min\limits_{s(t)}\max\limits_{m(t)}J(m(t),s(t))$,此时得到的策略中,攻击方采取的策略为最优值 $m^*(t)$;对于 WSS 防御方,最小化最大 J 的值可表示为

$V_1=\max\limits_{m(t)}\min\limits_{s(t)}J(m(t),s(t))$,此时防御方采取的策略为最优值 $s^*(t)$. 由此可得,$V_1\leqslant V_0$. $\forall m(t)$, $s(t)$,当博弈 K 存在鞍点策略$(m^*(t),s^*(t))$时,满足关系:

$$J(m(t),s^*(t))\leqslant J(m^*(t),s^*(t))\leqslant J(m^*(t),s(t)).$$

令 $V=J(m^*(t),s^*(t))$,当攻击方选择策略 $m^*(t)$时,无论防御方采取何种防御措施,成本函数的值至少为 V;当防御方采取策略 $s^*(t)$时,成本函数的值至多为 V. 当系统采取鞍点策略时,满足关系 $V=V_0=V_1$.

由于成本函数 J 与 WSS 的 QoS 相关,被蠕虫感染的节点数据的机密性和隐私性遭到破坏,且影响节点的正常工作,节点持有的内部资源被非法调用和读取,定义此时蠕虫在感染节点集合中引起的瞬时成本为 $\mu_I I(t)$,其中 μ_I 为协议中感染节点的瞬时成本相关系数. 失效节点的增加会导致网络中路由路径发生改变,部分数据无法正常采集和传递,使成本函数值增加,由此,定义失效节点的瞬时成本为 $\mu_D D(t)$,μ_D 为失效节点的瞬时成本相关系数. 同理可得,恢复节点的瞬时成本为 $\mu_R R(t)$,隔离节点的瞬时成本为 $\mu_Q Q(t)$,休眠节点的瞬时成本函数分别为 $\mu_{S_1}S_1(t)$,$-\mu_{I_1}I_1(t)$和 $\mu_{R_1}R_1(t)$.

隔离节点、植入安全补丁的恢复节点以及感染节点转换的休眠节点将有利于 WSS 中 QoS 收益的增加,其相应的瞬时成本分别为 $\lambda_{IQ}p^{N_q}(t)$, $\lambda_R p^{N_r}(t)$,$\lambda_{II_1}p^{N_s}(t)$. 终期成本 $\mu_D^{lT_0}(D(lT_0)-D((l-1)T_0))$是指失效节点在第 l 个阶段博弈结束时需要重新更换失效节点所带来的成本.

由以上分析可得,定义 1 中的成本函数可表示为

$$\begin{aligned}J(m(t),s(t))=&\int_{(l-1)T_0}^{lT_0}(\mu_I I(t)+\mu_R R(t)+\\&\mu_D D(t)+\mu_{S_1}S_1(t)-\mu_{I_1}I_1(t)+\mu_{R_1}R_1(t)+\\&\mu_Q Q(t)-\lambda_{IQ}p^{N_q}(t)-\lambda_R p^{N_r}(t)-\lambda_{II_1}p^{N_s}(t))\mathrm{d}t+\\&\mu_D^{lT_0}(D(lT_0)-D((l-1)T_0)).\end{aligned}\tag{12}$$

为了求解博弈 K 的鞍点策略,首先需要分析该博弈的鞍点是否存在.

定理 1. 如果 WSS 中蠕虫防御微分博弈满足定义 1 及其约束条件,那么该博弈存在使系统达到最优的鞍点策略.

证明. 微分博弈存在鞍点策略必须满足 4 个条件:

1) 系统的状态函数在各状态和控制策略上是连续且有界的;

2) 在成本函数中,瞬时成本和终期成本在各状态和控制策略上是连续的;

3) 状态函数和瞬时成本函数是由控制策略线性表示的,且博弈中定义的控制策略集合均是凸的;

4) 攻击方和防御方的控制策略是有界的.

由定义 1 可知,状态函数 $f(t,r(t),m(t),s(t))$是连续的且其各状态变量的取值由式(4)~(11)确定,M 和 S 对控制策略的上下限进行约束,因此,状态函数在各状态和控制策略上是连续且有界的,条件 1 显然满足. 瞬时成本和终期成本随着各状态和控制策略的增加而线性增加,其在各状态和控制策略上是连续的,满足条件 2. 由式(4)~(12)可知,状态函数和瞬时成本函数均由控制策略线性表示,且控制策略集合中的参数可在连续区间内取值,故博弈中控制策略集合是凸的,满足条件 3. 由协议中控制策略的约束条件可知,条件 4 显然满足.

因此,博弈 K 存在使系统性能达到最优的鞍点策略. 证毕.

3 博弈模型的求解

本节以动态微分博弈 K 的目标成本函数最小为求解依据,通过引入汉密尔顿函数,将目标成本函数最小问题转化为判定双方策略参数对应系数的正负问题. 在此基础上,结合状态函数和初始状态值,求解不同系数范围下的最优控制策略.

由于参与者双方的控制策略参数取值是一个连续的取值空间,博弈模型的最优策略求解无法通过线性分析和分类数值对比实现,且参数取值的无限特性使函数的凸性分析无法直接求解. 因此,引入汉密尔顿函数作为对目标成本函数和各策略控制参数进行定量分析的工具,简化最优控制策略的求解问题.

8 种不同状态随时刻 t 变化的轨迹为 $r(t)$,第 l 个博弈阶段起始状态为 $r((l-1)T_0)=r_{(l-1)T_0}$,则可以构造系统 $\dfrac{\mathrm{d}r(t)}{\mathrm{d}t}=f(t,r(t),m(t),s(t))$,此处的函数 f 即为定义 1 中的状态函数. 当博弈达到鞍点时,满足 $\dfrac{\mathrm{d}r^*(t)}{\mathrm{d}t}=f(t,r^*(t),m^*(t),s^*(t))$. 其中,$r^*(t)$为采取最优策略控制时的各状态轨迹. 性能泛函 $J(m(t),s(t))=\int_{(l-1)T_0}^{lT_0}w(t,r(t),m(t),s(t))\mathrm{d}t+u(r(lT_0))-u(r((l-1)T_0))$ 的目的是系统从初始状态 $r((l-1)T_0)=r_{(l-1)T_0}$ 转移到终态 $r(lT_0)$ 的过程中,寻求使 J 取得最小值的最优策略控制$(m^*(t),s^*(t))$.

由于动态博弈存在鞍点策略,可以利用目标成本函数的性能泛函构建汉密尔顿函数,其协状态函数可表示为

$$\boldsymbol{\kappa}(t)=(\kappa_S(t)\ \kappa_I(t)\ \kappa_R(t)\ \kappa_{S_1}(t)\ \kappa_{I_1}(t)\ \kappa_{R_1}(t)\ \kappa_Q(t)\ \kappa_D(t))^{\mathrm{T}}.$$

利用拉格朗日乘子法,构造新的增广泛函:

$$J_1=\int_{(l-1)T_0}^{lT_0} w(t,r(t),m(t),s(t))\mathrm{d}t+\int_{(l-1)T_0}^{lT_0}\boldsymbol{\kappa}(t)\left[f(t,r(t),m(t),s(t))-\frac{\mathrm{d}r(t)}{\mathrm{d}t}\right]\mathrm{d}t, \tag{13}$$

其中,$\boldsymbol{\kappa}(t)$为 8 维的拉格朗日乘子系数矢量. 纯量函数 H 可表示为

$$H[t,\boldsymbol{\kappa}(t),r(t),m(t),s(t)]=w(t,r(t),m(t),s(t))+\boldsymbol{\kappa}(t)f(t,r(t),m(t),s(t)). \tag{14}$$

$H[t,\boldsymbol{\kappa}(t),d(t),m(t),s(t)]$为构造的汉密尔顿函数,则式(13)可简化为

$$J_1=\int_{(l-1)T_0}^{lT_0}\left\{H(t,\boldsymbol{\kappa}(t),r(t),m(t),s(t))-\boldsymbol{\kappa}(t)\frac{\mathrm{d}r(t)}{\mathrm{d}t}\right\}\mathrm{d}t. \tag{15}$$

$$\begin{aligned}H[t,\boldsymbol{\kappa}(t),d(t),m(t),s(t)]&=w(t,d(t),m(t),s(t))+\boldsymbol{\kappa}(t)f(t,d(t),m(t),s(t))=[\kappa_Q(t)I(t)-\lambda_{IQ}-\kappa_I(t)I(t)]\times p^{N_{\mathrm{q}}}(t)+[\beta\kappa_R(t)(S(t)+I(t)+Q(t))-\beta\kappa_I(t)I(t)-\beta\kappa_S(t)S(t)-\lambda_R-\beta\kappa_Q(t)Q(t)]\times p^{N_{\mathrm{r}}}(t)+[\kappa_{S_1}(t)S(t)+\kappa_S(t)S(t)+\kappa_{I_1}(t)I(t)+\kappa_{R_1}(t)R(t)-\lambda_{II_1}-\kappa_I(t)I(t)-\kappa_R(t)R(t)]\times p^{N_{\mathrm{s}}}(t)+[\kappa_S(t)S_1(t)+\kappa_R(t)R_1(t)+\kappa_I(t)I_1(t)-\kappa_{R_1}(t)R_1(t)-\kappa_{S_1}(t)S_1(t)-\kappa_{I_1}(t)I_1(t)]\times p^{N_{\mathrm{w}}}(t)+[\alpha(\kappa_I(t)-\kappa_S(t))\times(a\sqrt{I(t)}+b)\frac{S}{N}\xi]\times p^{M_{\mathrm{r}}}(t)+[(\kappa_D(t)-\kappa_I(t))I(t)]\times p^{M_{\mathrm{d}}}(t)+\kappa_D(t)pS(t)+\kappa_D(t)pR(t)-\kappa_R(t)pR(t)+\mu_II(t)+\mu_RR(t)+\mu_DD(t)+\mu_{S_1}S_1(t)+\mu_{R_1}R_1(t)+\mu_QQ(t)-\mu_{I_1}I_1(t).\end{aligned} \tag{16}$$

由式(15)展开欧拉方程可导出上述泛函极值的必要条件为

$$\begin{cases}\dfrac{\partial\boldsymbol{\kappa}(lT_0)}{\partial z}=\dfrac{\partial}{\partial z}(u(d(lT_0))-u(d((l-1)T_0))),\\ H(t,\boldsymbol{\kappa}(t),d^*(t),m(t),s^*(t))\leqslant H(t,\boldsymbol{\kappa}(t),d^*(t),m^*(t),s^*(t)),\\ H(t,\boldsymbol{\kappa}(t),d^*(t),m^*(t),s^*(t))\leqslant H(t,\boldsymbol{\kappa}(t),d^*(t),m^*(t),s(t)),\\ \dfrac{\partial\boldsymbol{\kappa}(t)}{\partial z}=-\dfrac{\partial}{\partial z}H(t,\boldsymbol{\kappa}(t),d^*(t),m^*(t),s^*(t)),\\ \forall(m(t),s(t))\in M\times S.\end{cases}$$

由式(12)和式(14)可得汉密尔顿函数如式(16)所示.

设汉密尔顿函数中各策略控制参数的系数分别为

$$\chi^{N_{\mathrm{q}}}(t),\chi^{N_{\mathrm{r}}}(t),\chi^{N_{\mathrm{s}}}(t),\chi^{N_{\mathrm{w}}}(t),\chi^{M_{\mathrm{r}}}(t),\chi^{M_{\mathrm{d}}}(t),$$

可得:

$$\begin{cases}\chi^{N_{\mathrm{q}}}(t)=\kappa_Q(t)I(t)-\lambda_{IQ}-\kappa_I(t)I(t),\\ \chi^{N_{\mathrm{r}}}(t)=\beta\kappa_R(t)(S(t)+I(t)+Q(t))-\beta\kappa_I(t)I(t)-\beta\kappa_S(t)S(t)-\lambda_R-\beta\kappa_Q(t)Q(t),\\ \chi^{N_{\mathrm{s}}}(t)=\kappa_{S_1}(t)S(t)+\kappa_S(t)S(t)+\kappa_{I_1}(t)I(t)+\kappa_{R_1}(t)R(t)-\lambda_{II_1}-\kappa_I(t)I(t)-\kappa_R(t)R(t),\\ \chi^{N_{\mathrm{w}}}(t)=\kappa_S(t)S_1(t)+\kappa_R(t)R_1(t)+\kappa_I(t)I_1(t)-\kappa_{R_1}(t)R_1(t)-\kappa_{S_1}(t)S_1(t)-\kappa_{I_1}(t)I_1(t),\\ \chi^{M_{\mathrm{r}}}(t)=\alpha(\kappa_I(t)-\kappa_S(t))(a\sqrt{I(t)}+b)\dfrac{S}{N}\xi,\\ \chi^{M_{\mathrm{d}}}(t)=(\kappa_D(t)-\kappa_I(t))I(t).\end{cases} \tag{17}$$

依据汉密尔顿函数以及泛函极值的必要条件,可得协状态函数的微分方程及极值,进而结合相应的初始条件得到协状态函数 $\boldsymbol{\kappa}(t)$中的 8 个空间向量值,将其代入到式(17)可求解各策略控制参数的系数.

定理 2. 如果蠕虫与 WSS 之间的微分博弈满足定义 1,那么在动态微分博弈 K 中攻击方蠕虫 A 的最优控制策略为

$$p^{M_{\mathrm{r}}^*}(t)=\begin{cases}1, \chi^{M_{\mathrm{r}}}(t)\geqslant 0,\\ 0, \chi^{M_{\mathrm{r}}}(t)<0,\end{cases}$$

$$p^{M_{\mathrm{d}}^*}(t)=\begin{cases}1, \chi^{M_{\mathrm{d}}}(t)\geqslant 0,\\ 0, \chi^{M_{\mathrm{d}}}(t)<0.\end{cases}$$

防御方 WSS D 的最优控制策略为

$$p^{N_{\mathrm{q}}^*}(t)=\begin{cases}p_{\max}^{N_{\mathrm{q}}}, \chi^{N_{\mathrm{q}}}(t)<0,\\ 0, \chi^{N_{\mathrm{q}}}(t)\geqslant 0,\end{cases}$$

$$p^{N_{\mathrm{r}}^*}(t)=\begin{cases}1, \chi^{N_{\mathrm{r}}}(t)<0,\\ 0, \chi^{N_{\mathrm{r}}}(t)\geqslant 0,\end{cases}$$

$$p^{N_{\mathrm{s}}^*}(t)=\begin{cases}p_{\max}^{N_{\mathrm{s}}}, \chi^{N_{\mathrm{s}}}(t)<0,\\ 0, \chi^{N_{\mathrm{s}}}(t)\geqslant 0,\end{cases}$$

$$p^{N_{\mathrm{w}}^*}(t)=\begin{cases}1, \chi^{N_{\mathrm{w}}}(t)<0,\\ 0, \chi^{N_{\mathrm{w}}}(t)\geqslant 0.\end{cases}$$

证明. 由于汉密尔顿函数可表示为策略控制参

数与相应系数的线性表达式,且泛函极值的必要条件中要求 WSS 中蠕虫动态防御最优策略的鞍点必须满足:

$$\begin{cases}(m^*(t),s^*(t))\in \arg\min\limits_{s(t)}\max\limits_{m(t)} H[t,\boldsymbol{\kappa}(t),\\ \quad d(t),m(t),s(t)],\\ (m^*(t),s^*(t))\in \arg\max\limits_{m(t)}\min\limits_{s(t)} H[t,\boldsymbol{\kappa}(t),\\ \quad d(t),m(t),s(t)].\end{cases}$$

因此,当 WSS 策略控制参数相应的系数大于 0 时,策略控制参数取约束范围内的最小值,反之取最大值. 当蠕虫策略控制参数相应的系数大于 0 时,策略控制参数取约束范围内的最大值,反之取最小值.

证毕.

定理 2 给出了如何选择攻击方蠕虫 A 和防御方 WSS D 的最优控制策略. 该鞍点选取机制通过每个博弈阶段开始时策略控制参数的系数大小决定采取的策略,这种策略更新机制适合基于无线传感器网络技术的 WSS 计算资源有限的特点. 当系统初始状态确定时,利用式(4)~(11)求解每个阶段开始时各状态节点的数量,式(17)和协状态函数的微分方程求解协状态函数中状态空间的值,由此可得到确定最优控制策略的系数 $\chi^{N_q}(t),\chi^{N_r}(t),\chi^{N_s}(t),\chi^{N_w}(t),\chi^{M_r}(t),\chi^{M_d}(t)$. WSS 中嵌入上述蠕虫动态防御策略后,防御系统以博弈周期 T_0 更新防御策略参数,整个防御过程分为 $k=\lceil T/T_0\rceil$ 个不同的博弈阶段.

4 动态防御最优策略算法

依据第 l 个博弈阶段的状态节点数量初始值、状态函数、协状态函数以及各控制策略系数,由协议计算使蠕虫动态防御最优的鞍点策略. 该防御机制主要包括 4 个部分:参数存储模块(parameters storage module)、安全管理中心(security management center)、蠕虫 A(worm A)、WSS D. 在此基础上,设计适应于 WSS 的蠕虫动态防御最优策略机制如图 4 所示:

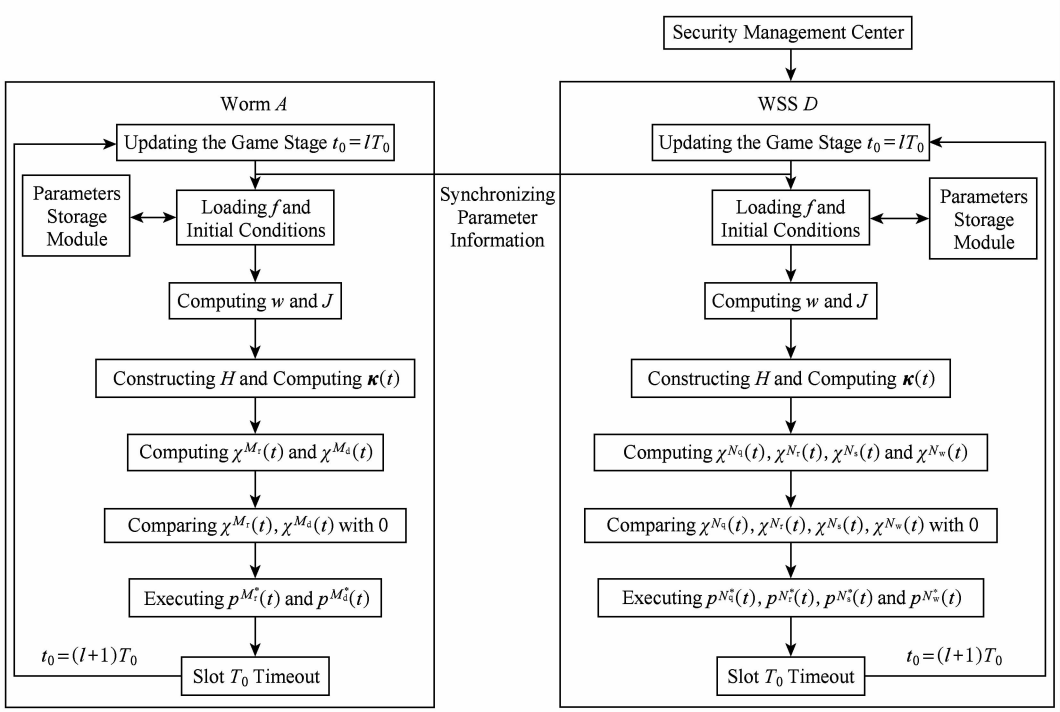


Fig. 4 Optimal defense strategy against worm propagation based on modified epidemic model

图 4 基于改进流行病模型的蠕虫动态防御最优策略机制

WSS 安全管理中心利用基站向整个网络配置网络参数、签名机制、安全加密算法以及博弈初始值,使各节点相互配合构建蠕虫防御机制. WSS D 和蠕虫 A 启动后更新博弈阶段(game stage)的起点 $t_0=lT_0$ 并同步参数信息,由各自的参数存储模块载入状态函数 f 以及上一个博弈阶段结束时各状态的初始值,利用状态转换微分方程计算瞬时成本 w 和目标成本函数 J. 结合瞬时成本函数、协状态函数和状态函数构造汉密尔顿函数,依据微分方程和协状态函数初始值求解各协状态 $\boldsymbol{\kappa}(t)$ 的取值. 将 $\boldsymbol{\kappa}(t)$

和各状态变量代入 $\chi^{M_r}(t)$ 与 $\chi^{M_d}(t)$，得到该博弈阶段 $\chi^{M_r}(t)$ 和 $\chi^{M_d}(t)$ 分别大于 0 时在时刻 t 的取值区间，进而得到蠕虫的最优策略控制集 $(p^{M_r^*}(t), p^{M_d^*}(t))$，蠕虫利用该策略集调整参数对网络实施攻击. 同理，可求解并更新 WSS D 的最优策略控制集 $(p^{N_q^*}(t), p^{N_r^*}(t), p^{N_s^*}(t), p^{N_w^*}(t))$，基站依据策略中参数的取值规律向网络中发送广播命令帧配置更新各节点. 当系统时钟计时结束(timeout)，更新博弈阶段的起点 $t_0=(l+1)T_0$，进入下一个博弈阶段.

本方案中基于目标成本值最优的蠕虫传播最优攻击算法和最优防御算法如下：

算法 1. WSS 蠕虫传播最优攻击算法.

输入：蠕虫控制参数的初始值 $p^{M_r^*}(0)=1$，$p^{M_d^*}(0)=1$；WSS 控制参数的初始值 $p^{N_q^*}(0)=p_{\max}^{N_q}$，$p^{N_r^*}(0)=1$，$p^{N_s^*}(0)=p_{\max}^{N_s}$，$p^{N_w^*}(0)=1$；不同状态节点的初始条件 $I(0)=m$，$R(0)=R_1(0)=S_1(0)=I_1(0)=Q(0)=D(0)=0$，$S(0)+I(0)+R(0)+S_1(0)+I_1(0)+R_1(0)+Q(0)+D(0)=N$；博弈阶段起点 $t=1$ 和博弈阶段总数 $k=\lceil T/T_0 \rceil$；

输出：从博弈开始到博弈阶段 k 对应的最优攻击策略集 $(p^{M_r^*}(0), p^{M_d^*}(0)), (p^{M_r^*}(1), p^{M_d^*}(1)), \cdots, (p^{M_r^*}(k), p^{M_d^*}(k))$.

Step1. 初始化式(4)～(11)的假设条件，设置时间窗 $t=1$，判断是否满足 $t\leqslant k$. 若满足条件，则利用状态微分方程求解该博弈阶段的各状态值；否则跳转到 Step5.

Step2. 结合攻击方和防御方控制参数初始值以及协状态微分方程，将时刻 lT_0 的协状态值代入求解不同博弈阶段的协状态值 $\kappa_S(t)$，$\kappa_I(t)$，$\kappa_R(t)$，$\kappa_{S_1}(t)$，$\kappa_{I_1}(t)$，$\kappa_{R_1}(t)$，$\kappa_Q(t)$，$\kappa_D(t)$.

Step3. 蠕虫控制的感染节点将当前博弈阶段的状态值、协状态值及初始条件代入式(16)，求解蠕虫控制参数的系数 $\chi^{M_r}(t)$ 和 $\chi^{M_d}(t)$.

Step4. 利用蠕虫控制参数系数的取值范围及定理 2 确定最优攻击策略取值 $p^{M_r^*}(t)$ 和 $p^{M_d^*}(t)$，设置 $t=t+1$.

Step5. 判断是否满足终止条件，若满足则输出各博弈阶段蠕虫最优攻击策略 $m^*(t)=(p^{M_r^*}(t), p^{M_d^*}(t))$，执行 Step6，否则跳转到 Step1.

Step6. 蠕虫依据 $m^*(t)=(p^{M_r^*}(t), p^{M_d^*}(t))$ 配置各博弈阶段的攻击策略值，算法结束.

算法 2. WSS 蠕虫传播最优防御算法.

输入：控制参数初始值及博弈阶段起点 $t=1$；

输出：从博弈开始到博弈阶段 k 对应的 WSS 系统最优防御策略集 $(p^{N_s^*}(0), p^{N_w^*}(0), p^{N_r^*}(0), p^{N_q^*}(0)), (p^{N_s^*}(1), p^{N_w^*}(1), p^{N_r^*}(1), p^{N_q^*}(1)), \cdots, (p^{N_s^*}(k), p^{N_w^*}(k), p^{N_r^*}(k), p^{N_q^*}(k))$.

Step1. 设置初始时间窗 $t=1$，配置初始化参数并由状态微分方程求解各博弈阶段状态值，判断是否满足 $t\leqslant k$，若满足执行下一步操作；否则跳转到 Step5.

Step2. 加载博弈中参与者的控制参数初始值以及协状态微分方程，将时刻 lT_0 的协状态值代入求解不同博弈阶段的协状态值.

Step3. WSS 中的基站将当前博弈阶段的状态值、协状态值及初始条件代入式(16)，求解 WSS 控制参数的系数 $\chi^{N_q}(t)$，$\chi^{N_r}(t)$，$\chi^{N_s}(t)$，$\chi^{N_w}(t)$.

Step4. 结合 WSS 控制参数系数的取值区间及定理 2，由判别条件确定最优防御策略取值 $p^{N_s^*}(t)$，$p^{N_w^*}(t)$，$p^{N_r^*}(t)$，$p^{N_q^*}(t)$，设置 $t=t+1$.

Step5. 判断是否满足终止条件，若满足则输出各博弈阶段 WSS 最优防御策略 $s^*(t)=(p^{N_s^*}(t), p^{N_w^*}(t), p^{N_r^*}(t), p^{N_q^*}(t))$，执行 Step6，否则跳转到 Step1.

Step6. 基站利用 $s^*(t)=(p^{N_s^*}(t), p^{N_w^*}(t), p^{N_r^*}(t), p^{N_q^*}(t))$ 发送广播命令帧配置各节点的防御策略值，算法结束.

5 实验与结果分析

5.1 实验设计

本实验在 HP Z238 工作站(3.2 GHz 8 核心超线程、8 GB 内存、Windows XP)上实现，使用的 WSS 网络仿真平台是 NS-2，NS-2 是由 UC Berkeley 开发的面向对象的无线网络仿真模拟器. 它通过设置虚拟时钟并逐个执行离散事件完成算法的运行. 从 VXHeaven 中抽取 5 个蠕虫样本并随机选取一种作为 WSS 蠕虫攻击源.

配置实验区域为半径 100 m 的圆形网络，采用 IEEE 802.15.4 MAC 层通信协议，整个网络由基站、簇头和终端节点构成，路由路径采取最短路径原则. 节点总数为 6 000 个，各节点通信距离(communication distance)为 50 m，数据收发速率(data transmission/reception rate)为 360 Kbps，单个数据帧平均长度(average length of the data frame)为 290 b. 攻击方和防御方的控制策略参数集合分别为 $(p^{M_r}(t), p^{M_d}(t))$ 和 $(p^{N_s}(t), p^{N_w}(t), p^{N_r}(t), p^{N_q}(t))$.

目标成本函数中相关的参数值分别为 $\mu_I=0.27$,$\mu_D=0.69$,$\mu_R=0.25$,$\mu_Q=0.48$,$\mu_{S_1}=0.82$,$\mu_{I_1}=0.71$,$\mu_{R_1}=0.36$,$\lambda_{IQ}=0.63$,$\lambda_R=0.83$,$\lambda_{II_1}=0.52$,$\mu_D^{IT_0}=1.79$.其他参数配置如表 1 所示:

Table 1 Parameter Configuration in NS-2

表 1 NS-2 参数设置

Parameters	Value
Radius of the Network Area/m	100
Number of the Nodes	6 000
Density, σ/m^2	0.192
Communication Distance, R_c/m	50
Data Transmission/Reception Rate/Kbps	360
Communication Protocol	IEEE 802.15.4
Average Length of the Data Frame/b	290
Initial Energy in Each Nodes/J	6
Power Dissipation of the Receiver/μW	800
Power Dissipation of the Sender/mW	22
Power Dissipation of the Sleep Node/μW	50
Initial Number of the Infected Nodes, $I(0)$	420
Total Number of the Game Stages, k	210
Correlation Coefficient of the Transition from S to I, α	0.6
Correlation Coefficient of the Transition from S or I to R, β	0.9
Maximum Probability of the Transition to Dormant Mode, $p_{\max}^{Ns}$	0.3
Maximum Probability that Infected nodes are quarantined, $p_{\max}^{Nq}$	0.7

NS-2 配置细胞自动机算法[12]的类型为混沌型,采用诺伊曼细胞空间对网络节点进行编号,节点 i 的状态为 $S_i\in\{1,2,3,4,5,6,7,8\}$,该节点的 2 邻居状态分别为$(S_i-1,t)$和$(S_i+1,t)$,利用转换规则函数表切换各节点状态.

Enhanced-AAWP 扫描算法[21]中蠕虫实例的平均扫描速率为 $\eta=7$,一个扫描时间单元的长度为 60 s,系统采取随机扫描和本地扫描的概率分别为 $P_1=0.875$ 和 $P_2=0.125$.

在上述网络环境下运行细胞自动机算法[12]、Enhanced-AAWP 扫描算法[21]和本文算法,利用 NS-2 中的 State sniffer 进程追踪并统计各状态节点以及策略参数的变化情况,结合 MATLAB 2010a 对实验结果进行数值分析.通过仿真实验对比这 3 种算法在 WSS 蠕虫防御方面的性能差异.

本次实验有 2 个目的:1)通过对比 3 种算法在不同博弈阶段各状态节点所占比例(fraction of nodes)及目标成本值(value of overall damage),验证本文算法在 WSS 蠕虫防御方面的优势;2)通过改变 WSS 持有的 $p^{N_s}(t)$,$p^{N_w}(t)$,$p^{N_r}(t)$和 $p^{N_q}(t)$以及蠕虫持有的 $p^{M_r}(t)$和 $p^{M_d}(t)$,得到相应的目标成本值并与本文算法中 State sniffer 记录的策略值进行比较,验证本文算法是否在参数取值上达到最优.

5.2 方案性能测试

通过仿真环境实现文献[12,21]中的方案及初始条件,对比不同状态节点所占比例随时间的变化曲线.为了得到更加精确的结果,重复实验 20 次并取所有结果的平均值.

如图 5 所示,在文献[21]中,网络模型通过 Enhanced-AAWP 随机扫描和本地扫描蠕虫,未考虑休眠节点和隔离节点对策略均衡的影响,导致在计算防御策略的阈值时存在较大误差,S 和 I 的实验测定值与文献[21]中均衡方程的计算值误差超过 6%.在第 1 个博弈阶段,算法中防御参数的切换延迟,易感节点所占的比例迅速下降.感染节点所占比例在第 43 个博弈阶段达到最大值.随着恢复节点所占比例的增加,感染节点缩小对周围节点发送蠕虫命令帧的数据量,开始执行使感染节点转变为失效节点的策略.该方案在建立脆弱性主机自治系统的全局可达网络时假设状态为 I 的节点均有蠕虫传播能力,而感染节点区域内距离外部易感节点距离超过 R_c 的感染节点不具备感染能力.在最终平衡态时易感节点的比例保持在 2.19%~3.06%,由感染节点转变为失效节点的比例为 32.29%,方案的均衡策略不能达到实际的最优抑制效果.

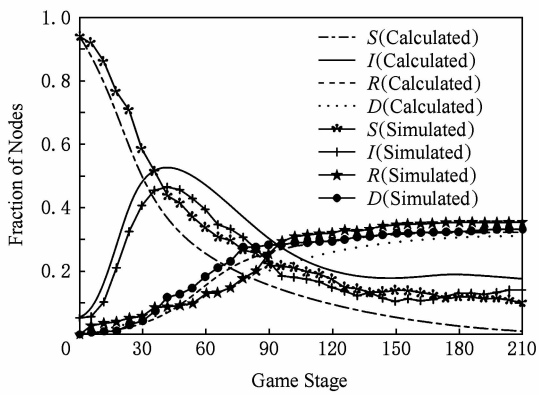


Fig. 5 Fraction curve of nodes in each state of Ref [21]

图 5 文献[21]中各状态节点曲线

与文献[21]相比,文献[12]在建立网络模型时考虑了靠近感染节点区域内侧时无法与外侧易感节点通信的情况,但在流行病模型的计算中未引入休眠

节点和隔离节点的微分方程. 如图 6 所示,易感节点所占比例在整个博弈阶段明显优于文献[21]. 当博弈结束时,易感节点所占比例的稳态值 7.69%大于文献[21]. 感染节点所占比例的曲线波峰到来的时间迟于文献[21]. 失效节点增加的速度有所减小,且稳态时所占比例保持在 21.31%,低于 Enhanced-AAWP 中的 32.29%,由于休眠节点和隔离节点对蠕虫传播具有抑制作用,该方案将这 2 种节点均作为易感节点进行数值计算,防御策略的参数取值时间不够准确. 失效节点和感染节点所占比例均超过 18%.

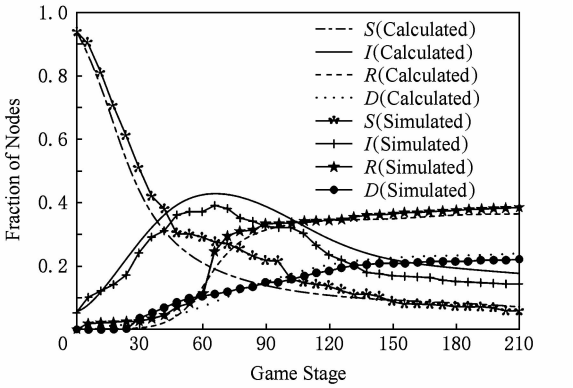


Fig. 6 Fraction curve of nodes in each state of Ref [12]

图 6 文献[12]中各状态节点曲线

本文方案中各状态节点所占比例的理论值与仿真计算值如图 7 所示. 动态防御最优策略算法在 *S-I-R* 动力学模型的基础上添加了休眠状态(S_1, I_1, R_1)以及安全隔离状态(Q),且利用 $R_c=50$ 及式(3)计算了具备蠕虫传播能力的感染节点. 易感节点在稳态时的比例保持在 19.17%～20.62%,失效节点的比例在整个博弈阶段低于 10%,感染节点比例控制在 17.29%～18.06%. 感染节点的比例在第 96 个博弈阶段时达到最大值 22.17%,远低于其他防御策略. 从第 96 个博弈阶段开始,基站向网络中发送大量安全补丁,同时大量节点处于休眠状态阻止蠕虫在系统中的传播,因此,恢复节点的比例并未迅速增长. 与上述 2 种方案相比,改进流行病模型得到的鞍点策略在蠕虫传播方面具有更好的抑制效果.

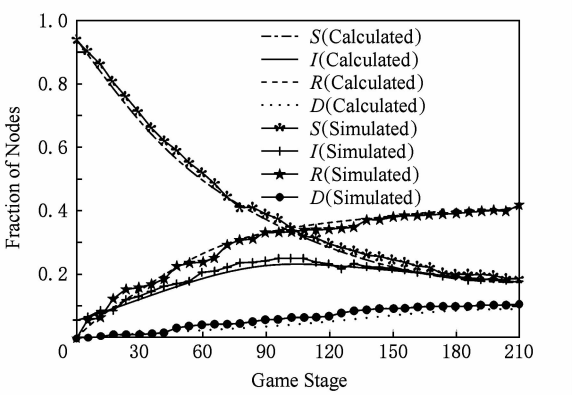


Fig. 7 Fraction curve of nodes in each state of this paper

图 7 本文中各状态节点曲线

State sniffer 采集到的数据代入式(12)得到各算法目标成本值随时间的变化情况,如图 8 所示. $I(0)=120$ 的目标成本值不到 $I(0)=240$ 的目标成本值的 40%,这是由于达到均衡时 $I(t)$ 和 $D(t)$ 在目标成本值中所占比例

$$\left[\int_{(l-1)T_0}^{lT_0}(0.27I(t)+0.69D(t))\mathrm{d}t+1.79(D(lT_0)-D((l-1)T_0))\right]\Big/J(m(t),s(t))$$

较大. 随着博弈阶段的更新,当 $I(0)=120$ 和 $I(0)=240$ 时,本文算法在各阶段达到策略均衡时的目标成本值均远小于其他算法,且差值逐渐扩大. 基于改进流行病模型和目标成本值最优的抑制算法在 WSS 蠕虫传播防御方面具有明显优势.

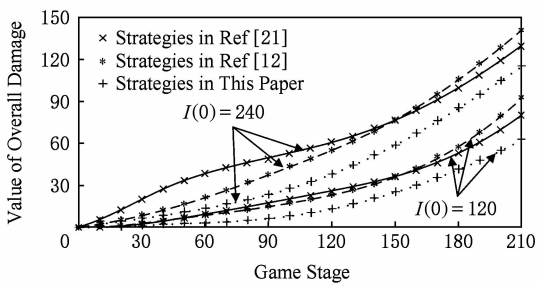


Fig. 8 Comparison of target costs in different strategies

图 8 不同策略下的目标成本值对比

5.3 参数分析

为了研究各参数之间的关系,设置 $I(0)=240$,在第 50 个博弈阶段,当 WSS 采取最优策略集($p^{N_s^*}(t)$, $p^{N_w^*}(t)$, $p^{N_r^*}(t)$, $p^{N_q^*}(t)$)时,蠕虫取不同参数值时的目标成本值如图 9 所示.

蠕虫的目标是使目标成本函数的值最大. 如图 9 中的取值点(1,0,27.2),当 $p^{M_r}(t)=1$ 且 $p^{M_d}(t)=0$ 时,目标成本值达到最大值;当 $p^{M_r}(t)=0$ 且 $p^{M_d}(t)=1$ 时,感染节点迅速减小,大量感染节点转变为失效节点,目标成本值达到最小值.

在第 125 个博弈阶段,保持其他参数依据表 1 中的策略取值,$p^{N_w}(t)$ 和 $p^{N_r}(t)$ 在取值空间[0,1]变化,得到的目标成本值分布如图 10 所示. 图 10 中取值点(1,1,28.6)为目标成本值取得最小值的点,在

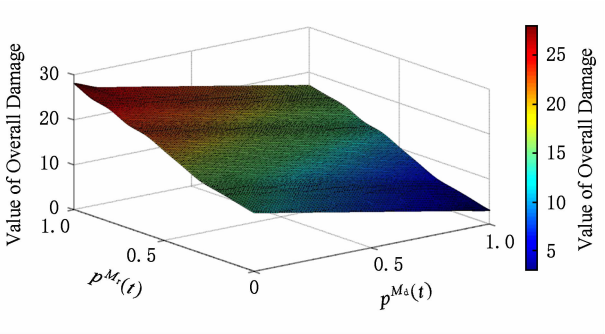


Fig. 9 Target cost under different $p^{M_r}(t)$ and $p^{M_d}(t)$

图 9 不同 $p^{M_r}(t)$和 $p^{M_d}(t)$下的目标成本值

这个博弈阶段,感染节点数量较小且隔离节点数量较大,当 $p^{N_w}(t)=1$ 且 $p^{N_r}(t)=1$ 时,被系统唤醒的易感节点和恢复节点数量增加,同时隔离节点转变为恢复节点加入网络进行数据处理,使目标成本降低.该控制变量的最优策略取值与图 8 保持一致.

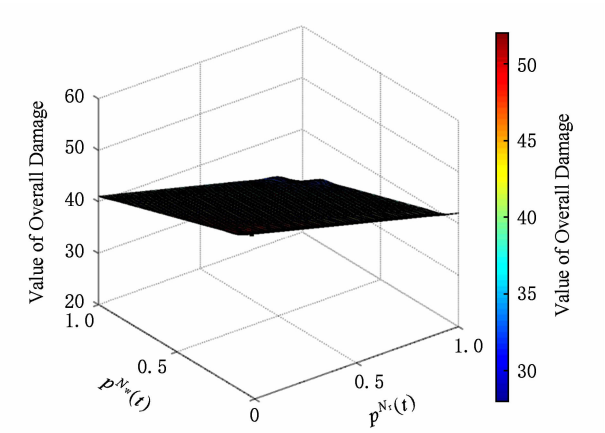


Fig. 10 Target cost under different $p^{N_w}(t)$ and $p^{N_r}(t)$

图 10 不同 $p^{N_w}(t)$和 $p^{N_r}(t)$下的目标成本值

在第 55 个博弈阶段,$p^{N_s}(t)$和 $p^{N_q}(t)$分别在取值空间[0,0.7]与[0,0.8]变化,蠕虫和 WSS 的其他控制参数依据本文算法的最优策略进行配置,得到目标成本值分布如图 11 所示.当且仅当 $p^{N_s}(t)=0$ 和 $p^{N_q}(t)=0.8$ 时,目标成本函数取得最小值.

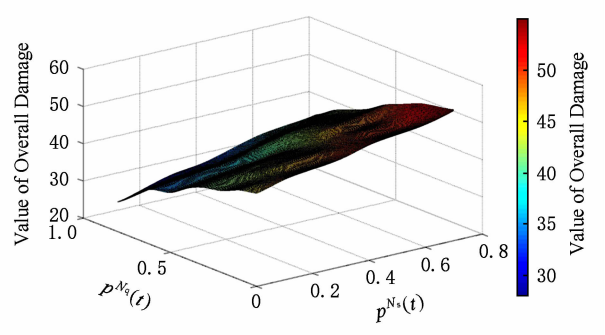


Fig. 11 Target cost under different $p^{N_s}(t)$ and $p^{N_q}(t)$

图 11 不同 $p^{N_s}(t)$和 $p^{N_q}(t)$下的目标成本值

由图 9~11 可知:本文中 WSS 蠕虫传播最优攻击算法以及最优防御算法的控制参数取值相对于目标成本函数具有最优特性,即图 4 中的方案在不同博弈阶段的参数取值对攻击者和防御者均为最优策略.基于目标成本值最优的蠕虫传播抑制算法可以最大限度地提高 WSS 的安全性能.

6 结束语

物联网 WSS 中的蠕虫传播问题严重限制了其在工业生产、灾害预警、智能管理等领域的应用.为了解决这一问题,自主研究具有最优特性的 WSS 蠕虫传播抑制算法.

本文提出了一种基于目标成本值最优的 WSS 蠕虫传播抑制算法,和其他同类型算法相比具有 3 个方面优点:

1)在传统 SIR 模型的基础上,引入休眠状态和隔离状态,建立了状态转换图和改进的流行病模型,并给出了各状态节点的微分方程及初始状态,使不同博弈阶段状态节点数量的计算更加准确;

2)本文构建了蠕虫传播的系统模型,利用节点的通信半径以及感染节点的分布规律,准确计算了具有蠕虫传播能力的感染节点数量,使感染节点的状态微分方程对下一时刻状态节点的预测更加精确;

3)本文建立了 WSS 和蠕虫的动态微分博弈模型,以目标成本值最优作为鞍点策略的求解条件,通过状态变量、协状态变量以及汉密尔顿函数的线性特点实现均衡策略的求解,在此基础上,提出了蠕虫传播抑制最优策略算法,与其他算法相比,本文算法在不同博弈阶段状态节点预测以及 WSS 蠕虫传播抑制方面具有明显优势.

参 考 文 献

[1] Tao Fei, Zuo Ying, Xu Lida, et al. IoT-based intelligent perception and access of manufacturing resource toward cloud manufacturing [J]. IEEE Transactions on Industrial Informatics, 2014, 10(2): 1547-1557

[2] He Wu, Yan Gongjun, Xu Lida. Developing vehicular data cloud services in the IoT environment [J]. IEEE Transactions on Industrial Informatics, 2014, 10(2): 1587-1595

[3] Gope P, Hwang T. BSN-Care: A secure IoT-based modern healthcare system using body sensor network [J]. IEEE Sensors Journal, 2016, 16(5): 1368-1376

[4] Hoff E, Laursen B, Bridges K. Security and privacy in Internet of things: Methods, architectures, and solutions [J]. Security and Communication Networks, 2014, 7(11): 2681-2682

[5] Zhang Yuanyu, Shen Yulong, Wang Hua, et al. On secure wireless communications for IoT under eavesdropper collusion [J]. IEEE Transactions on Automation Science and Engineering, 2016, 13(3): 1281-1293

[6] Xu Qian, Ren Pinyi, Song Houbing, et al. Security enhancement for IoT communications exposed to eavesdroppers with uncertain locations [J]. IEEE Access, 2016, 18(4): 2840-2853

[7] Pa Y M P, Suzuki S, Yoshioka K, et al. IoTPOT: A novel honeypot for revealing current IoT threats [J]. Journal of Information Processing, 2016, 24(3): 522-533

[8] Wang Fangwei, Zhang Yunkai, Wang Changguang, et al. Stability analysis of an e-SEIAR model with point-to-group worm propagation [J]. Communications in Nonlinear Science and Numerical Simulation, 2015, 20(3): 897-904

[9] Wang Xu, Ni Wei, Zheng Kangfeng, et al. Virus propagation modeling and convergence analysis in large-scale networks [J]. IEEE Transactions on Information Forensics and Security, 2016, 11(10): 2241-2254

[10] Tong Xiaojun, Zhang Miao, Wang Zhu. The cost optimal control system based on the Kermack - Mckendrick worm propagation model [J]. Journal of Algorithms & Computational Technology, 2016, 10(2): 82-89

[11] Chen Zhe, Ji Chuanyi. Spatial-temporal modeling of malware propagation in networks [J]. IEEE Transactions on Neural Networks, 2010, 16(5): 1291-1303

[12] Khanh N H. Dynamics of a worm propagation model with quarantine in wireless sensor networks [J]. Applied Mathematics & Information Sciences, 2016, 10(5): 1739-1746

[13] Bradley J T, Gilmore S T, Hillston J. Analyzing distributed Internet worm attacks using continuous state-space approximation of process algebra models [J]. Journal of Computer and System Science, 2008, 74(6): 1013-1032

[14] Ma Xiaolong, Liu Hui, Zhang Jing. Analysis of the impact of heterogeneous network environment on worm propagation [C] //Proc of the 3rd Int Conf on Multimedia Information Networking & Security. Piscataway, NJ: IEEE, 2011: 457-462

[15] Zhu Kenan, Yin Baolin, Mao Yaming, et al. Malware classification approach based on valid window and naïve Bayes [J]. Jounal of Computer Research and Development, 2014, 51(2): 373-381 (in Chinese)
(朱克楠, 尹宝林, 冒亚明, 等. 基于有效窗口和朴素贝叶斯的恶意代码分类[J]. 计算机研究与发展, 2014, 51(2): 373-381)

[16] Mao Weixuan, Cai Zhongmin, Tong Li. Malware detection method based on active learning [J]. Journal of Software, 2017, 28(2): 384-397 (in Chinese)
(毛蔚轩, 蔡忠闽, 童力. 一种基于主动学习的恶意代码检测方法[J]. 软件学报, 2017, 28(2): 384-397)

[17] Zou Changchun, Gong Weibo, Towsley D. On the performance of Internet worm scanning strategies [J]. Journal of Performance Evaluation, 2006, 63(7): 700-723

[18] Ma Hongliang, Wang Wei, Han Zhen. Detecting and de-obfuscating obfuscated malicious JavaScript code [J]. Chinese Journal of Computers, 2017, 40(7): 1699-1713 (in Chinese)
(马洪亮, 王伟, 韩臻. 混淆恶意 JavaScript 代码的检测与反混淆方法研究[J]. 计算机学报, 2017, 40(7): 1699-1713)

[19] Das S, Liu Yang, Zhang Wei, et al. Semantics-based online malware detection: Towards efficient real-time protection against malware [J]. IEEE Transactions on Information Forensics and Security, 2016, 11(2): 289-302

[20] Guo Yanghe, Ten C W, Hu Shiyan, et al. Preventive maintenance for advanced metering infrastructure against malware propagation [J]. IEEE Transactions on Smart Grid, 2016, 7(3): 1314-1328

[21] Liu Bo, Wang Huaimin, Xiao Fengtao, et al. Enhanced-AAWP, a heterogeneous network oriented worm propagation model [J]. Journal on Communications, 2011, 32(12): 103-113 (in Chinese)
(刘波, 王怀民, 肖枫涛, 等. 面向异构网络环境的蠕虫传播模型 Enhanced-AAWP [J]. 通信学报, 2011, 32(12): 103-113)

[22] Saracino A, Sgandurra D, Dini G, et al. Madam: Effective and efficient behavior-based android malware detection and prevention [J]. IEEE Transactions on Dependable and Secure Computing, 2018, 15(1): 83-97

[23] Theodoros S, Konstantinos M, Alexios M, et al. A game theoretical method for cost-benefit analysis of malware dissemination prevention [J]. Information Systems Security, 2015, 24(6): 164-176

[24] Mishra B K, Pandey S K. Dynamic model of worm propagation in computer network [J]. Applied Mathematical Modelling, 2014, 38(7): 2173-2179

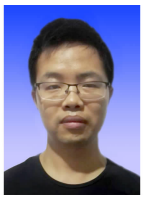
[25] Pandey S K, Samanta B. Malware attack through botnet in e-commerce network: A dynamic model [J]. International Journal Series in Engineering Science, 2016, 2(3): 13-25

[26] Sarah S, Ness B S, Saurabh B. Modeling and automated containment of worms [J]. IEEE Transactions on Dependable and Secure Computing, 2008, 5(2): 528-537

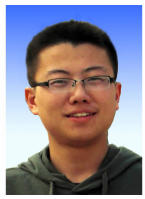
[27] Liu Bo, Zhou Wei, Gao Lei, et al. Malware propagations in wireless ad hoc networks [J]. IEEE Transactions on Dependable and Secure Computing, 2016, 34(4): 1365-1376

[28] Chen Pinyu, Cheng Shiming, Chen Kangcheng. Optimal control of epidemic information dissemination over networks [J]. IEEE Transactions on Cybernetics, 2014, 44(12): 2316-2328

[29] Yu Shui, Gu Guofei, Barnawi A, et al. Malware propagation in large-scale networks [J]. IEEE Transactions on Knowledge and Data Engineering, 2015, 27(1): 170-179



Huang Yicai, born in 1985. Master candidate. His main research interests include IoT, computational intelligence, wireless sensor networks, evolutionary algorithm and information security.



Zhou Weiwei, born in 1990. PhD candidate. His main research interests include IoT, computational intelligence, wireless sensor networks, evolutionary algorithm and information security.

Yu Bin, born in 1964. Professor and PhD supervisor of PLA Information Engineering University. His main research interests include the design and analysis of algorithms, visual cryptography, dynamic multi-objective optimization and network security.