

# 二阶差分的自适应 RED 算法研究

范训礼<sup>1</sup> 郑 锋<sup>1</sup> GUAN Lin<sup>2</sup> 高 黎<sup>1</sup>

(西北大学信息科学与技术学院 西安 710127)<sup>1</sup> (拉夫堡大学计算机系 英国莱斯特 LE11 3TU)<sup>2</sup>

**摘 要** 在现有的 AQM 算法的基础上,使用经典控制理论中的时域分析方法对 ARED 算法的内部结构进行分析研究,根据分析的结果设计了新的基于二阶差分的离散控制器,并将其应用到 AQM 算法中。在 NS-2 仿真平台上进行了大量的仿真,结果表明:基于二阶差分的 ARED 算法使得队列长度变得更加稳定,对队列具有更好的控制能力。

**关键词** 二阶差分 ARED, 主动队列管理, 队列长度

**中图分类号** TP301 **文献标识码** A

## Study of Adaptive RED Algorithm Based on 2<sup>nd</sup>-order Difference Equation

FAN Xun-li<sup>1</sup> ZHENG Feng<sup>1</sup> GUAN Lin<sup>2</sup> GAO Li<sup>1</sup>

(School of Information Science & Technology, Northwest University, Xi'an 710127, China)<sup>1</sup>

(Department of Computer Science, Loughborough University, Loughborough, LE11 3TU, UK)<sup>2</sup>

**Abstract** This paper analyzed and studied the internal structure of the adaptive RED algorithm through time-domain analysis using the classic control theory, which is based on AQM. According to the analyzing results, the paper designed a new discrete controller based on 2<sup>nd</sup>-order difference equation and applied it to the AQM algorithm. To illustrate the performance of the given algorithm, the paper simulated the prompted algorithm on NS-2 platform. Numerical simulation results show that the 2<sup>nd</sup>-order difference ARED algorithm can keep the queue length much more stable and has better control capability for the queue length.

**Keywords** 2<sup>nd</sup>-order difference ARED, Active queue management, Queue length

### 1 引言

网络中的拥塞来源于网络资源和网络流量分布的不均衡,使用拥塞控制算法对于网络的稳定具有十分重要的意义<sup>[1]</sup>。近年来,作为路由端拥塞控制的典范,基于主动队列管理(Active Queue Management, AQM)策略的拥塞控制已经成为人们的研究热点<sup>[3,6-8]</sup>。

随机早期检测算法(Random Early Detection, RED)<sup>[2]</sup>是由 Floyd 和 Jacobson 于 1993 年提出的,RED 比较突出的问题是参数的正确配置对性能有很大的影响。Feng Wu-chang 等人于 1999 年提出了参数自配置的 RED 算法<sup>[3]</sup>。Floyd 等于 2001 年提出自适应随机早期检测算法(Adaptive Random Early Detection, ARED)<sup>[4]</sup>,ARED 能够根据网络的负载变化情况动态地调整参数。

本文根据经典控制理论中差分方程与脉冲传递函数的关系,将 ARED 算法中的平均队长系统采用二阶差分方程描述,根据其动态响应的特性提出了基于二阶差分方程的 ARED 算法(2DE-ARED),并采用连续域-离散化的方法来设计新的离散控制器,从时域的角度分析系统的动态特性。在 NS-2 仿真平台上进行了大量的仿真实验分析,结果表明 2DE-ARED 使得队列长度的抖动性、延迟、丢包数等性能指

标均优于 ARED 算法,而且对队列具有很好的控制能力。

### 2 基于流体流理论的 TCP/IP 传输模型

2000 年 V. Misra 首先建立了基于单个拥塞路由器的模型,将该模型扩展为一个网络<sup>[5]</sup>。该模型描述了源端 TCP 和中间节点 AQM 之间的交互行为,是 PI 算法和 PID 算法的基础,其微分方程的描述如下:

$$\begin{aligned} W'(t) &= \frac{1}{R(t)} - \frac{W(t)W(t-R(t))}{2R(t-R(t))} p(t-R(t)) \\ q'(t) &= \frac{W(t)}{R(t)} N(t) - C \end{aligned} \quad (1)$$

式中,  $W(t)$  表示源端 TCP 的拥塞窗口大小;  $q(t)$  表示路由器的队列长度;  $R(t)$  表示一个数据包传输的往返时间,并且  $R(t) = q/C + T_p$ ;  $C$  表示链路容量;  $T_p$  表示链路上的传输延迟;  $N$  表示链路的负载;  $p$  表示数据包丢弃概率。

取  $(W, q)$  作为系统的状态变量,而取  $p$  作为系统的输入量,并且由  $q' = 0$  得到工作点为  $(W_0, q_0, p_0)$ ,  $p_0 = 2/W_0^2$ ,  $W_0 = R_0 C / N$ ,  $R_0 = q_0 / C + T_p$ 。假定  $N(t) \equiv N$ , 并且  $R(t) \equiv R_0$ , 在工作点将其展开,得到:

$$\begin{aligned} \delta W'(t) &= -\frac{N}{R_0^2 C} (\delta W(t) + \delta W(t-R_0)) - \frac{R_0^2 C}{2N^2} \delta p(t-R_0) \\ \delta q'(t) &= \frac{N}{R_0} \delta W(t) - \frac{1}{R_0} \delta q(t) \end{aligned} \quad (2)$$

到稿日期:2010-06-03 返修日期:2010-08-30 本文受陕西省教育厅自然科学基金(10JK684),教育部留学回国人员科研启动基金资助项目(20103045)和英国皇家科学项目(UK-RS-09-078)资助。

范训礼(1970-),男,博士,副教授,主要研究方向为网络控制、网络安全, E-mail: xunlifan@nwu.edu.cn; 郑 锋(1984-),男,硕士,主要研究方向为网络控制; Guan Lin(1978-),女,博士,讲师,主要研究方向为网络控制、排队理论。

式中,  $\delta W = W - W_0$ ,  $\delta q = q - q_0$ ,  $\delta p = p - p_0$ , 3个变量均表示相应量在工作点附近的微小扰动。在假定  $W_0 \gg 1$  时, 可以进一步去掉高频项, 得到如下的微分方程组:

$$\begin{aligned} \delta W'(t) &= -\frac{2N}{R_0^2 C} \delta W(t) - \frac{R_0 C^2}{2N^2} \delta p(t - R_0) \\ \delta q'(t) &= \frac{N}{R_0} \delta W(t) - \frac{1}{R_0} \delta q(t) \end{aligned} \quad (3)$$

根据微分方程组(3), 可以得到与之对应的方框图, 如图1所示。

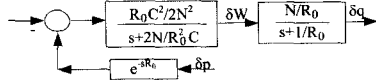


图1 简化模型的方框图

### 3 二阶差分的 ARED 算法

在文献[5]中, C. Hollot 结合 V. Misra 提出的 TCP/IP 传输模型并使用经典控制理论对 RED 算法的整体进行了分析, 文献[6]对 RED 的参数进行了分析, 而文献[7]中提出的 PID 算法也是基于 C. Hollot 所做的工作。不同于 PI 算法和 PID 算法, 本文关注的重点是 ARED 算法内部的计算平均队列长度的模块, 而不是将 ARED 算法和基于流体流理论的 TCP/IP 模型作为一个大的系统进行分析。

ARED 算法中, 采用了一阶差分方程来描述其中的平均队长系统。ARED 算法在描述平均队长时采用一阶差分方程, 其调节时间为  $14.98s^{[8]}$ 。本文期望将其调节时间的目标值定位为一阶差分方程的十分之一, 即  $1.4985s$ , 并根据这一目标值来设计控制器。本文设计二阶差分方程的离散控制器, 进而设计二阶差分的 ARED 算法, 并通过实验分析和验证了算法的有效性。

#### 3.1 二阶传递函数的一般形式:

二阶传递函数的一般形式为:

$$\frac{C(s)}{R(s)} = \frac{K}{T_m s^2 + s + K} \quad (4)$$

为了方便分析起见, 将其化为标准形式:

$$\frac{C(s)}{R(s)} = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \quad (5)$$

式中,  $\omega_n = \sqrt{K/T_m}$  称为系统的无阻尼自然频率,  $\zeta = 1/2\sqrt{T_m K}$  称为系统阻尼比。

对欠阻尼的二阶系统, 在动态过程中, 系统输出值与稳态值之间的偏差为:

$$\Delta c = c(\infty) - c(t) = \frac{e^{-\zeta\omega_n t}}{\sqrt{1-\zeta^2}} \sin(\sqrt{1-\zeta^2} \omega_n t + \theta) \quad (6)$$

求解方程(6)比较麻烦, 因此忽略掉正弦函数的影响, 即认为指数项衰减到 0.05 时, 动态调整完毕, 得到欠阻尼系统的调节时间为:

$$t_s(5\%) = \frac{1}{\zeta\omega_n} \left[ 3 - \frac{1}{2} \ln(1-\zeta^2) \right] \approx \frac{3}{\zeta\omega_n} \quad (7)$$

欠阻尼二阶系统的调节时间  $t_s$  和系统阻尼比  $\zeta$  与自然振荡角频率  $\omega_n$  的乘积成反比。根据前面的控制器设计目标, 有:

$$t_s(5\%) = 1.4985 \approx \frac{3}{\zeta\omega_n} \quad (8)$$

在系统阻尼比的取值为 0.1 时, 得到  $\omega_n$  为 20, 系统的二阶传递函数为:

$$G(s) = \frac{400}{s^2 + 4s + 400} \quad (9)$$

#### 3.2 基于二阶差分方程的 ARED 算法

路由器队列是一个典型的离散系统, 为了在路由器中实施算法, 需要将连续域的传递函数进行离散化, 得到相应的脉冲传递函数, 然后根据这样的脉冲传递函数来计算与之对应的差分方程, 最终根据差分方程来确定算法。

将零阶保持器的传递函数和系统的传递函数串联起来, 并且在经过  $z$  变换后, 就可以得到与系统的传递函数所对应的差分传递函数:

$$\begin{aligned} D(z) &= Z\left[\frac{G(s)}{s} - \frac{G(s)}{s} e^{-sT}\right] = Z\left[\frac{G(s)}{s}\right] - Z\left[\frac{G(s)}{s}\right] z^{-1} \\ &= (1-z^{-1}) Z\left[\frac{G(s)}{s}\right] \end{aligned} \quad (10)$$

通过  $z$  变换表得到差分方程:

$$D(z) = \frac{0.01967z + 0.01941}{z^2 - 1.922z + 0.9608} \quad (11)$$

对于平均队长而言, 其输入是中间节点的缓存队列长度, 而其输出则是平均队列长度:

$$D(z) = \frac{Avg(z)}{Q(z)} = \frac{0.01967z + 0.01941}{z^2 - 1.922z + 0.9608} \quad (12)$$

通过变换整理得到的差分方程为:

$$\begin{aligned} avg(k) &= 1.922avg(k-1) - 0.9608avg(k-2) + \\ &0.01967q(k) + 0.01941q(k-1) \end{aligned} \quad (13)$$

## 4 仿真分析

### 4.1 仿真环境

采用图 2 所示的拓扑结构对算法进行仿真。

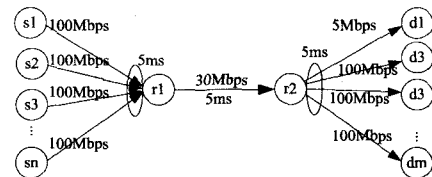


图2 实验的拓扑结构

源端 TCP 节点与路由器节点  $r_1$  之间的链路容量为 100Mbps, 链路延迟为 50ms。而对于路由器  $r_1$  与路由器  $r_2$  之间的瓶颈链路, 其容量和延迟分别为 30Mbps 和 5ms, 路由器节点  $r_2$  和节点  $d_n$  之间的链路容量为 100Mbps, 并且其链路的时延为 20ms。除了路由器节点  $r_1$  的队列使用 ARED 算法或者 2DE-ARED 算法外, 其余队列均使用 Drop Tail 算法, 并且设定所有队列的物理缓冲大小为 300packets。

### 4.2 队列长度性能仿真实验

实验时间设置为 200s, 200 个源端 TCP 节点与路由器节点 A 相连接。参数设置如下: 最小和最大阈值为 100 和 200 个包, 权重为 0.002, 最大丢弃概率的初值设置为 0.1。

图 3 给出了 ARED 和 2DE-ARED 所控制的瞬时队列长度。当设置最小阈值为 100 个包, 且最大阈值为 200 个包时, ARED 的期望队列长度为 150 个包。2DE-ARED 能够使得路由器的队列较为稳定, 比较有效地将队列长度稳定在期望队列长度左右。ARED 的瞬时队列长度的标准差为 23.9998, 2DE-ARED 为 11.5301, 说明 2DE-ARED 的波动性较小。

图 4 给出了 ARED 和 2DE-ARED 对丢包个数的影响情况。图 4 是对图 3 所示实验的数据处理结果, 分别统计路由

器 A 节点每 0.5s 丢弃的数据包的个数。2DE-ARED 的丢包个数的变化不是很剧烈,2DE-ARED 在丢弃数据包方面表现得比较温和,并且其每半秒钟丢弃的数据包的个数基本上都在 200 到 250 个之间变动,本文将路由器成功转发数据包的个数与路由器收到的数据包的个数之比用作传输效率。

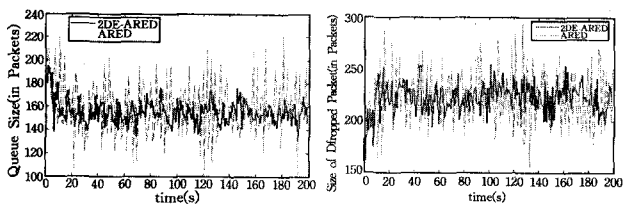


图 3 不同算法的瞬时队长比较

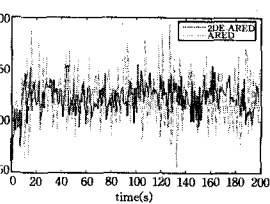


图 4 不同算法的丢包数

图 5 给出了 ARED 和 2DE-ARED 的传输效率。2DE-ARED 在传输效率方面表现得更加平稳一些,这主要是采用 2DE-ARED 对丢包个数的变化影响比较平稳而导致的。

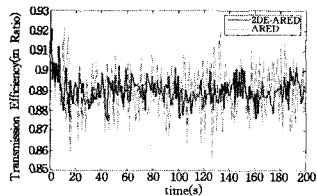


图 5 不同算法的传输效率

#### 4.3 阈值不断改变的仿真实验

在 ARED 算法中,队列长度的最小阈值和最大阈值也是一对比较重要的参数。针对这两个参数的改变,研究 2DE-ARED 在最小阈值、最大阈值不断改变时其性能的变化情况。图 6 是在同时增大最小阈值和最大阈值时瞬时队长长度的标准差结果。

仿真中,首先是固定最大阈值和最小阈值的差值,将差值固定为 100,逐步增加最小阈值,使得最小阈值从 85 开始,以 5 为步长,增加到 180,那么相应地,最大阈值就会从 185 增加到 280。其次,设置源端 TCP 节点的个数为 200 个,其他参数的设置同前。从图 6 中可以看出,2DE-ARED 在最小阈值和最大阈值不断抬升的过程中,能够较有效地将瞬时队长长度的标准差维持在一个较低的水平,即能够较为有效地控制瞬时队长长度的波动性。

图 7 给出了随着最小阈值的不断增加瞬时队长长度的平均值。在最小阈值和最大阈值不断抬升的过程中,算法瞬时队长长度的平均值是伴随上升的。

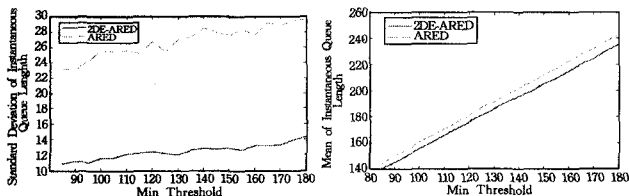


图 6 不同最小阈值下瞬时队长的标准差

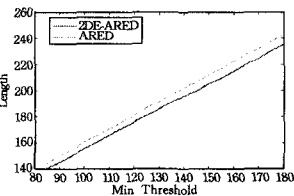


图 7 不同最小阈值下瞬时队长的平均值

不管一阶差分方程的算法还是 2DE-ARED,其瞬时队长长度的平均值的变化规律都是一条随着最小阈值的增加而增加的直线。

图 8 和图 9 分别描绘了上述实验中丢包率的标准差以及丢包率的平均值的变化情况。

随着最小阈值和最大阈值的增加,2DE-ARED 的丢包率的变化相对比较平缓。由于最小阈值和最大阈值的增加不会给路由器节点带来太多额外的数据量,因此算法没有必要对丢包率进行非常灵活的控制,丢包率的变化也相对平稳。

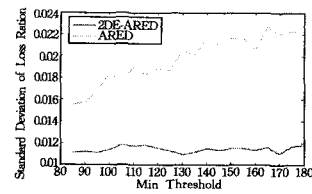


图 8 不同最小阈值下丢包率的标准差

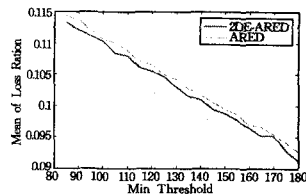


图 9 不同最小阈值下丢包率的平均值

#### 4.4 TCP 链接数和传输延迟同时改变

实验中,同时改变 TCP 链接数和网络的传输延迟,以更好地模拟现实中变化频繁的网络环境,并通过这样的实验来进一步观察算法的实际性能,同时实验中增加了与 PID 算法的比较。

图 10 绘制了算法的瞬时队长长度的标准差。在 TCP 链接数较少时,瞬时队长长度的标准差比较大,随着传输延迟的增加而增大,TCP 链接较少时,来自各个源端 TCP 链接的数据包也比较少。在 TCP 链接较少时丢包所造成的数据量的减少对路由器的影响较大,造成了瞬时队长长度的较大波动,并且该波动性也会随着传输延迟的增大而变得更加剧烈。在 TCP 链接数为 1000、传输延迟为 100ms 时,算法的瞬时队长长度的标准差达到 20,而在平稳状态下的瞬时队长长度的标准差也在 15 左右。

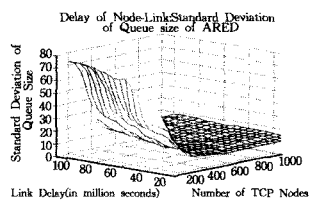


图 10 ARED 的队长标准差

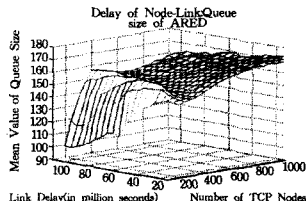


图 11 ARED 的队长平均值

如图 11 所示,在 TCP 链接数较小时,瞬时队长长度的平均值较低,而且随着 TCP 链接数的增大,队长长度的平均值会有一个短暂的向下趋势,之后就一直处于增加状态。

当固定 TCP 的链接数为某个数值时,随着传输延迟的增加,瞬时队长长度的平均值逐步减小。传输延迟的增大会导致 RTT 的增大,使得 TCP 链接的吞吐量减小,最终导致瞬时队长长度的平均值减小。在 TCP 链接数为 1000,并且节点 B 和节点 C 之间的链路时延为 5ms 时,其瞬时队长长度的平均值达到了 170 左右。

图 12 给出了采用二阶差分方程的算法在 TCP 链接数和传输延迟同时改变时其瞬时队长长度的标准差。在源端 TCP 的链接数较少时,瞬时队长长度的波动性也会比较大。相比之下,2DE-ARED 能够较为有效地控制其标准差。而在平稳区域内,算法的瞬时队长长度的标准差最小会达到 10 左右。在 TCP 链接数较大、网络传输延迟也较大的区域,虽然有一定的波动,但其最大不超过 20。

图 13 则给出相应的瞬时队长长度的平均值。2DE-ARED 即便是在 TCP 链接数和传输延迟不断变化的情况也能够较为有效地使瞬时队长的平均值维持在比较平稳的曲面上,并且平均值最大也只是在 160 附近。

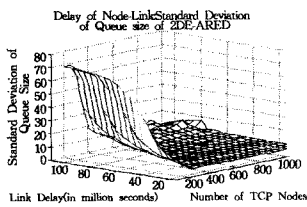


图 12 2DE-ARED 算法的队长标准差

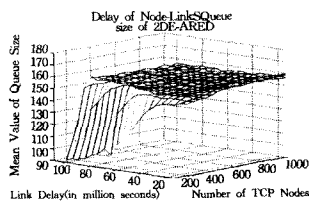


图 13 2DE-ARED 算法的队长平均值

图 14 和图 15 分别给出 PID 算法的瞬时队列长度的标准差和队长平均值。对比图 12 和图 14, 相比之下, 2DE-ARED 能够较为有效地降低瞬时队列长度的波动性, 并且在瞬时队列长度的均值方面也有比较好的表现。

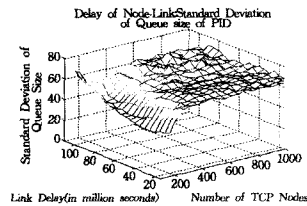


图 14 PID 控制器的队长标准差

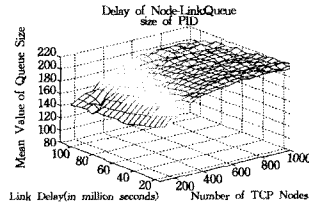


图 15 PID 控制器的队长平均值

**结束语** 本文首先介绍二阶差分方程的特点, 选择连续域-离散化的方法来设计二阶控制器, 给出二阶传递函数形式。然后根据控制器的设计目标来确定主要的参数, 并确定系统的二阶传递函数, 将传递函数转换为离散域的脉冲传递函数, 求得相应的差分方程。最后在 NS2 上通过仿真实验

验证了算法的正确性和有效性。

## 参考文献

- [1] Jacobson V. Congestion avoidance and control [J]. ACM SIGCOMM Computer Communication Review, 1995, 25(1): 157-187
- [2] Floyd S, Jacobson V. Random early detection gateways for congestion avoidance [J]. IEEE/ACM Transactions on Networking, 1993, 1(4): 397-413
- [3] 戴航, 慕德俊, 王林, 等. Internet 拥塞控制系统在不同源控制算法作用下的资源竞争分析 [J]. 计算机科学, 2009, 36(11): 40-42
- [4] Floyd S, Gummadi R, Shenker S. Adaptive RED: An algorithm for increasing the robustness of RED's active queue management [OL]. <http://www.icir.org/floyd/papers/adaptiveRed.pdf>, August 2001
- [5] Holot C, Misra V, Towsley D, et al. A control theoretic analysis of RED [C] // Proceedings of IEEE INFOCOM, 2001, 3: 1510-1519
- [6] Zheng B, Atiquzzaman M. A framework to determine bounds of maximum loss rate parameter of RED queue for next generation routers [J]. Journal of Network and Computer Applications, 2008, 31(4): 429-445
- [7] Zamani M, Karimi-Ghartemai M, Sadati N, et al. Design of a fractional order PID controller for an AVR using particle swarm optimization [J]. Control Engineering Practice, 2009, 17(12): 1380-1387
- [8] 陈晓龙, 章云, 刘治. 多 TCP 协议多链路端拥塞控制特性研究 [J]. 计算机科学, 2009, 36(5): 88-91

(上接第 13 页)

- [42] Bate I, Hawkins R, McDermid J. A contract-based approach to designing safe systems [C] // Proceedings of the 8th Australian Workshop on Safety Critical Systems and Software (SCS'03). Darlinghurst: Australian Computer Society, 2003, 33: 25-36
- [43] Kornecki A J. Assessment of software safety via catastrophic events coverage [C] // Proceedings of the 21st IASTED International Multi-conference Applied Informatics (AI 2003). Innsbruck (Austria): ACTA Press, 2003: 1139-1144
- [44] Wu W, Kelly T. Failure modeling in software architecture design for safety [C] // Proceedings of the 2005 Workshop on Architecturing Dependable Systems. New York: ACM, 2005: 1-7
- [45] 杨捷. 安全软件体系结构的形式化方法研究 [D]. 武汉: 武汉大学, 2004
- [46] Bieber P, Bougnol C, Castel C, et al. Safety assessment with AltaRica-lessons learnt based on two aircraft system studies [C] // 18th IFIP World Computer Congress. Toulouse: Springer Verlag, 2004, 156: 505-510
- [47] Fenelon P, McDermid J. An integrated toolset for software safety analysis [J]. Journal of Systems and Software, 1993, 21(3): 279-290
- [48] Fenelon P, McDermid J A, Nicholson M, et al. Towards integrated safety analysis and design [J]. ACM SIGAPP Applied Computing Review, 1994, 2(1): 21-32
- [49] International Electrotechnical Commission. IEC 61508—1999 Fundamental safety of electrical/electronic/programmable electronic safety related systems [S]. Geneva: International Electrotechnical Commission, 1999
- [50] Kelly T P. Arguing safety—a systematic approach to managing safety cases [D]. Heslington: University of York, 1999
- [51] Greenwell W S, Knight J C, Holloway C M, et al. A taxonomy of fallacies in system safety arguments [C] // Proceedings of the 24th International System Safety Conference. Albuquerque: System Safety Society, 2006
- [52] Nelson S. Certification processes for safety-critical and mission-critical aerospace software [R]. NASA/CR-2003-212806. Washington DC: National Aeronautics and Space Administration, 2003
- [53] NATO. STANAG 4626—2005 Modular and open avionics architecture [S]. Brussels: Military Agency for Standardization, 2005
- [54] Rushby J. Modular certification [R]. NASA/CR-2002-212130. Washington DC: National Aeronautics and Space Administration, 2002
- [55] Industrial Avionics Working Group. Modular Software Safety Case Process [R]. IAWG-AJT-301. Leatherhead: Industrial Avionics Working Group, 2007
- [56] Fenn J L, Hawkins R D, Kelly T P, et al. The who, where, how, why and when of modular and incremental certification [C] // 2nd IET International Conference on System Safety. London: IET Software, 2007, CP532: 135-140
- [57] Ministry of Defence. DEF-STAN 00-56 draft issue 3—2004 Safety Management Requirements for Defence Systems [S]. London: Ministry of Defence, 2004
- [58] Radio Technical Commission for Aeronautics, Inc. RTCA DO-297—2005 Integrated Modular Avionics (IMA) Development Guidance and Certification Considerations [S]. Washington DC: Radio Technical Commission for Aeronautics, Inc., 2005
- [59] Alberico D, Bozarth J, Brown M, et al. Software system safety handbook [M]. Patuxent River (USA): Joint Services Software Safety Committee, 1999