

AFDX 交换机警管算法研究及实现

张栩培, 贺占庄, 王瑞晓, 李龙飞

(西安微电子技术研究所, 陕西 西安 710054)

摘 要: AFDX 交换机主要采用流量警管技术来确保网络传输过程中每条虚链路的数据按照既定的速度传输. 据此实现对 AFDX 网络的流量控制, 从而保证 AFDX 网络数据传输的确定性、可靠性. 基于 ARINC664 规范, 提出了一种 AFDX 交换机警管算法, 完成了该算法的设计实现, 并通过仿真实验证明了该设计具有良好的流量控制效果.

关键词: AFDX 网络; 警管算法; 虚链路; 带宽分配间隙

中图分类号: TP31

文献标识码: A

文章编号: 1000—7180(2016)01—0114—05

Research and Implementation of AFDX Switch Policing Algorithm

ZHANG Xu-pei, HE Zhan-zhuang, WANG Rui-xiao, LI Long-fei

(Xi'an Microelectronics Technology Institute, Xi'an 710054, China)

Abstract: The Switch of AFDX adopted policing technology to ensure that the data in every virtual link transmit as a fixed speed. On these grounds it can achieve the flow control of AFDX, and guarantee the determinacy and reliability for AFDX. According to the ARINC664 specification, a policing algorithm for the AFDX switch is given out, and the design of the algorithm was completed. The simulation experiments show that the design has good effect on flow control.

Key words: AFDX network; Policing Algorithm; Virtual Link; Bandwidth Assignment Gap

1 引言

随着航空电子技术的飞速发展, 现有的航空电子系统通信网络无论在传输性能、服务质量, 还是在可靠性等方面均无法满足其发展的要求. 因此研究一种新型的航空电子通信网络势在必行. ARINC 644 规范是航空电子技术委员会审议通过的新一代机载以太网标准, 该规范的第七部分定义了一种新的适用于航电系统间信息传输的交换式网络, 即航空全双工交换式以太网 AFDX (Avionic Full-Duplex Switched Ethernet).

AFDX 网络基于标准的 IEEE802.3 协议, 在现有的商用以太网的技术特点上增加了确定性定时机制, 以及可靠的数据传输机制以适应复杂的航空电子系统应用.

AFDX 网络的确定性通过具有有限带宽和确定的帧传输间隔的虚拟链路技术实现, 通常情况下

AFDX 网络中各个虚链路都根据系统配置的带宽分配间隙(BAG)来控制数据发送的速率, 从而实现流量控制. 为了实现这一功能, AFDX 交换机需要提供一种流量控制技术来保证虚链路传输的数据不会超过设定的流量, 因此警管功能是 AFDX 网络所特有的功能, 它是保证 AFDX 网络数据传输延时确定性和可靠性的基础. 本文基于 ARINC644 规范, 提出了一种警管算法, 完成算法的设计实现, 并进行综合与验证.

2 警管算法的研究

AFDX 交换机警管模块的功能是进行分组的流量控制, 通过调整发往交换机接受端系统帧的速度, 从而避免因为大量的数据帧传输而导致网络拥塞. 现有的网络流量控制算法主要有两种: 漏桶算法和令牌桶算法.

漏桶算法的主要特点是可以将一个不均匀分组

流变成网络上的一个均匀分组流,从而降低网络拥塞的概率.而令牌桶算法是按照用户制定好的速度向桶中放置令牌,当需要发送数据帧时检查令牌桶中的令牌数是否足够,如果令牌数够用则发送数据,并且从令牌桶中减去相应数量的令牌,因此令牌桶算法能够接受一定时间内的突发数据,比漏桶算法更具灵活性^[1].漏桶算法及令牌桶算法的原理如下所示.

在 AFDX 网络中并不需要通过警管模块进行流量整形,而只是需要将速度过快的帧丢弃,并且由于调度所产生的抖动会使得实际发送数据帧的间隔不能严格符合 VL 的带宽分配间隙,基于上述考虑本文采用令牌桶算法作为本文设计的警管算法的核心.

AFDX 网络警管算法的基础是虚拟链路(VL),警管模块需要根据配置表中的 VL 相关参数对相应的 VL 传输业务流进行流量控制.下面以一条 VL 为例对本文的警管算法进行详细介绍.

首先我们通过配置表获得第 i 条虚链路 VL_i 的相关参数:

- (1) BAG_i : VL_i 的带宽分配间隙;
- (2) $J_{i,switch}$: VL_i 所允许的抖动;
- (3) S_i^{max} : VL_i 所能传输的最大帧长;
- (4) AC_i : VL_i 的信用量.(相当于令牌桶算法中的令牌桶容量)

AC_i 的初始值为

$$AC_i^{max} = S_i^{max} \cdot (1 + \frac{J_{i,switch}}{BAG_i}) \quad (1)$$

式中,带宽分配间隙(BAG)和抖动 $J_{i,switch}$ 均以时间为单位, S_i^{max} 及 AC_i 均已字节为单位.

由上述公式可知, AC_i 是以 $\frac{S_i^{max}}{BAG_i}$ 的比率随时间增加,其最大值即为 $S_i^{max} \cdot (1 + \frac{J_{i,switch}}{BAG_i})$.

ARINC644 规范中为 AFDX 交换机提供了两种警管策略,即基于字节的警管策略和基于帧的警管策略.^[2]

(1) 基于字节的警管策略:如果 AC_i 大于等于帧长 S (字节)时,该帧通过警管并被接收,并且 AC_i 值要减去该帧长 S ,当 AC_i 小于帧长 S (字节)时,该帧被丢弃,相应的 MIB 信息被更新,且 AC_i 值保持不变;

(2) 基于帧的警管策略:如果 AC_i 大于等于最大帧长 S_i^{max} (字节)时,该帧通过警管并被接收,并且 AC_i 值要减去 S_i^{max} ,当 AC_i 小于最大帧长 S_i^{max} (字节)

时,该帧被丢弃,相应的 MIB 信息被更新,且 AC_i 值保持不变.

由此可以看出这两种策略的区别在于 AC_i 每次减少的是数据帧的实际帧长 S_i 还是最长帧长 S_i^{max} . 由于基于字节的警管算法中实际帧长 S_i 是一个变量,在设计实现方面较为复杂.因此本文采用基于帧的警管策略.

下面我们选取两个特定的情况对基于帧的警管过程进行简单的描述.

(1) 当 $J_{i,switch} = 0$ 时: AC_i 的初始值为 AC_i^{max} , 图 1 为该警管过程示意图.

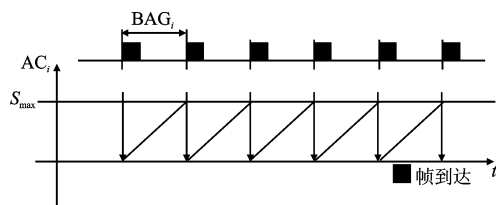


图 1 抖动为 0 时的警管过程

(2) 当 $J_{i,switch} = BAG/2$ 时: AC_i 的初始值为 1.5 倍的,图 2 为该警管过程示意图.

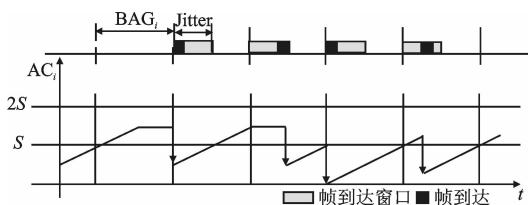


图 2 抖动为 BAG/2 时的警管过程

值得注意的是 ARINC644 规范中特别指出未通过过滤功能的无效帧(如 CRC 校验失败)不会使 AC_i 的值减少.

3 警管算法的设计实现

本设计针对的 AFDX 交换机共有 16 个端口,其中每个端口均支持 256 条虚链路,各虚链路的 BAG_i 和 $J_{i,switch}$ 均可通过配置表配置.

根据警管算法及警管模块所应实现的功能,本文将警管模块的设计实现划分为 4 个子模块:警管状态机、AC RAM、警管控制和 AC RAM 访问控制.

3.1 警管状态机

根据上述分析,本设计中警管状态机分为 8 个状态:空闲状态、存定时值状态、获取 AC 值状态、计算 AC 状态、比较 AC 状态、更新 AC 状态、警管成功状态、警管失败状态.状态间的转移如图 3 所示.

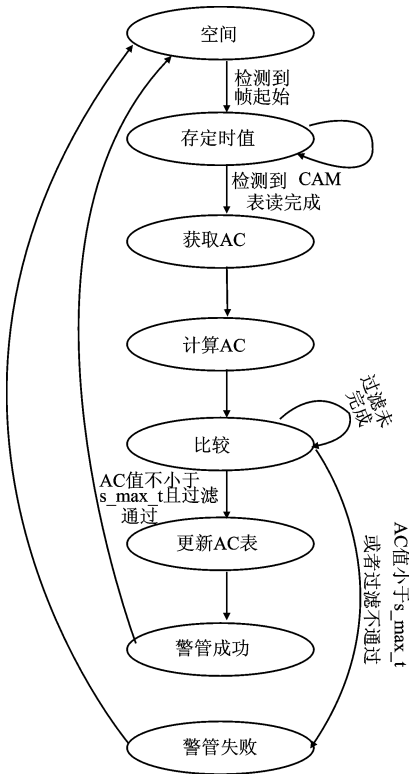


图 3 警管状态机状态间转移

(1)空闲状态:在该状态检测到帧起始标识有效后,跳转到存定时值状态;否则,在空闲状态等待。

(2)存定时值状态:在该状态,检测到虚链路配置表读完成标识有效后,跳转到获取 AC 状态;否则,在该状态等待. 该状态下获得数据帧到达警管模块的时刻。

(3)获取 AC 状态:该状态有两个周期,该状态结束后直接跳转至计算 AC 状态. 在该状态的第一个周期产生 AC 表 RAM 读使能信号. 第二个周期对从 AC 表中读取的数据进行解析得到上一帧到达时刻及剩余信用量. 同时读取虚链路的抖动、带宽分配间隙等配置信息. 根据上述警管算法,可以看到实现过程中要进行乘法运算,为了简化设计,需要将算法改进. 警管的重要参数信用量 AC_i 的值随时间变化,因此可以将相关参数 AC_i 、 $S_i^{\max}$ 、 $AC_i^{\max}$ 转换为与时间相关的参数^[3-4]. 将 AC_i 、 $S_i^{\max}$ 、 $AC_i^{\max}$ 均除以 $S_i^{\max}/(BAG_i)$,就可以将 $AC_i^{\max}$ 的值转化为 $BAG_i + J_{i,switch}$; $S_i^{\max}$ 为 BAG_i ; AC_i 按 t 的速率自增. 这样所有参数就都转换成时间单位. 因此就可以利用全局时钟来实现警管功能。

(4)计算 AC 状态:该状态结束后直接跳转至比较 AC 状态. 在该状态下利用获取 AC 状态中所得到的上一帧到达时刻,剩余信用量以及当前帧到达

时刻计算出当前信用量 = 当前帧到达时刻 - 上一帧到达时刻 + 剩余信用量。

(5)比较 AC 状态:该状态下等待过滤通过标识,若过滤通过且当前信用量大于等于 $S_i^{\max}$,则跳转至更新 AC 状态;否则,跳转至警管失败状态。

在计算 AC 及比较 AC 状态下,警管模块对输入的数据帧进行警管处理,警管模块对数据帧进行警管的具体流程如图 5 所示,图中 para 为当前信用量,Time_now 为当前帧到达时刻,Time 为上一帧到达时刻,AC 为剩余信用量。

若数据帧通过警管则在比较 AC 状态下产生对 AC 表的写使能信号,并将当前信用量和当前帧到达时刻进行拼接;若数据帧未通过警管或未通过过滤则丢弃该帧,提供丢弃控制信号给 MIB 统计,该分组接收完毕后,清空 MAC 内部的接收 fifo。

(6)更新 AC 状态:该状态结束后直接跳转至警管成功状态. 在该状态下将警管成功后拼接好的数据写入到 AC 表中。

(7)警管成功状态:该状态结束后跳转至空闲状态. 在该状态中产生警管成功标识。

(8)警管失败状态:该状态结束后跳转到空闲状态,在该状态产生警管失败标识。

3.2 AC 表 RAM

本设计中使用的 AC 表 RAM 为 quartus 生成的 RAM,深度为 256,宽度为 64 bit,使用的 FAGA 器件为 stratix 3. 每个 RAM 单元,低 16 位用来存放剩余信用量 AC,高 48 位用来存放上一帧到达时刻。

3.3 警管控制

警管控制逻辑根据警管状态机的状态产生相应的控制信号。

3.4 AC 表 RAM 访问控制控制

AC 表 RAM 访问控制逻辑,根据警管状态机的不同状态警管模块信号描述如表 1 所示。

信号名称	信号描述
sys_clk_i	输入,系统全局时钟
sys_rst_n_i	输入,系统复位
cur_ram_adr_i[7:0]	输入,RAM 当前访问地址
cur_fra_sof_i	输入,当前分组到达标识
cur_vl_fil_ok_i	输入,过滤成功标识
cnt_plic_now_i[47:0]	输入,1/32 ms 时钟计数器的当前值
cur_vl_plic_finish_o	输出,当前 VL 警管完成标识
cur_vl_plic_ok_o	输出,当前 VL 的警管通过标识.
cur_vl_plic_fail_o	输出,当前 VL 未通过警管

4 仿真验证

本设计的验证采用 System Verilog 语言, 该语言是 Verilog 的升级扩展, 它可以从较高的层次搭建验证平台, 因此具有较高的功能覆盖率, 可以使得验证工作尽快收敛, 大大提高了调试程序的效率. 下

面我们主要针对警管模块的三方面的功能进行了仿真验证.

(1)VL_i成功通过警管模块:该验证中我们将 VL_i的 BAG_i配置为 4 ms, J_{i,switch}配置为 2 ms. 验证结果如图 4 所示.

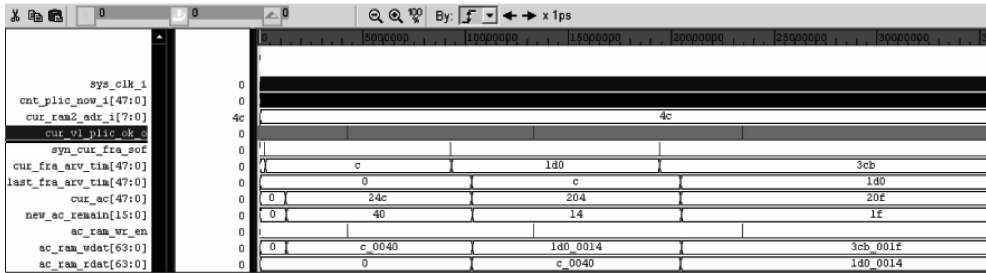


图 4 VL_i成功通过警管

由验证结果可以看出, VL_i中的数据帧传输过程中, 只要前后两帧的到达时间间隔满足要求, 即当前 VL_i的当前信用量大于等于最大帧长, 则数据帧通过警管.

(2)VL_i未通过警管模块:该验证中我们将第一帧和第二帧的发送间隔缩短, 验证中 VL_i的 BAG_i配置为 4 ms, J_{i,switch}配置为 2 ms 结果如图 5 所示.

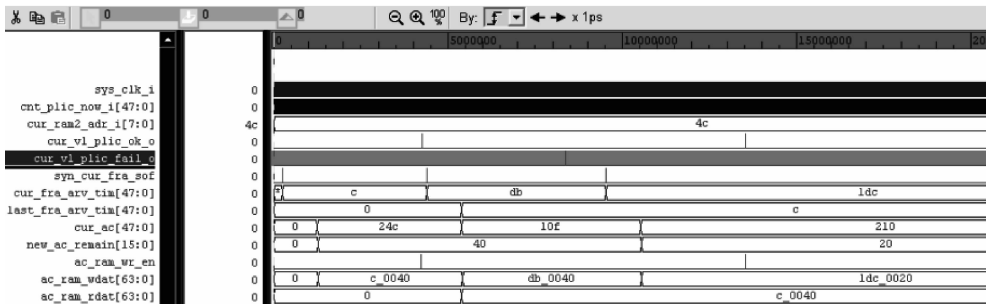


图 5 VL_i未通过警管

由验证结果可以看出, 当 VL_i的当前信用量小于最大帧长时, 数据帧未通过警管, 丢弃该帧, 同时警管模块向 MIB 提供丢弃控制信号.

(3)VL_i未通过过滤模块:该验证中我们采用与上述验证相同的条件, 但在该验证中我们假定第二帧数据未通过过滤模块. 验证结果如图 6 所示.

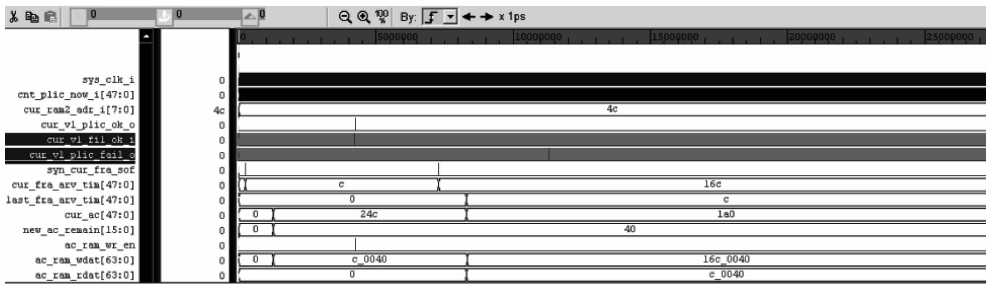


图 6 VL_i未通过过滤

由验证结果可以看出, 当 VL_i未通过过滤模块时, 当前数据帧不能通过警管功能模块, 丢弃该帧, 同时警管模块向 MIB 提供丢弃控制信号.

5 结束语

本文在 ARINC 644 规范的基础上, 提出了一种

AFDX 交换机警管模块的警管算法,并基于该算法对 AFDX 警管模块进行硬件设计.通过仿真实验证明了该算法可以满足 ARINC 664 规范中对 AFDX 网络流量控制的要求^[5],对进入 AFDX 交换机内的数据帧进行警管,具有良好的流量控制功能.本算法已经在相关的 AFDX 交换机设计中得以应用,接下来我们将继续研究该警管算法在不同网络交换系统中实现的可能性.

参考文献:

[1] (美)AndrewS Tanenbaum. 计算机网络(第四版)[M]. 潘爱民,译. 北京:清华大学出版社,2004.

[2] Roy Courtney. Circulation prior to adoption consideration Draft3 of project paper 644: Aircraft data network, part 7-avionics full duplex switched ethernet (AFDX) network, september 13, 2004 [S]. Stockholm; AEEC. 2004.

[3] Teresa Schuster, Dinesh Verma. Networking concepts

comparison for avionics architecture[C]//IEEE/AIAA 27th Digital Avionics Systems Conference, st. paul, μ N;IEEE,2008:1-11.

[4] 刘增基,鲍民权,邱智亮. 交换原理与技术[M]. 北京:人民邮电出版社,2007.

[5] 王红春,张涛,牛文生. 面向 AFDX 网络的交换机流量管制算法[J]. 西北工业大学学报, 2012, 30 (02): 301-304.

作者简介:

张栩培 男,(1991-),硕士研究生. 研究方向为计算机控制技术. E-mail:panfeng1102@163. com.

贺占庄 男,(1962-),博士,研究员,博士生导师. 研究方向为计算机系统结构、SoC 和 IP 核设计.

王瑞晓 女,(1985-),工程师. 研究方向为计算机系统结构.

李龙飞 男,(1988-),博士研究生. 研究方向为计算机系统结构.

(上接第 113 页)

[6] 陈智罡,王箭,宋新霞. 全同态加密研究[J]. 计算机应用研究,2014,30(4):35-37.

[7] Gentry C. Fully homomorphic encryption using ideal lattices [C] // Proceedings of the forty-first annual ACM Sysposium on Theory of Computing. NY, USA; IEEE,2009:169-178.

[8] Kantarcioglu, M Kardes. Privacy-preserving data mining in the malicious model[C]// Proceeding of the Seventh IEEE International Data Uining Wo9rkshops. O-

maha,NE;IEEE, 2007:717-722.

作者简介:

李娟 女,(1975-),硕士,副教授. 研究方向为云计算、网络安全、社交网络与隐私保护.

马飞 男,(1976-),博士,副教授. 研究方向为数据挖掘、云计算,社交网络分析与隐私保护.

E-mail:feixiangflying33@nxu. edu. cn.