

AFDX 网络配置测试技术研究

陈芳¹, 吕广喆², 林伟³, 侯溪溪¹, 马春燕³

(1.中国航空无线电电子研究所,上海 200233; 2.航空工业西安航空计算技术研究所,陕西西安 710068;
3.西北工业大学 软件学院,陕西 西安 710072)

摘 要:终端系统作为 AFDX 网络数据传输的关键部分,其优劣决定着整个 AFDX 网络的通信性能。为了测试验证端系统功能,评估 AFDX 网络的传输性能,提出了一种 AFDX 网络配置测试方法。设计端系统的配置测试规范,建立端系统测试总体框架,对端系统的测试响应机制进行研究,规范化测试流程,生成的配置文件作为算法设计的输入信息。设计激励数据生成算法,建立激励数据生成框架,对数据发送的约束条件进行形式化定义,利用 SMT 求解器,生成激励数据发送时刻表。激励数据发送时刻表作为输入,利用 TestCenter 网络测试平台进行仿真实验,实验结果证明该方法能模拟当前 AFDX 网络配置在高负载数据流下的有效性。

关 键 词:AFDX 网络;端系统;虚拟链路;自动化测试

中图分类号:TP393

文献标志码:A

文章编号:1000-2758(2024)03-0514-07

为了实现航空电子系统间大容量高速的数据交换^[1],以波音公司与空客公司为代表的航空设备制造商共同组织研制新一代航空总线标准,在此标准中,定义了一种新型网络结构^[2-3]——AFDX(航空电子全双工交换式以太网, avionics full duplex switched ethernet)网络。AFDX 网络在传统的以太网技术基础上,通过引入虚链路、双冗余管理等技术,使之满足航空电子系统通信对高带宽、低延迟、强实时性、确定性的通信服务要求。

AFDX 包括端系统、交换机和虚拟链路 3 部分。AFDX 端系统作为航空电子子系统与 AFDX 交换机之间的接口,负责发送和接收来自电子系统的信息,建立各子系统之间的连接,保证整个综合电子系统之间数据交换的安全性和可靠性。终端系统的性能优劣决定着整个 AFDX 网络的安全性,为了保证端系统在 AFDX 网络中正常工作,对终端的功能和性能进行验证测试显得尤为重要。

本文所提出的机载网络系统下 AFDX 网络配置

测试技术研究,主要目的是对 AFDX 网络中端系统进行测试,对终端的功能和性能进行评估,提升航空电子系统可靠性和安全性。本文贡献如下:

1) 设计 AFDX 端系统配置测试规范。研究 AFDX 终端功能,对终端的测试内容进行确认,建立测试总体框架,并对端系统的配置文件格式进行设计。

2) 研究激励数据生成算法。建立激励数据生成架构,对数据发送的约束条件进行形式化定义,通过 SMT 求解器,设计并生成激励数据发送时刻表,保证消息按照时间间隔无冲突发送。

3) 仿真实验验证及分析。本仿真实验环境是 TestCenter 网络测试平台,利用激励数据发送时刻表作为输入,模拟当前网络高负载的流量调度。

1 相关工作

在航空电子领域,机载网络要求的是高实时性、高可靠性、高数据传输率和低成本。检测 AFDX 端系统功能和性能是 AFDX 网络领域的研究热点。本节将从 AFDX 端系统测试介绍国内外研究现状。

对 AFDX 网络的研究工作中最重要的是对 AFDX 网络中的端系统网络节点进行测试和验证,

收稿日期:2023-05-16

基金项目:航空科学基金(20185853038,201907053004)资助

作者简介:陈芳(1987—),高级工程师

通信作者:马春燕(1978—),副教授

e-mail:machunyan@nwpu.edu.cn

以判断端系统能否正常工作^[4]。AFDX 端系统测试技术需要从测试需求分析、测试方案设计以及测试工具研发等方面进行研究,建立完整的测试体系^[5]。

分析研究 AFDX 端系统的功能特性,总结终端系统的测试需求。Wang 等^[6-7]对终端的应用协议展开研究,并且从端系统基本属性、虚拟链路、端口通信和运行稳定性方面进行了测试需求分析;Wang 团队^[8]通过研究 ARINC 664 第七部分,根据协议针对故障注入和数据完整性检测提出测试方法。

根据端系统测试需求,制定测试方案。苗佳旺等^[9]从通信测试、网络监控和故障注入三方面对 AFDX 端系统及其通信端口制定测试方法。

研制端系统测试仿真工具,以具体实验评估 AFDX 网络的功能与性能。文献[10]设计了基于 WinPcap 的 ARINC664 总线网络测试系统,硬件基于普通 PC 机和高性能以太网卡,并且在 WinPCap 网络抓包工具基础上二次开发完成,实现对整个网络性能参数的分析功能;文献[11]通过编制 AFDX 网络收发测试程序,采用商用的 AFDX 仿真板卡和 AFDX 网络交换机,实现远程对 AFDX 端系统设备的功能测试和性能测试。

基于以上研究现状可知,当前国内外对于端系统测试领域存在以下问题:测试需求难以覆盖端系统的重要功能点;测试方案的可实施性较差,测试过程中的指令信息和操作流程不够完善,无法实时获取端系统的运行状态,测试用例的执行步骤缺乏必要细节;端系统配置和测试执行过程存在重复和繁琐的人力操作,测试工具与硬件板卡之间耦合性较高。

满足性模理论求解器是软件工程研究和实践中的重要基础设施。SMT 求解器广泛用于各种重要的软件工程技术,例如符号执行^[12]、程序合成等。SMT 求解器用于验证一阶谓词逻辑的公式。如果给定的公式是可满足的,那么 SMT 求解器给出自由变量的替换,使公式为真。

本文基于激励数据生成算法,建立激励数据生成架构,对数据发送的约束条件进行形式化定义,通过 SMT 求解器,生成激励数据发送时刻表,保证消息按照时间间隔无冲突发送。

2 AFDX 端系统配置测试规范设计

2.1 AFDX 端系统配置文件设计

终端的配置信息项通过 XML 文件的形式进行保存,对终端的配置信息项给出详细的设计方案。

终端作为 XML 文件的根节点,所包含的属性信息描述如表 1 所示。

表 1 终端属性定义表

标签名	标签等级	包含属性	描述
EndSystemTable	一级	DomainID	端系统唯一标识符
		Name	终端名称
		Speed	数据传输速率

发送链路表示用于数据发送的链路,定义为终端的下一级节点,包含的属性信息描述如表 2 所示。

表 2 发送链路属性定义表

标签名	标签等级	包含属性	描述
SndVL	二级	VLID	发送链路唯一标识
		BAG	数据发送最小间隔
		SubvlCount	包含子链路数目
		$L_{\max}$	允许发送最大帧长
		Redundancy	是否开启冗余模式
		NetSelect	数据发送子网类型
		Jitter	数据发送间隔 允许抖动时间

接收链路表示用于数据接收的链路,定义为终端的下一级节点,包含的属性信息描述如表 3 所示。

表 3 接收链路属性定义表

标签名	标签等级	包含属性	描述
RevVL	二级	VLID	接收链路唯一标识
		IntegrityCheck	数据完整性检查
		Redundancy	是否开启冗余模式
		SkewMax	接收数据帧所允许的 最大时间间隔

发送端口用于数据发送,定义为终端的下一级节点,包含的属性信息描述如表 4 所示。

表 4 发送端口属性定义表

标签名	标签等级	包含属性	描述
SendPort	二级	PortId	发送端口唯一标识
		TxVlId	端口所属链路号
		TxSubVlId	使用子链路序列号
		SrcUDP	源 UDP 端口号
		DstUDP	目的 UDP 端口号
		DstIP	目的 IP 地址
		PortType	使用端口类型, 包括采样、队列和 SAP
		MessageLength	传输消息长度

接收端口用于数据接收,定义为终端的下一级节点,包含的属性信息描述如表 5 所示。

表 5 接收端口属性定义表

标签名	标签等级	包含属性	描述
ReceivePort	二级	PortId	接收端口唯一标识
		RxVlId	端口所属链路号
		RxSubVlId	使用子链路序列号
		SrcUDP	源 UDP 端口号
		DstUDP	目的 UDP 端口号
		SrcIP	源 IP 地址
		DstIP	目的 IP 地址
		PortType	端口类型,包括 采样、队列和 SAP
		MessageLength	传输消息长度

2.2 AFDX 网络调度模型的建立

AFDX 网络采用星形拓扑网络结构,网络通信节点包括端系统和交换机,端系统之间通过交换机连接,定义 $N = \{E_1, E_2, SW_1, SW_2, \cdots, E_n, SW_n\}$, 表示对于一个 AFDX 网络中,存在 n 个端系统相互连接。

2.3 基于 SMT 求解器激励数据生成架构

通过对消息长度和发送时间间隔进行约束,实现对板卡激励数据的规整,保证同一时刻链路只能发送一条消息,消息发送的时间窗口不出现重叠,并且保证消息的长度和发送时间间隔满足链路的

限制。

定义 $M = \{v_1, v_2, \cdots, v_m, \cdots, v_n\}, 0 < m < n$,表示端系统中包含 n 条虚拟链路。虚拟链路的属性为 $\langle \text{VLID}, l_{\max}^i, \text{BAG}_i \rangle$,其中 VLID 代表虚拟链路号,链路号确定且唯一, $l_{\max}^i$ 代表第 i 条链路允许发送消息的最大长度, BAG_i 表示第 i 条链路数据流发送的最短间隔,链路在一个 BAG 中只能发送一个消息。

约束建立如下:

1) BAG 约束

端系统配置信息中对每一条链路都分配一个 BAG 值,每条链路在一个 BAG 时间段内只能发送一帧数据,要求激励数据发送时间间隔和链路的 BAG 保持一致,并且激励数据在一个 BAG 间隔只能发送一次。要求 BAG 范围在 1~128 ms

2) 帧长约束

当端系统发送数据时,提前对数据帧长进行检测,如果数据长度大于规定的最大帧长,端系统将会把数据划分成较小的数据片封装之后按照相应的 BAG 发送到物理链路上;当数据帧长度小于最小帧长时,就需要对数据帧进行填充,使其符合帧长要求,因为当数据帧小于最小帧长时,容易造成丢包的情况发生。故在进行消息包规整时,需要满足的帧长约束的数学表达式为

$$L_{\min} \leq V_{\text{frame}} \leq L_{\max}$$

3) 最大延迟约束

主机端应用程序传输数据较长需要分片的数据时,这条 VL 需要处理的其他数据将会被延迟发送。对于传输数据的 VL,如果 $P - 1$ 个帧已被处理,第 P 帧的最大延迟为

$$\max\text{Latency}(\text{frame}P) \leq P * \text{BAG}_i + \max\text{Jitter} + \text{Thnological\_Latency\_intrasmission}$$

故最大延迟约束的数学表达式为

$$V\text{Latency}(\text{frame}P) \leq \max\text{Latency}(\text{frame}P)$$

4) 最大允许抖动约束

在 ARINC664 规范中要求,在终端节点系统输出的虚连接传输抖动最大值要求满足

$$\begin{aligned} \max\text{Jitter} &\leq 40 \mu\text{s} + \\ &\frac{\sum (20 \text{ bytes} + L_{\max} \text{ byte}) * 8 \text{ bits/byte}}{N_{\text{bw}} \text{ bits/s}} \\ \max\text{Jitter} &\leq 500 \mu\text{s} \end{aligned}$$

其中,最大抖动的单位为 μs 。 N_{bw} 是介质的带宽,以 bits/s 为单位; $L_{\max}$ 以 byte 为单位。针对全网

络,抖动的上限为 500 μs 。选择端系统配置参数,使抖动尽量小于 500 μs 。一个 BAG 时间段内一个数据帧等待的时间不能超过 Jitter,如果超过 Jitter,该 VL 这个 BAG 内将不能再继续发送数据。

5) 冗余约束
端系统通过多个独立冗余的网络进行通信,这样可以避免某个网络由于连接或交换出现的故障,端系统发送数据帧时,对于冗余网络需要满足:

- (1) 冗余 VL 数据需要通过网络 A 和网络 B 同时发送;
 - (2) 非冗余 VL 数据只通过网络 A 或网络 B 发送。
- 6) 无冲突约束

发送激励数据的时间占用不能出现重叠,即同一时刻只能发送一条激励数据;同一条链路上激励数据发送时间窗口不能发生交集,即一条链路在当前激励数据没发送完成之前,不会发送其他数据。

2.4 基于 SMT 求解器激励数据发送时刻表生成流程设计

基于 SMT 求解器消息发送时刻表算法的思想是通过将 AFDX 网络中数据传输的属性和特征抽象为数学约束,通过 SMT 求解器计算,生成激励数据的发送时刻表。算法执行步骤如图 1 所示。首先,读取终端配置文件,对配置文件进行处理,获取链路属性参数,并设计数据结构存储虚拟链路信息;其次,终端配置求解器为终端的链路分配求解参数,建立约束条件,将表达式和链路信息一起传入求解器求解;最后对求解的参数列表进行分析整理,生成数据发送时刻表。时刻表记录链路发送消息的规划方案,作为算法的输出内容。

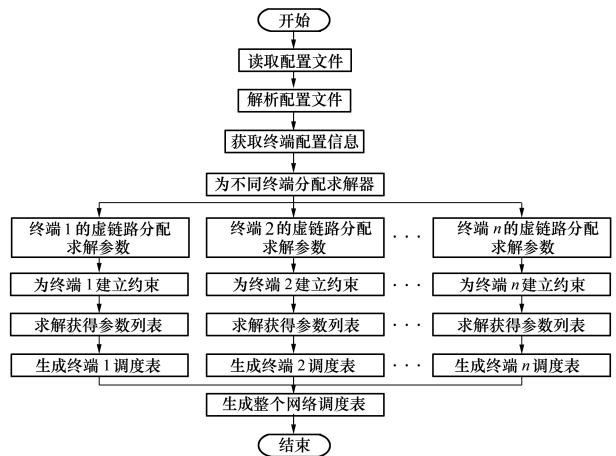


图 1 算法流程示意图

1) 终端配置文件的预处理
读取终端 XML 配置文件,获取端系统发送链路的配置信息,设计数据结构进行存储,数据结构用 VL 表示,VL 数据结构属性描述如表 6 所示。

表 6 VL 属性描述表

组成部分	数据类型	描述
VLID	字符串类型	链路号,标识唯一虚拟链路
ESystem	字符串类型	端系统名称,标识当前链路所属终端
$L_{\max}$	整型	消息最大长度,表示当前链路允许发送消息的最大长度
BAG	浮点型	带宽分配间隔,表示同一条链路发送 2 个连续帧之间的最小间隔

对 XML 文件进行处理,定位文件中的 SndVL 标签,获取标签下的节点信息,以链路 ID 作为标识,提取对应的链路信息,最终获取链路 ID 与链路信息的映射关系;定位文件中的 EndSystem 标签,获取该标签下的 SndVL 节点信息,以终端名称作为标识,提取该终端下发送链路信息,并生成链路集合,最终获取端系统与其包含的所有链路集合的映射关系;定位文件中的 EndSystem 标签,以终端名称作为标识,记录终端数据发送速率,最终获取端系统与其数据发送速率的映射关系。

2) 求解器计算
将约束条件抽象成 SMT 表达式,传入到求解器中进行计算,从基本约束,无冲突约束,传输周期性约束 3 个方面设计约束条件。约束条件建立如算法 1 所示。

算法 1 Constraint establishment

输入: Z3 is the solver, VLList<VL> is the list of virtual links, Speed is the link sending rate, and TIME is the number of messages sent

输出:

```
1. for(i=0 to VLList.size()) do
2.   for(j=1 to TIME) do
3.     dispatch_point←
         Slover.getVariable( VLList.get(i).VLID)
4.     Slover.add( dispatch_point≥0)
5.   end for
```

```
6.  end for
7.  for( $k_1 = 1$  to TIME) do
8.      for( $i = 0$  to ViList.size() $-1$ ) do
9.          dispatch_point1 $\leftarrow$ 
              Slover.getVariable( ViList.get( $i$ ).VLID)
10.         for( $j = i+1$  to ViList.size() $)$  do
11.             for( $k_2 = 1$  to TIME) do
12.                 dispatch_point2 $\leftarrow$ 
                     Slover.getVariable(
                         ViList.get( $i$ ).VLID
                    )
13.                 Slover.add(
                     dispatch_point>
                     dispatch_point1+
                     ViList.get( $i$ ). $L_{\max}$ /Speed ||
                     dispatch_point>
                     dispatch_point1+
                     ViList.get( $j$ ). $L_{\max}$ /Speed
                    )
14.             end for
15.         end for
16.     end for
17. end for
18. for( $i = 0$  to ViList.size() $)$  do
19.     for ( $k = 1$  to TIME) do
20.         dispatch_point1 $\leftarrow$ 
                Slover.getVariable(
                    ViList.get( $i$ ).VLID+
                    toString( $k$ )
                )
21.         dispatch_point2 $\leftarrow$ 
                Slover.getVariable(
                    ViList.get( $i$ ).VLID+
                    toString( $k+1$ )
                )
22.         Bag $\leftarrow$ ViList.get( $i$ ).BAG
23.         Slover.add(
             dispatch_point2>
             dispatch_point1+ViList.get( $i$ ). $L_{\max}$ /
             Speed+Bag
24.         )
25.     end for
26. end for
```

3) 消息发送时刻表生成

对求解的参数列表进行整理,生成消息发送时刻表。消息发送时刻表格式如表 7 所示。

表 7 消息发送时刻表格式

组成部分	描述
VLID	当前链路号
seqID	当前链路发送消息序列号
START/ μ s	当前链路发送消息的开始时间
END/ μ s	当前链路发送消息的结束时间

3 实验验证

本节对算法的实现进行验证,提取终端配置文件链路的基本参数:①链路标识 VLID,对链路进行唯一性标识;②2 个相邻帧之间的最小带宽分配 BAG,其范围为 1~128 ms;③链路的最大帧长 $L_{\max}$ 。链路配置信息(部分)如表 8 所示。

表 8 端系统链路配置信息表(部分)

VLID	BAG/ms	$L_{\max}$ /byte
33 818	7.977	664
33 639	12.977	728
33 636	7.977	856
33 643	7.977	920
33 641	9.657	1 560
33 634	7.977	2 072

通过 SMT 求解器计算,得出测试板卡的消息发送时刻表,如表 9 所示。消息时刻表为测试板卡消息发送提供解决方案,实现消息发送不冲突,并且满足链路消息发送时间间隔的规划表(部分)。

表 9 消息发送时刻表(部分)

VLID	seqID	START/ μ s	END/ μ s
33 818	1	0	63.32
33 639	1	191.28	260.71
33 636	1	382.56	464.19
33 643	1	592.15	679.89
33 641	1	775.62	924.94
33 634	1	1081.71	1 279.31

由表 9 可得,在每个端口所发送的 VL 之间时间间隔紧密排列,且同一 VL 2 次发送之间的时间间隔较小,具有发送频率高、数据规模大的特点。

最后在 TestCenter 测试平台进行仿真实验,实验结果证明当前网络中所有的交换机能按照本文生成的激励数据时刻表进行数据流的收发,实现消息发送不冲突。同时证明本文提到的算法能很好地满足压力测试以及负载测试对数据量的要求。

4 结 论

本文首先研究 AFDX 终端功能,对终端的测试

内容进行确认,建立测试总体框架,设计了 AFDX 端系统配置文件的格式,建立 AFDX 网络调度的模型,并基于 SMT 求解器激励数据生成架构,对数据发送的约束条件进行形式化定义,通过 SMT 求解器,设计并生成激励数据发送时刻表,保证消息按照时间间隔无冲突发送。最后通过在 TestCenter 网络测试平台进行仿真实验,实验结果证明本文提到的算法能使 AFDX 网络配置中的 VL 在限定的约束范围内紧密排列,也能够很好地模拟高负载下 AFDX 网络配置所属的网络运行状态,很好地满足压力测试以及负载测试对数据量的要求。

参考文献:

[1] 王庆伟. 新一代军用飞机航空电子系统发展趋势与发展现状[J]. 教练机, 2019(3): 5-11
WANG Qingwei. The development trend and current situation for avionics system of new generation of military aircraft[J]. Trainer, 2019(3): 5-11 (in Chinese)

[2] 王元勋, 张双, 孔德岐, 等. AFDX 网络端系统测试与分析[C]//中国航空学会 2020(第九届)民用飞机航电国际论坛, 2020
WANG Yuanxun, ZHANG Shuang, KONG Deqi, et al. Test and analysis of AFDX network end system[C]//9th Annual Civil Avionics International Forum 2020, 2020 (in Chinese)

[3] 曹阳. 浅谈航空电子数据总线技术[J]. 数字通信世界, 2018(3): 35
CAO Yang. A brief introduction to avionics data bus technology[J]. Digital Communication World, 2018(3): 35(in Chinese)

[4] WIJUTOMO C W, TRILAKSONO B R, KISTIANTORO A I. Development of test environment platform for ima using cots components[C]//2019 International Seminar on Application for Technology of Information and Communication, 2019

[5] MOUYSET S, ERMONT J, SCHARBARG J L. Limiting over sampling to improve transmission schedulability in a mixed NoC/AFDX architecture[C]//Emerging Technologies and Factory Automation, 2019

[6] WANG K, LI W, WANG L, et al. Research on the test method of AFDX end system[C]//2022 7th International Conference on Intelligent Computing and Signal Processing, 2022: 806-809

[7] KAO W W. Research on AFDX network end system testing technology[J]. Aviation Maintenance and Engineering, 2015, 8: 92-94

[8] WANG S K, WEN L. Technical research and platform implementation of the protocol conformance test of AFDX network switch [C]//2018 International Conference on Networking and Network Applications, 2018

[9] 苗佳旺, 乔飞, 盖峰. AFDX 网络系统测试设计与实现[J]. 计算机测量与控制, 2018, 26(5): 33-36
MIAO Jiawang, QIAO Fei, GAI Feng. Design and implement of AFDX network system testing[J]. Computer Measurement & Control, 2018, 26(5): 33-36 (in Chinese)

[10] HE J, ZHANG Y, HU G. Design of ARINC664 bus network test system based on WinPCap[C]//2020 5th International Conference on Electromechanical Control Technology and Transportation, 2020

[11] KERKECH M, BUI V T, AFRICANO M, et al. Protocol recognition in virtual avionics network based on efficient and lightweight convolutional neural network[C]//2022 IEEE/ACM 26th International Symposium on Distributed Simulation and Real Time Applications, 2022: 9-16

[12] ZHANG C. Stress testing SMT solvers via type-aware mutation[C]// 42nd International Conference on Software Engineering, 2020

Research on AFDX network configuration testing technology

CHEN Fang¹, LYU Guangzhe², LIN Wei³, HOU Xixi¹, MA Chunyan³

- 1.China Institute of Aeronautical Radio Electronics, Shanghai 200233, China;

2.Xi'an Aeronautical Computing Technique Research Institute, Xi'an 710068, China;

3.School of Software, Northwestern Polytechnical University, Xi'an 710072, China

Abstract: The terminal system, as a key part of AFDX network data transmission, determines the communication performance of the entire AFDX network. In order to test the functionality of the verification end system and evaluate the transmission performance of the AFDX network, this article proposes an AFDX network configuration testing method. Firstly, the configuration testing specifications for the end system is designed, an overall framework for end system testing is established; the testing response mechanism of the end system is studied; the testing process is standardized; and generated configuration files are as input information for algorithm design in the following stage. Secondly, incentive data generation algorithms is designed, incentive data generation frameworks is established, the constraints for data transmission are formalized, and incentive data transmission schedules are generated using SMT solvers. Next, an AFDX network configuration testing tool is designed and implemented; the performance from the perspectives of operational stability and response time is evaluated to prove the effectiveness of the tool. Finally, the types and performance of the seven clock function modules in the overlay network configuration are tested, and the experimental results prove the effectiveness of the network configuration testing method.

Keywords: AFDX network; end system; virtual link; automated testing

引用格式:陈芳, 吕广喆, 林伟, 等. AFDX 网络配置测试技术研究[J]. 西北工业大学学报, 2024, 42(3): 514-520
CHEN Fang, LYU Guangzhe, LIN Wei, et al. Research on AFDX network configuration testing technology[J]. Journal of Northwestern Polytechnical University, 2024, 42(3): 514-520 (in Chinese)