

AFDX

焦文喆¹,翟正军¹,王国庆^{1,2}

1. 西北工业大学计算机学院, 陕西 西安 710129

2. 中国航空无线电电子研究所, 上海 200233

: 为了解决航空电子全双工交换式以太网(AFDX, Avionics Full Duplex Switched Ethernet)时间关键消息传输的确定性问题,本文提出一种时间触发AFDX(Time-Triggered AFDX, TTAFDX)网络的体系结构,同时设计了基于周期优先的端系统时间触发虚拟链路调度算法。通过网络演算方法对比了TTAFDX与采用FIFO调度算法的AFDX网络的实时性能。计算结果表明:TTAFDX中的时间触发虚拟链路的延迟主要由固定延迟部分组成,而且速率限制虚拟链路的实时性也较AFDX有所提高。证明了TTAFDX体系结构和调度算法在兼容AFDX的同时能够改善网络的时间确定性,并适用于具有硬实时传输要求的航空电子系统。

: 航空电子全双工交换式以太网;时间触发;调度算法;实时性分析;网络演算

: TP393

: A

: 1000-2324(2016)01-0111-07

Scheduling Strategy Design and Real-time Analysis of Time-triggered AFDX

JIAO Wen-zhe¹, ZHAI Zheng-jun¹, WANG Guo-qing^{1,2}

1. School of Computer Science/Northwestern Polytechnical University, Xi'an 710129, China

2. China National Aeronautical Radio Electronics Research Institute, Shanghai 200233, China

Abstract: To solve the temporal determinism problem of time-critical message transmission in Avionics Full Duplex Switched Ethernet (AFDX), this paper proposed the network architecture of Time-triggered AFDX (TTAFDX) and designed the scheduling algorithm of time-triggered virtual link based on cycle-prior end system to contrast the real-time performance between TTAFDX and AFDX by the method of network calculus. The results showed that the delay of time-triggered virtual link in TTAFDX mainly consisted of fixed part and the real-time of the rate constraint virtual link was higher than AFDX. At the same time, TTAFDX architecture was compatible to AFDX and could enhance temporal determinism to be suitable for the avionics with hard real-time transmission requirement.

Keywords: AFDX; time-triggered; scheduling strategy; real-time analysis; network calculus

AFDX^[1]作为应用于空客 A380 和波音 B787 上的新一代航空电子网络,具有全双工、高速率、双冗余、易扩展等优点^[2],但当多任务传输时可能产生的竞争链路共享冲突,仍会影响数据传输的实时性和通信系统的时间确定性。对于时间关键消息,其在 AFDX 网络中的传输时延及时延抖动的确定性界限仍不易得到保证。

为了提高航电系统中时间关键消息的时间确定性,并尽可能降低既有设备改造的技术与商业风险,本文在 AFDX 网络基础上引入时间触发机制,设计完成了时间触发 AFDX(Time-Triggered AFDX, TTAFDX)的体系结构,并提出了基于周期优先原则的端系统调度策略。通过采用网络演算的方法分别对 TTAFDX 及 AFDX 网络的时延进行比较分析,验证了 TTAFDX 网络传输时间确定性的特点。

1 AFDX

1.1 AFDX

AFDX 网络通过采用虚拟链路技术、冗余传输技术和流量管制技术对通信任务的速率进行了约束,称此类传输数据为速率限制(Rate Constraint, RC)流量^[3]。TTAFDX 在 AFDX 基础上引入了时间触发机制,通过合理地调度将通信任务定时触发发送,称此类数据传输为时间触发(Time-Triggered, TT)流量。时间触发机制建立在全局时钟精确同步的前提下,因而 TTAFDX 需要加入时间同步技术,并对虚拟链路技术,冗余传输技术做出适应性改造以兼容 AFDX 网络。

1.1.1 时钟同步技术 精确的全局时钟同步能够确保任一时刻仅有一个节点对网络进行访问,能够避免排队所造成的时间延迟和抖动,使得数据传输延迟可预测^[4]。TTAFDX 采用 SAE AS6802 协议中所提出的同步技术,并在每个调度周期的固定时刻使用时钟同步协议控制帧(Protocol Control Frame,

: 2014-10-20

: 2014-11-06

: 焦文喆(1985-),男,博士,主研方向:航电网络,网络存储技术,数据安全. E-mail:wenzhejiao@163.com

数字优先出版: 2015-11-06 http://www.cnki.net

PCF)进行时钟同步。TTAFDX 网络中的交换机采用 IEEE 1588 V2 提出的透明时钟(Transparent Clock, TC)概念,使用 PCF 帧中的 8 字节透明时钟域,以 2^{16} ns 为计时单位累计同步协议控制帧的传输延迟。

1.1.2 虚拟链路技术 TTAFFDX 网络按照所支持的流量类型定义了两类 VL: 用以承载时间关键消息的时间触发 VL (Time-Triggered VL, TTVL) 和继承 AFDX 网络中 VL 的速率限制 VL (Rate-Constrained VL, RCVL) [6]。TTVL 的发送活动严格按照时刻调度表的规划时刻来触发,在全局时钟精确同步的条件下,可以避免数据在链路上发生共享冲突,确保其时间确定性和实时性。RCVL 的优先级低于 TTVL,在保障 TTVL 按规划传输的基础上才能进行传输。

1.1.3 冗余管理技术 TTAFFDX 设计了两种冗余管理机制: 异步冗余管理 (asynchronous Redundancy Management, aRM) 和同步冗余管理 (synchronous Redundancy Management, sRM)。

TTAFDX 网络中的 RC 流量数据帧的到达时刻具有一定的不确定性,目的端系统无法通过接收时间窗进行冗余管理,因此沿用符合标准 ARINC 664 Part 7 的异步冗余管理机制 [5]。

同步冗余管理 (synchronous Redundancy Management, sRM) 以全局时钟同步为基础,对 TT 消息进行冗余管理。目的端系统通过查询调度表能够知道 TTVL 中数据帧的最坏接收时间点,当最坏接收时间点到达时,控制器将通过完整性检查的拷贝传递至应用层并清除所有临时存储的冗余拷贝。

1.2 AFDX

TTAFDX 网络的端系统协议与 AFDX 相同,按照 OSI 分层模型共分为 7 层,其中网络层、传输层和应用层与普通的以太网相同。为了完全兼容 AFDX,TTAFDX 仅在数据链路层中增加了 VL 的时间触发通信机制,所增加的 VL 时间调度层不影响数据链路层以上的协议。TTAFDX 同样采用端口、队列端口以及 SAP 与应用程序进行通信。

普通 AFDX 数据帧中的目的 MAC 地址具有 32 位的常量域和 16 位的 VL 标志符,可将 32 位的常量域改造为 8 位固定域与 24 位的虚拟链路类型设置域。当端系统检测到 TT 标识符域的高 4 位为“0110”时,则数据帧所在的虚拟链路为时间触发虚拟链路,采用时间触发调度策略进行调度;若不是,则为速率限制虚拟链路,采用常规调度策略。

2

TTAFDX 中时间触发调度本质上是一种基于表的静态调度,但不同与 TTCAN 等总线结构的时间触发调度 [7]——它们的各个端系统具有相同的调度表,TTAFDX 网络中每个 TT 帧的发送端都都有一个 TT 帧发送时刻调度表。VL 时刻调度表包含一个矩阵周期 (Matrix Cycle, MC),它由若干个基本周期 (Basic Cycle, BC) 组成 [3]。其中 MC 表示为时间触发虚拟链路数据帧调度的任务周期;BC 为时间触发虚拟链路数据帧传输的最小周期。对于系统中的 N 个周期消息,取最小周期作为 BC。为了保证每一消息的平均周期,MC 的持续时间必须是所有消息周期的最小公倍数或最小公倍数的整数倍。根据 AFDX 协议中规定的 BAG 取值范围为 2^k ms, $k=0, \dots, 7$,可以设定 VL 时刻调度表的 MC 为所有 BAG 取值的最小公倍数,即 128 ms, BC 为所有 BAG 取值的最大公约数 1 ms。

VL 时刻调度表中的每个 BC 都开始于一个时钟同步数据帧 (SYNC),并且以下一个 SYNC 的开始作为结束。SYNC 采用 SAE AS6802 协议中的 PCF 格式,具有 28 字节长度,其中前 4 个字节用于记录时间所处的 BC 位置,最后 8 个字节用于累计 SYNC 帧的传输延迟。以 BC 为时间单位进行时钟同步可以保障网络中各个节点的时钟是完全同步的,使得完善的离线设计 VL 时刻调度表能够保证 TT 帧在端系统不会发生争用物理链路的情况。每个 BC 又分为两段,前一段专用于发送 TT 流量数据 (包含 SYNC 帧),后一段用于发送 RC 流量数据 [6]。RC 端的最后可预留出一个当前所有通信任务中最大帧长的传输时间作为保护间隔,以避免对下一 BC 内 TT 帧的发送造成延迟。在 TT 段中,时间资源又被分为时窗 (Time window)。由于 TTAFFDX 中 TT 流量的帧长不固定,BC 中各个时窗的大小也不确定,但 MC 中同一列时窗的大小相同,其大小等于在此时窗中传输消息的最大帧长。

2.1

每个端系统对 TTVL 发送时刻的规划都将按照如下的调度算法执行。

1) 对端系统发送的 N 条 TTVL 按照 BAG 从小到大的顺序排序, 若 BAG 相同则按照 L_{\max} 长度由大到小的顺序排序, 排序后的 TTVL 对应标记为 VL_k (k 从 1 到 N), 相应的最大线路帧长记为 L_k , 转到第 2 步;

2) 按照标记顺序对 TTVL 发送时刻进行规划, 初始状态令 $k=1$, $L_{\text{sum}}=L_{\text{SYNC}}$, 其中 L_{sum} 表示基本周期已配置的帧长和, 转至第 3 步;

3) 若 $k>N$, 跳至第 4 步; 否则配置 VL_k 的发送时刻, 虚拟链路 VL_k 的 BAG 值为 G_k , 则在一个 MC 内 VL_k 被循环调度的次数为 $128/G_k$; 从调度表 0ms 开始, 在 G_k 时间范围中, 按照“左紧凑”原则, 即尽量靠近左侧的时窗列中分配发送时窗。当所分配的时窗列为空, 则更新 BC 已配置帧长和 $L_{\text{sum}} = L_{\text{sum}} + L_k$, 若 L_{sum} 大于 BC 上最多可以配置的帧长和, 则跳到第 5 步; 否则, 将 VL_k 的 $128/G_k$ 个调度时刻 (单位 ms) 配置为 $t_{k,m} = (m-1) \times G_k + 8 \times (L_{\text{sum}} - L_k) \times 1000 / (C \times 10^6)$, $m \in \{1, \dots, 128/G_k\}$, 其中 $t_{k,m}$ 表示 VL_k 在 MC 中的第 m 次调度时刻, 并赋值当前所在时窗列长度 $L_{TW} = L_k$ 。当所分配的时窗列不为空时, 记所找到的第一个合适位置在第 a 个 BC 中, 则将 L_k 与当前所在时窗列的长度 L_{TW} 进行比较, 若 $L_{TW} \geq L_k$, 则 L_{TW} 不变, 否则将 L_{TW} 赋值为 L_k 并更新 $L_{\text{sum}} = L_{\text{sum}} + L_k - L_{TW}$; 若 L_{sum} 大于 BC 上最多可以配置的帧长和, 则跳到第 5 步; 否则, 将 VL_k 的 $128/G_k$ 个调度时刻配置为 $t_{k,m} = (a-1) + (m-1) \times G_k + 8 \times (L_{\text{sum}} - L_k) \times 1000 / (C \times 10^6)$, $m \in \{1, \dots, 128/G_k\}$;

4) 该端系统已完成所有 TTVL 的发送时刻规划;

5) 带宽有限, 无法对该端系统中所有 TTVL 的发送时刻完成规划。

2.2

交换机每个输出端口都维护一张通过此端口转发的 TTVL 转发时刻调度表。TTVL 数据帧到达交换机输出端口的时刻不由交换机端口决定, 所以转发表的规划将不采用基本周期组成矩阵周期的方式, 而是直接以 128 ms 为周期进行转发时刻表的规划。当所有 VL 的静态路由和所有 TTVL 的发送时刻配置完成后, 交换机内部的所有端口的转发时刻调度表配置算法如下:

1) 按照 BAG 从小到大的顺序对需要转发的 N 条 TTVL 排序, 若 BAG 相同则按照 L_{\max} 长度由大到小的顺序排序, 排序后的 TTVL 对应标记为 VL_k (k 从 1 到 N), 相应的最大线路帧长记为 L_k 转到第 2 步;

2) 按照标记顺序规划 TTVL 在交换机中的转发时刻, 初始状态令 $k=1$, 其转到第 3 步;

3) 若 $k > N$, 跳至第 4 步; 否则按照 VL_k 所经过交换机转发端口的顺序规划其在这些端口的转发时刻。TT 虚拟链路 VL_k 的 BAG 值为 G_k , 则在一个调度周期中都将转发 $128/G_k$ 帧该 TT 数据, 以 $V_{k,i}$ ($1 \leq i \leq 128/G_k$) 依次标识各个帧。规划 VL_k 在其转发路径中的任一端口的转发时刻时, 按照帧 $V_{k,1}$ 到帧 $V_{k,w}$ ($w=128/G_k$) 的顺序规划各帧数据的转发时刻。根据帧 $V_{k,i}$ ($1 \leq i \leq 128/G_k$) 在上一节点的发送时刻 t_1 通过公式 $t_2 = t_1 + \frac{8 \times L_k \times 1000}{C \times 10^6} + D_{\text{switch}} + 2 \times \Delta T$ 能够计算接收窗口的关闭时刻 t_2 。其中

$D_{\text{switchtech}}$ 为交换机时延; ΔT 为系统所允许的最大时钟漂移。在不改变已经规划好的 TTVL 帧的原则下, 按照 $0 \leq \Delta t \leq 128$ 从小到大的方向, 在一个调度周期中寻找最近的能够完成转发任务的空闲时间段, 若能找到, 则将该空闲时间段的时刻起点 $t_3 = (t_2 + \Delta t) \% 128$ 作为转发时刻, 更新交换机端口转发时刻表; 若找不到, 则跳至第 5 步。若 $V_{k,i}$ ($1 \leq i \leq 128/G_k$) 在所经过的交换机转发端口都能完成转发时刻的规划, 则 $k++$, 循环第 3 步, 否则跳至第 5 步;

4) 该交换机端口已完成所有 TTVL 的转发时刻规划;

5) 带宽有限, 无法对该交换机端口中所有 TTVL 的转发时刻完成规划。

3 TTAFDX

TTAFDX 网络端到端时延主要是指数据从网络中的源端节点传输到目的端节点所花费的时间, 包括链路传输时延 $D_{\text{propagation}}$ 、源端系统时延 D_{tx} 、交换机时延 D_{switch} 和目的端系统时延 D_{rx} , 虚拟链路 VL_k 的网络时延定义可表示如下:

$$D = D_{propagation} + D_{tx, k} + \sum_{i=0}^j D_{switch, k} + D_{rx, k} \tag{1}$$

式中, $D_{propagation}$ 是数据帧在物理链路上传输所引起的时延, 由于物理介质上传播速度是固定的, 这一时延具有确定值; 数据在发送端的时延 $D_{tx, k}$ 可用公式 (2) 定义:

$$D_{tx, k} = G_k + \tau_{tech} + \tau_{frame, k} + EJ_{max, k} \tag{2}$$

式中, G_k 为虚拟链路 VL_k 的带宽分配间隔; τ_{tech} 为源端系统的技术时延, ARINC 664 协议中规定端系统的技术延时不超过 $150 \mu s$; $\tau_{frame, k}$ 是虚拟链路 VL_k 将数据帧发送至物理链路层所造成的延迟; $EJ_{max, k}$ 为复用调度策略所导致的数据队列等待时延。 $D_{rx, k}$ 主要为技术延时, ARINC 664 协议中规定目的端的技术延时不得超过 $150 \mu s$ 。

TTAFDX 数据帧在交换机中将经过数据帧过滤、数据帧转发和数据帧发送三个步骤, 其时延计算方法为:

$$D_{switch, i} = \tau_{filter} + \tau_{fw} + \tau_{rv} + SJ_{max, i} \tag{3}$$

式中, τ_{filter} 为数据帧过滤时延, AFDX 协议规定每个交换机端口必须能及时过滤以线速到达的任意大小的有效数据帧, 通常设为 $8 \mu s$; τ_{fw} 为数据帧转发时延, AFDX 规定交换机在 $1 ms$ 内要求转发的数据帧数量一般为交换机的端口数*125, 则 τ_{fw} 可设为 $8 \mu s$; τ_{rv} 是交换机接收端口在接收数据帧时产生的时延, 由各虚拟链路所传输的最大帧长度决定; $SJ_{max, i}$ 为交换机系统多路复用时延。

3.1 TTAFDX

TTAFDX 采用与传统 AFDX 相同的星型拓扑网络结构, 具有转发速度快, 时延小, 实时性好, 能够降低网络负荷等特点。本文采用的网络模型如图 1 所示, 包括 3 个交换机 SW1, SW2, SW3, 8 个端系统 ES1, ..., ES8 和 12 条虚拟链路 VL1, ..., VL12, 各个虚拟链路均为单播虚拟链路, 且相邻节点间的距离为 $100 m$, 所使用物理链路的传输速率为 $2 \times 10^8 m/s$ 。

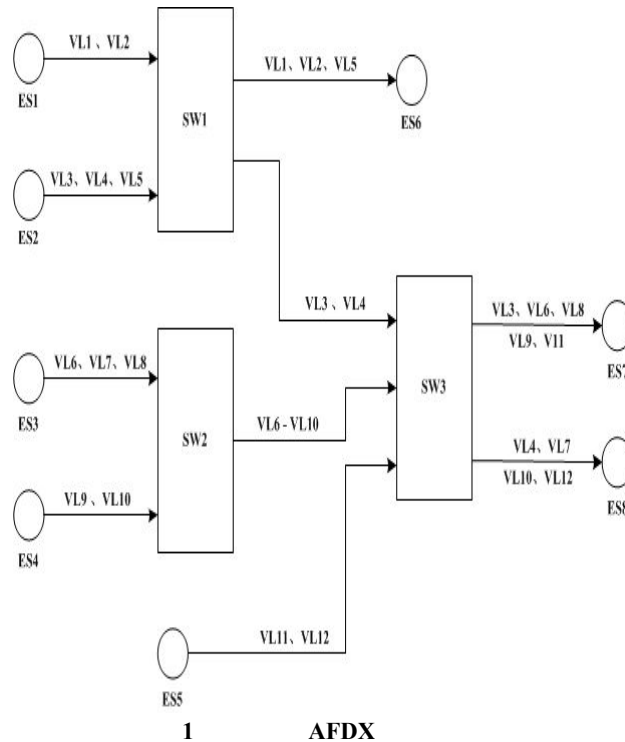


Fig.1 Time-triggered AFDX network model

表 1 列出了图 1 中 12 条虚拟链路的相关参数:

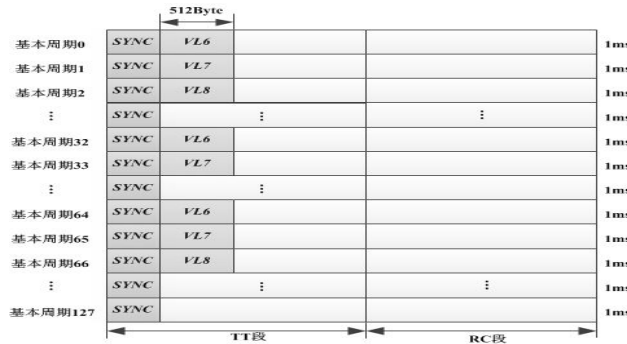
1 AFDX

Table 1 Virtual link configuration of time-triggered AFDX network model

虚拟链路ID Virtual link ID	$L_{max}(\text{Byte})$	BAG(ms)	虚拟链路类型 Virtual link type
VL1	512	16	时间触发
VL2	256	8	速率限制
VL3	128	32	时间触发
VL4	256	64	时间触发
VL5	1024	32	速率限制
VL6	512	32	时间触发
VL7	256	32	时间触发
VL8	512	64	时间触发
VL9	128	128	速率限制
VL10	128	4	速率限制
VL11	1024	16	时间触发
VL12	64	64	速率限制

3.2 TTAFDX

以时间触发虚拟链路 VL6 为例进行分析, VL6、VL7 和 VL8 由 ES3 发送至 SW2, 按照周期优先调度表设计方法可得 ES2 的时刻调度表, 如图 2 所示。

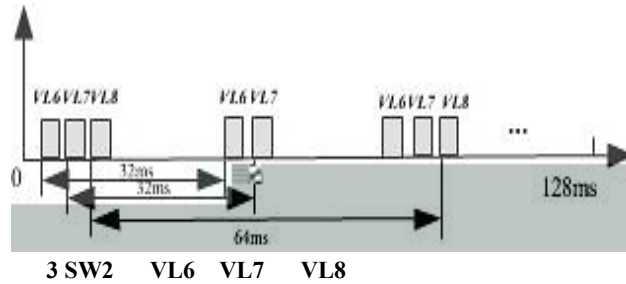


2 ES3

Fig.2 End system ES3 schedule time diagram

在端系统 ES3, VL6 首次被调度的时刻 $t_{6,1} = \frac{L_{SYNC} \times 8 \times 1000}{100 \times 10^6} = \frac{28 \times 8 \times 1000}{100 \times 10^6} = 0.00224ms$, 其它数据帧被调度的时刻根据设计方法步骤 3 中所述的不同情况选用相应的计算公式。

VL6、VL7 和 VL8 三条 TTVL 到达交换机 SW2 之后转发至交换机 SW3 的时刻调度表如图 3:



3 SW2 VL6 VL7 VL8

Fig.3 Schedule time diagram at SW2 for VL6 VL7 and VL8

依照公式 (3) 可得 VL6 在交换机 SW2 中第一次被调度的时刻为:

$$t_{6,1} = \frac{L_{SYNC} \times 8 \times 1000}{100 \times 10^6} + 2 \times \frac{L_{VL6} \times 8 \times 1000}{100 \times 10^6} + \tau_{filter} + \tau_{fw} + D_{propagation} = 0.10066ms$$

TTVL6 到达交换机 SW3 之后将转发至端系统 ES7, 其第一次被调度的转发时刻为:

$$t_{6,1} = \frac{L_{SYNC} \times 8 \times 1000}{100 \times 10^6} + 4 \times \frac{L_{VL6} \times 8 \times 1000}{100 \times 10^6} + 2 \times (\tau_{filter} + \tau_{fw} + D_{propagation}) = 0.19908ms$$

根据对各 TTVL 在传输路径中的调度时刻, 能够计算出各个 TTVL 的端到端延迟, 如表 2 所示:

2 TTAFFDX

Table 2 Time-triggered virtual link end-to-end delay statistics in TTAFFDX

TTVL	时延上界(μs) Delay upper bound	TTVL	时延上界(μs) Delay upper bound
VL1	139.88	VL7	135.9
VL3	84.7	VL8	238.3
VL4	156.38	VL11	262.76
VL6	238.3		

由于 TTVL 数据流在配置的时域内独享整个带宽, 且在传输路径中完全按照调度时刻表进行发送, 无需进行排队等待, 所以 TTVL 的端到端时延仅包含固定时延部分, 也不存在时延抖动的现象。表 2 中各 VL 的时延能够有效地证明以上结果。

3.3 AFDX 中 FIFO 调度算法实时性分析

对于通过同一交换机端口输出的 n 条虚拟链路, 它们将共享输出端口物理带宽 C 。虚拟链路 $VLi(1 \leq i \leq N)$ 的带宽分配间隔为 BAG_i , 最大数据帧长度为 L_{max}^i , 且满足 $\sum_{i=1}^n (L_{max}^i / BAG_i) \leq C$ 时, 各虚拟链路的到达曲线可表示为: $\alpha_i(t) = L_{max}^i + (L_{max}^i / BAG_i)t$ 。另一方面, 虚拟电路在到达交换机后依照抵达顺序存储在 FIFO 缓冲区中, 其提供给所有虚拟链路的总服务曲线为: $\beta(t) = C[t - 0]^+$ 。

由 FIFO 最小服务曲线推论^[8], 可得到交换机提供给第 i 个虚拟链路的服务曲线为: $\beta_i(t) = [(\beta(t) - (\alpha(t - \theta) - \alpha_i(t - \theta)))]^+$ 。等式右边是聚合虚拟链路中除第 i 条虚拟链路以外的其它数据流在 0 时刻的积压数据, 由 $\beta(\theta) = \alpha(0) - L_{max}^i$ 可得 $\theta = \left(\sum_{j=1}^n L_{max}^j - L_{max}^i \right) / C$ 。综上所述, 可推导出公式(4):

$$\beta_i(t) = \left[C - \left(\sum_{i=1}^n \frac{L_{max}^i}{BAG_i} - \frac{L_{max}^i}{BAG_i} \right) \left(t - \frac{\sum_{i=1}^n L_{max}^i - L_{max}^i}{C} \right) - \left(\sum_{i=1}^n L_{max}^i - L_{max}^i \right) \right]^+ = \left[C - \left(\sum_{i=1}^n \frac{L_{max}^i}{BAG_i} - \frac{L_{max}^i}{BAG_i} \right) \right]^+ \left(t - \frac{\sum_{i=1}^n L_{max}^i - L_{max}^i}{C} \right) \quad (4)$$

由公式(5)可得交换机提供给第 i 条虚拟链路的服务速率和服务时延分别为 $R_i(t) = C - \left(\sum_{i=1}^n \frac{L_{max}^i}{BAG_i} - \frac{L_{max}^i}{BAG_i} \right)$ 和 $d_i = \left(\sum_{i=1}^n L_{max}^i - L_{max}^i \right) / C$ 。在级联网络系统中, 第 i 条虚拟链路数据流在此交换机的离开曲线为: $\alpha_i^* = \alpha_i(t + d_i)$ 。此离开曲线即为下一级交换机的到达曲线。

以虚拟链路 VL6 为例, 其串联服务曲线为:

$$\beta_6^\Sigma(t) = \delta_\tau \otimes \beta_{SW2,6}(t) \otimes \delta_\eta \otimes \delta_{rv} \otimes \delta_r \otimes \beta_{SW3-ES7,6}(t) \otimes \delta_\eta \otimes \delta_{rv} \otimes \delta_r$$

根据表 2 中所设定的各虚拟链路参数, 经过计算, 得到 12 条虚链路的延迟上界(表 3)。

3 AFDX FIFO

Table 3 Virtual link end-to-end delay statistics in AFDX using FIFO schedule strategy

TTVL	时延上界(μs) Delay upper bound	TTVL	时延上界(μs) Delay upper bound
VL1	242.49	VL7	274.62
VL2	201.43	VL8	464.26
VL3	289.81	VL9	371.73
VL4	182.47	VL10	243.54
VL5	324.63	VL11	365.52
VL6	464.1	VL12	83.94

3.4 TTAFFDX

TTAFFDX 网络中 TT 数据流具有高优先级, 而 RC 数据流具有低优先级。TT 数据流在配置时域内独占整个带宽, 不受 RC 数据流的影响。当 TT 数据流按照配置时间完成传输后, 调度器会对所在端口的 RC 数据流执行发送。当有多个 RC 数据流等待发送时, 调度器按照 FIFO 顺序, 依次对 RC 数据流进行操作。对于共享 TTAFFDX 交换机同一输出端口的多个 VL, 定义 $G^{TT} = \cup TTVLi$ 和

$G^{RC} = \cup RCVLi$ 分别代表 TTVL 和 RCVL 的聚合流。由于 TT 数据流不会受到 RC 数据流的阻塞，因此输出端口提供给 TTVL 的服务曲线即交换机输出端口能够提供的总服务曲线，为： $\beta_{G^{TT}}(t) = \beta(t) = C[t - 0]^+$ 。而对于 RCVL 聚合流 G^{RC} 的服务曲线为： $\beta_{G^{RC}}(t) = [\beta(t) - A^{TT}(t)]^+$ ，假设 VL 的到达曲线为 $\alpha(t) = rt + b$ ，则 G^{TT} 的到达曲线 $A^{TT}(t)$ 为： $A^{TT}(t) = \sum_{\{i \in G^{TT}\}} \alpha_i(t) = \sum_{\{i \in G^{TT}\}} r_i \times t + \sum_{\{i \in G^{TT}\}} b_i$ 。

综上所述，可推导出公式 (5)：

$$\beta_{G^{RC}}(t) = \left[Ct - \sum_{\{i \in G^{TT}\}} r_i \times t - \sum_{\{i \in G^{TT}\}} b_i \right]^+ = \left[C - \sum_{\{i \in G^{TT}\}} r_i \right] \left[t - \frac{\sum_{\{i \in G^{TT}\}} b_i}{C - \sum_{\{i \in G^{TT}\}} r_i} \right]^+ \quad (5)$$

记 RCVL 聚合流 G^{RC} 的服务速率和服务时延参数分别为 $R_{G^{RC}} = C - \sum_{\{i \in G^{TT}\}} r_i$ 和 $T_{G^{RC}} = \sum_{\{i \in G^{TT}\}} b_i / R_{G^{RC}}$ 。

对于 RCVL 聚合流 G^{RC} 的组成虚拟链路 VL_i ，其在 G^{RC} 中仍以 FIFO 调度策略接受交换机的服务，按照 3.3 小节中 FIFO 调度策略部分的相关推导，可得：

$$\begin{aligned} \beta_{VL_i}(t) &= \left[R_{G^{RC}}(t - T_{G^{RC}}) - \left[\sum_{\{j \in G^{RC}\}} r_j - r_i \right] \times \left[t - T_{G^{RC}} - \frac{\sum_{\{p(j)=L\}} b_j - b_i}{R_{G^{RC}}} \right] - \left[\sum_{\{j \in G^{RC}\}} b_j - b_i \right] \right]^+ \\ &= \left[R_{G^{RC}} - \sum_{\{j \in G^{RC}\}} r_j + r_i \right] \times \left[t - \frac{R_{G^{RC}} \times T_{G^{RC}} + \sum_{\{j \in G^{RC}\}} b_j - b_i}{R_{G^{RC}}} \right]^+ \end{aligned} \quad (6)$$

根据表 2 中所设定的各虚拟链路参数，经过计算，得到 5 条速率限制虚链路的延迟上界(表 4)。

4 TTAFDX

Table 4 Rate-constraint virtual link end-to-end delay statistics in TTAFDX

RCVL 帧 RCVL frame	时延上界(μs) Delay upper bound	RCVL 帧 RCVL frame	时延上界(μs) Delay upper bound
VL2	201.74	VL10	243.8
VL5	324.78	VL12	119.48
VL9	373.3		

4

TTAFDX 网络在时间同步的基础上，将时间触发机制引入 AFDX，通过对虚拟链路、网络协议等进行适应性从而达到在兼容速率限制流量的同时，能够确保时间关键消息的时间确定性和实时性。针对时间触发虚拟链路的特性，提出了周期优先的端系统调度时刻设计算法以及相应的交换机调度算法。采用网络演算方法对 AFDX 所支持的“先到先服务”调度算法进行了时延分析。通过与 TTAFDX 中的时间触发虚拟链路和速率限制虚拟链路时延进行比较，能够发现时间触发虚拟链路能够支持硬实时特性的数据传输，同时速率限制虚拟链路的时延也能够有所降低。

[1] Pickles B. Avionics Full Duplex Switched Ethernet (AFDX)[J]. SBS Technologies, 2006, 15(2): 62-65.
 [2] 徐科华.AFDX 总线网络数据传输分析[J].民用飞机设计与研究,2009(3):35-40
 [3] 刘晚春,李 峭,何 锋,等.时间触发以太网同步及调度机制的研究[J].航空计算技术,2011,41(4):122-127
 [4] Steinbach T, Korf F, Schmidt TC. Real-time Ethernet for automotive applications: A solution for future in-car networks[C]//Consumer Electronics-Berlin (ICCE-Berlin). Berlin: 2011 IEEE International Conference, 2011:216-220
 [5] 熊华钢,李 峭,黄永葵.航空电子全双工交换式以太网标准研究(续)[J].航空标准化与质量,2008(2):20-23
 [6] 刘 成,王 彤,李 铮,等.时间触发 AFDX 网络的设计和实时性分析[J].北京航空航天大学学报,2013,39(6):728-733
 [7] Li X. Research on the Real-Time Property of TTCAN Protocol in Vehicle Communication Network[J]. Applied Mechanics and Materials, 2014,487:674-677
 [8] Boudec JYL, Thiran P. Network calculus: a theory of deterministic queuing systems for the internet[M]. Germany: Springer-Verlag, 2001:272