

## 面向空间应用的时间触发以太网\*

邱爱华<sup>1,2</sup>, 张涛<sup>1</sup>, 顾逸东<sup>1</sup>

(1. 中国科学院空间科学与应用总体部, 北京 100094; 2. 中国科学院大学, 北京 100049)

**摘要:**将时间触发协议增加到采用星型拓扑结构的以太网业务上,且保持完全兼容 IEEE802.3 标准,可解决空间有效载荷使用标准以太网通信遇到的实时性、确定性和可靠性问题。时间触发以太网基于全局基准时间,实时数据传输遵循严格的时序,支持时间和事件触发的两种通信过程。时间触发通道适合同步或周期性实时消息的传输,而事件触发通道适用于偶发或非周期性消息的传输。设计了空间应用以太网的拓扑结构和协议栈,介绍了系统通信的过程,进行了时间触发协议的研究,并对同步性能和网络性能进行了仿真验证。该以太网在保证可靠性、安全性的同时,增强了网络的灵活性,提高了网络利用率。

**关键词:**时间触发;以太网;空间应用;实时性;确定性

**中图分类号:**V19 **文献标志码:**A **文章编号:**1001-2486(2014)05-117-07

## Time-triggered Ethernet for space utilization

QIU Aihua<sup>1,2</sup>, ZHANG Tao<sup>1</sup>, GU Yidong<sup>1</sup>

(1. General Establishment of Space Science and Application, Chinese Academy of Sciences, Beijing 100094, China;

2. University of Chinese Academy of Sciences, Beijing 100049, China)

**Abstract:** Ethernet for space utilization added time-triggered protocol to a star topology network that was used for Ethernet service; moreover it was fully compatible with the standard IEEE802.3. This Ethernet can effectively solve the problems of real-time, uncertainty and reliability on communications for payload communicating through standard Ethernet. Time-triggered Ethernet was based on global reference time, and the real-time data transmission kept strict timing, supported time and event-triggered communication simultaneously. Time-triggered channel in this Ethernet was suitable for real-time synchronous or periodic message transmission, while event-triggered channel was suitable for sporadic or aperiodic message transmission. To prove the result that Ethernet can ensure the network reliability and security, enhance flexibility, and can greatly improve network availability, many work have to be done. For example, it designed an Ethernet topology and protocol stack for space utilization, described system communication process, studied time-triggered protocol, and simulated synchronization and network performance.

**Key words:** time-triggered; Ethernet; space utilization; real-time; deterministic

美国航空航天局已着手将以太网推广到航天计划中,正在制定空间以太网的标准,猎户座飞船已经采用了确定性的时间触发以太网作为其通信主干网。而空间数据系统咨询委员会(Consultative Committee for Space Data Systems, CCSDS)的航天器在轨接口业务体系结构中也已将以太网作为其数据链路层的一种接口。尽管以太网由于其良好的传输特性一直受到国际、国内航天应用界的关注,但将普通以太网应用到空间安全关键系统中,肯定会有缺陷。

### 1 普通以太网空间应用的缺陷

1) 实时性:普通以太网采用载波监听多路访问/冲突检测机制来解决通信介质层的接入,该机

制会引起通信冲突并导致重发。这种重发机制在网络受到干扰时更为严重,因而不具备强的实时性。

2) 确定性:普通以太网在网络负荷较大时的重发机制增加了网络通信的不确定性。而对于基于交换机的全双工以太网来说,是将报文缓存成队列然后分别转发,那么队列头和尾的发送时间就有差距,对于要求及时响应的通信会带来非常大的不确定性。

3) 可靠性:普通以太网的可靠性依赖于网络节点的可靠,交换机或路由器的性能和传输媒介的稳定,但并未考虑在节点、交换设备以及媒介产生故障之后的网络性能保证问题。针对在空间应用中的使用,TCP/IP 协议(Transmission Control

\* 收稿日期:2014-03-22

基金项目:国家部委资助项目

作者简介:邱爱华(1976—),男,湖北洪湖人,副研究员,博士研究生,E-mail:qpaihua@aoe.ac.cn

Protocol/Internet Protocol) 的重传、恢复和超时机制不完全适用于安全关键系统,而 UDP 协议(User Datagram Protocol)又属于不可靠的面向无连接的协议。

## 2 空间应用以太网的特性需求

### 2.1 实时性需求

实时性是指通信链路上的数据包和控制数据能在指定的时间延迟级别以下被响应,在不同的场合需要达到 ns 级、 $\mu$ s 级。空间有效载荷的通信实时性一般在 100 $\mu$ s 级可以满足任务需求。

### 2.2 确定性需求

确定性是指在最大的时间延迟保证上,确保控制消息的准确到达和载荷数据所需的带宽,其实质是适应当前应用要求的每一次通信运行均具备可预测性和一贯性。常用的 Spacewire 总线、AFDX、TTEthernet 等都将确定性呈现在网络特征的重要方面<sup>[1]</sup>。确定性可分为通信确定性和时间确定性,通信确定性不保证传输响应的一致性,而时间确定性能同时保证数据完整和响应的一致性,完全控制时序抖动。

### 2.3 数据传输特性需求

空间应用上使用以太网的主要目的是借助以太网的灵活性满足载荷接入的方便性和保持一定的高带宽和速率,同时在控制领域具有强实时性。目前,我国航天器内有效载荷通信网络数据传输特性有:①循环周期的规律性,周期性信息(周期性的上行控制指令和数据注入,周期性的下行数字量遥测、工程参数、科学数据等)较多,而非周期信息(随机的上下行数据、突发数据等)较少;②大数据量的传输通常是短时传输,而且有具体时间段;③控制通道的时间精确,可达到毫微秒级,骨干网的时间精度一般在 100 $\mu$ s 级可以满足要求;④骨干网络的规模和节点数在航天器设计初始阶段就可以规划好,能够预知,网络负荷较为平稳;⑤通信协议遵循 CCSDS 要求,数据包大小和长度比较固定。

到目前为止,我国航天器上有效载荷控制与数据通信尚未大规模使用过以太网。国际空间站采用了普通以太网来传输中速载荷数据,使用时间触发以太网的猎户座飞船已经进行了试飞。国外对时间触发协议研究较早,主要在航空和汽车领域,并有相关出版物和标准<sup>[1]</sup>。国内最近几年才开始在航空界有学术研究,主要集中在同步理论阶段,而在航天界还未开展对时间触发协议的

研究。

## 3 网络拓扑与协议栈设计

### 3.1 网络拓扑

空间应用以太网的网络拓扑如图 1 所示,采用双交换机冗余拓扑结构。通过时间触发以太网交换机可以连接时间触发节点和普通以太网节点,节点之间的信息传输相互兼容,并可传输标准以太网和时间触发以太网两种数据。

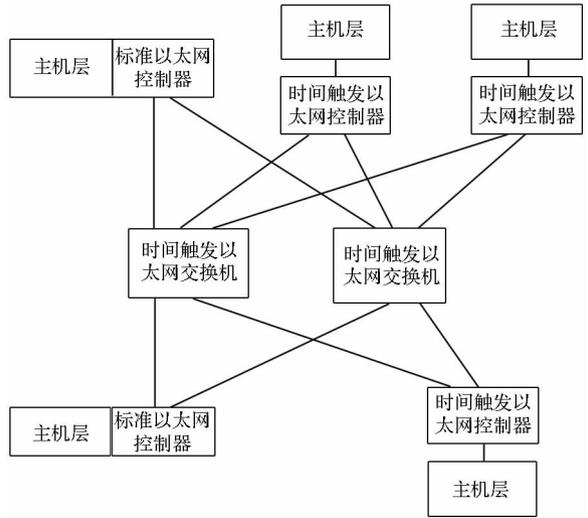


图 1 空间应用以太网的网络拓扑

Fig. 1 Network topology of Ethernet for space utilization

### 3.2 协议栈

空间应用以太网的协议栈集成了标准以太网协议(事件触发)和时间触发协议两种协议,协议栈结构如图 2 所示:

1) 协议栈中以太网中的数据分为实时数据和普通数据两种,实时信道的优先级比标准以太网信道的优先级要高。在链路层(Media Access Control, MAC)之上增加实时调度层,以实时调度普通数据和实时数据,两种数据在实时调度层采用基于优先级的抢占式调度方式;

2) 标准以太网数据采用事件触发,在实时通信通道采用时间触发协议,实时通道的时钟同步采用专门的协议控制帧(Protocol Control Frame, PCF)<sup>[2]</sup>;

3) 时间触发数据报文在预先定义的时刻进行通信,在交换机确认没有时间触发数据报文时,发送普通数据报文;

4) 每个网络节点都按照通信配置表(Message Descriptor List, MEDL)的要求,在规定的时刻开始收发数据。通信配置表可保证每一个共享资源都不相互冲突;

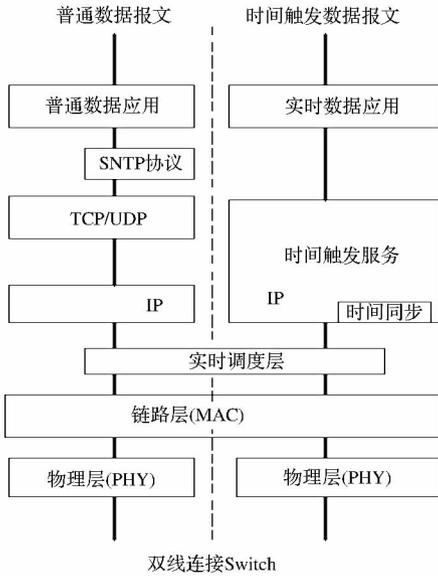


图2 空间应用以太网协议栈结构

Fig. 2 Protocol stack of Ethernet for space utilization

5) 网络具有高的冗余性与可靠性,每个节点双线连接交换机,节点的一个线路若有故障,自动切换到另一个线路连接。

### 3.3 数据流优先级

空间应用以太网提供3种通信业务:PCF、时间触发通信、标准以太网通信,其中PCF又分为冷启动帧(Cold-start frame, CS)、冷启动确认帧(Cold-start Acknowledge frame, CA)、集成帧(Integration frame, IN)。优先级从高到低依次为协议控制帧(最高优先级)、时间触发通信(其次)、标准以太网(最低)。

### 3.4 网络同步过程

#### 3.4.1 启动同步

节点的启动同步是从网络节点上电到加入同步集群中的过程。在节点加电后,同步主机和压缩主机通过CS帧和CA帧进行通信,执行同步操作直到集群同步。我们把发送CS帧和CA帧的过程称为容错握手过程。

#### 3.4.2 IN帧同步

除启动时各节点通过发送CS帧和CA帧同步外,在正常通信过程中,应由同步主机(Synchronization Master, SM)不断地发送IN帧到压缩主机(Compression Master, CM)来维持时间同步。同步分为两步,见图3。第一步,同步主机发送PCF帧到压缩主机。然后压缩主机计算这些PCF帧的到达时间的平均值,发送一个新的PCF帧作为反馈。第二步,压缩主机发送生成的新协议控制帧给各同步主节点和同步客户端。之后各节点进行各自的时钟校正。

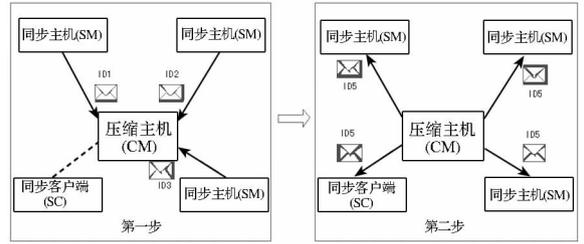


图3 同步操作中的两步同步方法

Fig. 3 Two steps of synchronization

### 3.5 时间触发数据通信

#### 3.5.1 帧传输时刻点

图4是时间触发传输帧 $f_i$ 从一个发送节点经过中间交换机到一个接收节点的过程。 $[V_x, V_y]$ 表示一个链路, $f_i^{[V_x, V_y]}$ 表示在这个链路上传输的帧 $f_i^{[2]}$ 。

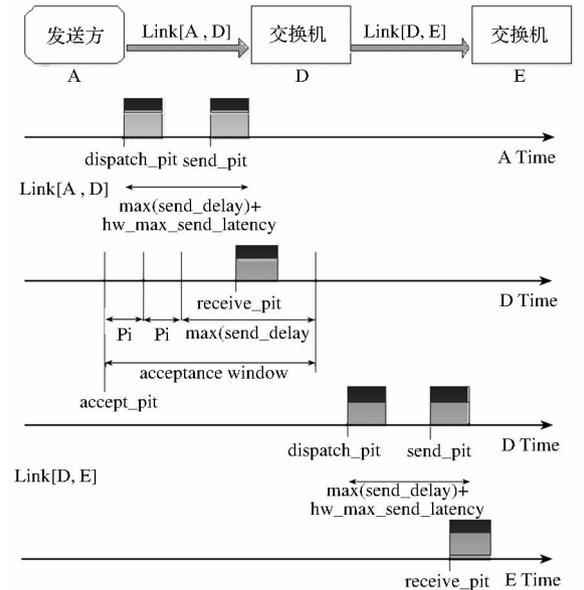


图4 时间触发帧的数据流中的相关时间点

Fig. 4 Time point in data flow of a time-triggered frame

$f_i$ 帧传输过程中的各个时间点如下:①分派时间点(dispatch\_pit),帧 $f_i$ 的分派时间点是静态定义的周期性中断,可触发一个帧的传输;②发送时间点(send\_pit),在该时刻 $f_i$ 帧的开始(Start-of-Frame, SOF)定界符之后的第一个字符的第一位的前沿通过通信链路被传输;③接收时间点(receive\_pit),帧 $f_i$ 的接收时间点是指接收帧的第一位的接收时刻;④开始接收窗口时间点(accept\_pit),用于交换机检查作为时间触发传输帧时间的正确性。

#### 3.5.2 数据通信过程

时间触发协议要求网络中的每一个终端和交换机都按照通信配置表的要求,在规定的时刻开

始收发数据。整个网络的通信配置表采用离线方式生成,并保证每一个共享资源都不相互冲突。在网络中每个终端在某一时刻最多只有一条消息需要发送,不会出现多条消息竞争使用同一输出链路的情况。

图 5 中有 3 个节点通过交换机进行互联。节点 1 和节点 2 分别发送时间触发消息到节点 3。节点 1 发送消息的周期为 3ms,节点 2 发送消息的周期为 2ms,节点 3 接收到的数据序列如图 5 所示。

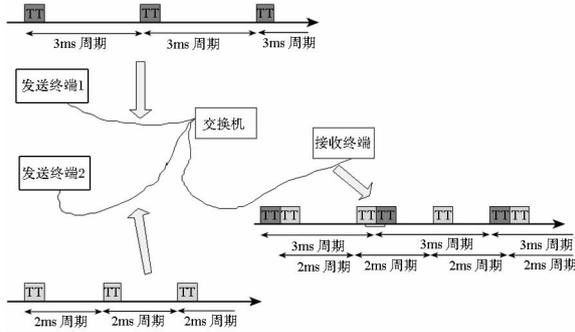


图 5 时间触发通道数据流

Fig. 5 Data flow of time-triggered channel

## 4 时间触发通信协议

### 4.1 基本参数

#### 4.1.1 全局基准时间

全局基准时间是时间同步的基础和唯一依据,采用统一时间格式,用两种参数来表示:颗粒度 (granularity) 和跨度 (horizon)。颗粒度表示两个相邻的数字时钟片之间的间隔,也就是可以按时间格式进行测量的最小时间间隔,跨度就是时间环绕流逝的瞬间。时间格式是一个基于秒的 64 位二进制数。

#### 4.1.2 数据帧格式

以时间触发方式传输的帧仍采用标准以太网的数据帧,在帧中增加了时间触发的内部标识和域类型。时间触发数据以太网类型的值为 0x88d7,事件触发数据以太网类型的值为 0x8800。

#### 4.1.3 节点定义

网络中存在三种节点,同步主机指网络中参与 IN 帧同步计算的终端节点。压缩主机指网络中参与压缩计算的交换机,而其他节点 (可能是交换机或终端节点) 称作同步客户端 (Synchronization Client, SC)。

#### 4.1.4 节点的透明时钟

透明时钟就是从发送节点到接收节点传输的

累积延时,时间以 PS 为单位,包括动态传输延迟、静态传输延迟和线路延迟等。各节点采用透明时钟的值存储在 PCF 帧的透明时钟域中。发送节点、中继转发节点和接收节点,它们的透明时钟可统一成式(1)计算, $i$  为节点的编号:

$$\begin{aligned} \text{pcf\_transparent\_clock}_i = & \\ & \text{pcf\_transparent\_clock}_{i-1} + \text{dynamic\_relay\_delay}_i + \\ & \text{static\_relay\_delay}_i + \text{wire\_delay}_i \end{aligned} \quad (1)$$

其中, $\text{pcf\_transparent\_clock}_i$  为当前节点  $i$  的透明时钟, $\text{pcf\_transparent\_clock}_{i-1}$  为第  $i-1$  个节点的透明时钟值, $\text{dynamic\_relay\_delay}_i$  为节点  $i$  的动态传送延迟, $\text{static\_relay\_delay}_i$  为节点  $i$  的静态传送延迟, $\text{wire\_delay}_i$  为节点  $i-1$  到节点  $i$  的链路延迟。

#### 4.1.5 网络最大传输延迟

最大传输延迟  $\text{max\_transmission\_delay}$  表示网络中所有发送节点到接收节点间的最大传输时间延迟。 $\text{max\_transmission\_delay}$  是离线时的统计值,如式(2)所示,一个集群内的每个节点将使用同样的值。

$$\begin{aligned} \text{max\_transmission\_delay} = & \\ & \text{max}(\text{pcf\_transparent\_clock}_n) \end{aligned} \quad (2)$$

#### 4.1.6 压缩算法延迟时间

压缩算法延迟时间为压缩主机在执行压缩算法时的时间开销。计算公式见式(3)。

$$\begin{aligned} \text{compression\_master\_delay} = & \\ & \text{max\_observation\_window} + \\ & \text{calculation\_overhead} + \text{dispatch\_delay} \end{aligned} \quad (3)$$

其中, $\text{max\_observation\_window}$  为最大观察窗口的值。 $\text{observation\_window}$  为压缩算法观察窗口,可进行离线配置。 $f$  为可容纳的故障同步主机数,最大观测窗口的计算如式(4)所示。

$$\begin{aligned} \text{max\_observation\_window} = & \\ & (f+1) \times \text{observation\_window} \end{aligned} \quad (4)$$

$\text{calculation\_overhead}$  为压缩算法的计算延时, $\text{dispatch\_delay}$  为压缩主机发送帧的时延。

#### 4.1.7 PCF 帧预定接收时间

压缩主机中的 PCF 帧预定接收时间<sup>[3]</sup>由式(5)计算:

$$\begin{aligned} \text{cm\_scheduled\_receive\_pit} = & \\ & \text{sm\_dispatch\_pit} + \text{max\_transmission\_delay} + \\ & \text{compression\_master\_delay} \end{aligned} \quad (5)$$

同步主机/同步客户端中的 PCF 帧预定接收时间由式(6)计算。

$$\begin{aligned} \text{smc\_scheduled\_receive\_pit} = \\ \text{sm\_dispatch\_pit} + 2 \times \text{max\_transmission\_delay} + \\ \text{compression\_master\_delay} \end{aligned} \quad (6)$$

其中,  $\text{sm\_dispatch\_pit}$  为同步主机发送 PCF 的发送时间,  $\text{max\_transmission\_delay}$  为最大传输延迟,  $\text{compression\_master\_delay}$  为压缩算法延迟。

## 4.2 基本算法

### 4.2.1 时序保持

时序保持的主要目的是对接收到的数据帧进行重新排序,以便和各帧的发送顺序保持一致,在各类节点中均执行。receive\_pit 指接收节点物理上接收到 PCF 的时间点,而 permanence\_pit 指经过消息时序保持算法后得到的保持时间点,压缩算法采用的是 permanence\_pit 而不是 receive\_pit,因而需进行时序保持计算。

时序保持算法主要的计算过程为:①接收节点(压缩主机/同步客户端/同步主机)接收到 PCF,读取本地时钟值记为 receive\_pit,并启动消息时序保持算法;②读取 PCF 的透明时钟域得到透明时钟值 pcf\_transparent\_clock,即为 PCF 在网络中传输的实际总延迟;③计算保持延迟 permanence\_delay;④计算保持时间点 permanence\_pit。

### 4.2.2 压缩

压缩是在 CM 中执行的算法,CM 中收集到的各 PCF 保持时间点之间的时间差可以真实代表发送原 PCF 的 SM 本地时钟之间的时间差。压缩算法根据在配置时间槽内的各 PCF 的保持时间点,计算出压缩时间点 ( $\text{cm\_compressed\_pit}$ )。最优压缩时间点反映了网络内各 SM 节点本地时钟的平均值<sup>[4]</sup>。CM 根据最优压缩时间点向 SM/SC 发送新的 PCF,作为 SM/SC 同步时钟调整的基准。在 CM 中,由最优压缩时间点和预计接收点之差来校正 CM 的本地时钟。

压缩算法针对的帧类型是 IN 帧和标准集成模式下的 CA 帧,对于 CS 帧和高度集成模式下的 CA 帧则不压缩,参见表 1。

表 1 压缩算法计算的帧类型<sup>[2]</sup>

Tab.1 Frame type computed in compress algorithm

	CS 帧	CA 帧	IN 帧
标准集成同步主机	不压缩	压缩	压缩
高度集成同步主机	不压缩	不压缩	压缩

压缩算法分为 3 个阶段。①采集阶段:在

PCF 被保持后,如果此时没有压缩算法处于采集阶段,则开启新的压缩算法。此时压缩算法开启一个观测窗口(Observation Window, OW),如果集群内所有节点都同步时,所有 SM 发送的集成周期相同的各 PCF 保持点可以在第一个 OW 内收集完全。②计算阶段:采用容错平均算法去计算压缩的时钟校正值  $\text{compression\_correction}$ <sup>[2]</sup>。③延迟阶段:继计算阶段之后,根据延迟  $\text{compression\_correction}$  的时间长度,得到压缩点  $\text{cm\_compressed\_pit}$ 。

### 4.2.3 时钟校正

同步主机/客户端以及压缩主机都执行时钟校正算法,在接收窗口结束时,同步主机和同步客户端选择在新成员变量 pcf\_membership\_new 参数中具有最高位数据的 PCF 作为每一个通信的链路。在遇到多个 PCF 含有同样的 pcf\_membership\_new 位数据时,选择的 PCF 就是在接收窗口内最新的 smc\_permanence\_pit 时间点的那个,或者是最靠近接收窗口结束的数据帧。那么,同步主机和同步客户端的时钟校正可由预计 PCF 接收时间点  $\text{smc\_scheduled\_pit}$  和该 PCF 所携带的参数来计算。

压缩主机挑选所有能接收到 PCF 帧的信道中最好的那个信道,假设在该信道上接收的 PCF 帧经过压缩算法之后,选取  $\text{max}(\text{cm\_compressed\_pit})$ 。压缩主机将会使用含在 pcf\_membership\_new 中位设置最多的 PCF 以及在一系列 pcf\_membership\_new 中最近的那个 PCF,这里  $\text{cm\_compressed\_pit}$  为压缩主机完成压缩时刻。同样,压缩主机根据 PCF 预定接收时间点  $\text{cm\_scheduled\_receive\_pit}$  和该 PCF 携带的参数来计算时钟校正。

根据计算出的时钟校正  $\text{clock\_corr}$  值,压缩主机、同步主机和同步客户端的本地时钟  $\text{local\_corr}$  校正见式(7)。

$$\text{local\_corr} = \text{local\_corr} + \text{clock\_corr} \quad (7)$$

## 5 实验

采用基于组件的模块化网络仿真平台(Objective Module Network Testbed in C++, Omnet++)和网络仿真框架(Internet Framework, INET)进行网络仿真验证,INET 提供了标准以太网的协议模型<sup>[5]</sup>,实验在此基础上设计了面向空间应用的时间触发以太网的仿真协议模型,如图 6 所示。

仿真模型在 INET 模型的 MAC 层增加了时

间触发调度的链路扩展层,用于调度普通数据和时间触发数据,而交换机模型只需设计到链路层即可,其余网络节点模型采用对等设计。

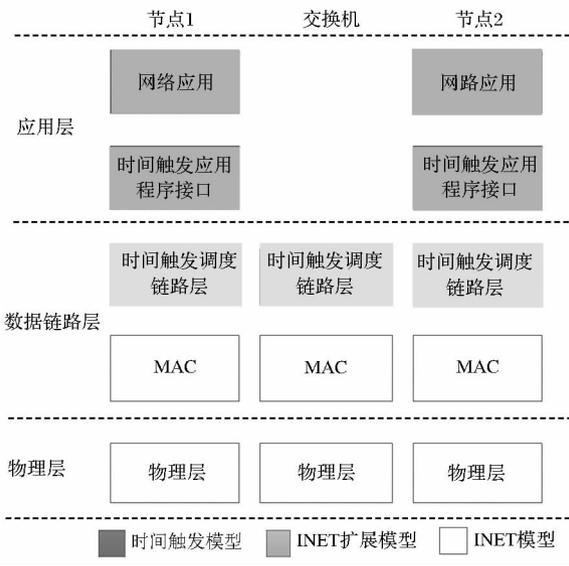


图 6 仿真协议模型

Fig. 6 Simulation protocol model

### 5.1 同步性能仿真

建立含有 7 个标准集成度的同步主机、2 个交换机构成的中等规模网络,如图 7 所示。

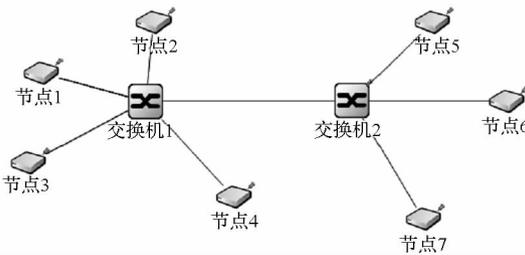


图 7 同步性能仿真网络

Fig. 7 Simulation network of synchronization performance

网络的通信时间槽设置为 80ns,各节点和交换机的时钟精度设置为 80ns,PCF 同步 IN 帧长度为 72Byte,各节点、交换机和链路的硬件延迟为 0,集群周期设为 1ms。网络传输速率为 100Mbps,仿真时间设定为 20min。仿真完成后,统计 2 个交换机的同步操作网络占用率如图 8 所示。

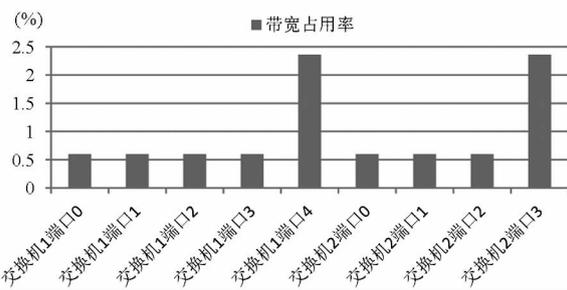


图 8 交换机同步操作的网络占用率

Fig. 8 Network occupancy of switch synchronizing

同步操作带宽占用率较大的为交换机 1 的端口 4 和交换机 2 的端口 3,因为这 2 个端口是级联端口,但仅为 2.36%。可以看出同步操作占到交换机的带宽比例很小,平均为 0.98%。

在同步操作过程中,每个节点和交换机的处理 IN 帧的延迟如图 9、图 10 所示。可以看到,交换机进行 IN 帧接收和发送的延迟均在  $5\mu\text{s} \sim 25\mu\text{s}$  之间,压缩计算导致的延迟不超过  $20\mu\text{s}$ 。各个节点发送 IN 帧不占延迟时间,而接收处理 IN 帧延迟在  $20\mu\text{s} \sim 27\mu\text{s}$  之间,各节点时钟同步延迟不超过  $27\mu\text{s}$ 。可以看到该网络可以满足空间以太网的时间精度  $100\mu\text{s}$  内的需求。

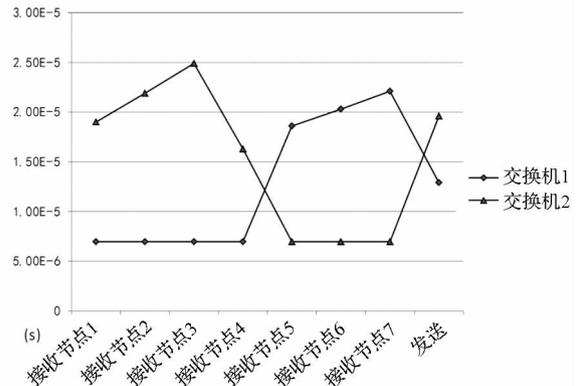


图 9 交换机接收各节点 IN 帧的延迟和发送 IN 帧的延迟

Fig. 9 Latency of IN frame switching in switch

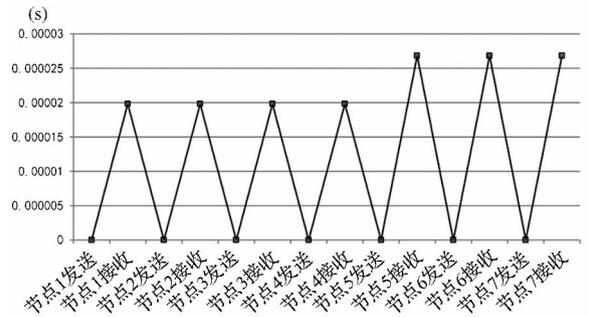


图 10 各个节点发送 IN 帧和接收 IN 帧的延迟

Fig. 10 Latency of IN frame switching in nodes

### 5.2 网络性能仿真

建立以 3 个标准集成度的同步主机、1 个交换机构成的网络,如图 11 所示。

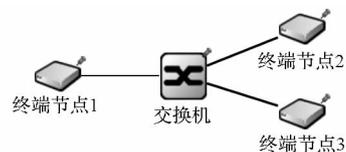


图 11 网络性能仿真网络

Fig. 11 Simulation network of network performance

网络的通信时间槽设置为 80ns,各节点和交

交换机的时钟精度为 500ns, 集群周期为 5ms, 网络链路速率为 100Mbps, 各节点和交换机的硬件延迟设置为 0。节点 1 发送时间触发报文给节点 2 和节点 3, 数据包长度为 46Byte, 同时发送普通数据 1500Byte 的普通数据包, 发送间隔采用在 (200 $\mu$ s, 500 $\mu$ s) 之间的随机间隔时间。节点 2 发送时间触发报文给节点 3, 数据包长度 46Byte, 同时发送普通数据 1500Byte, 发送间隔采用在 (200 $\mu$ s, 500 $\mu$ s) 之间的随机间隔。节点 3 发送给节点 1 和节点 2 的 1500Byte 的普通数据, 发送间隔采用在 (200 $\mu$ s, 500 $\mu$ s) 之间的随机间隔, 仿真运行时间 20min。交换机和 3 个节点进行时间触发数据包通信的延迟如图 12 所示。

图 12 中最下面的是节点 1 的通信延迟, 为 10 $\mu$ s; 中间是交换机的延迟, 为 35 $\mu$ s; 最上面 2 条线是节点 2 和节点 3 的延迟, 为 60 $\mu$ s, 说明在同时进行高负载的普通数据和时间触发数据通信时, 网络性能仍可以满足 100 $\mu$ s 内的实时性和确定性需求。节点 2 和节点 3 时间触发通信延迟大是因为接收的时间触发报文和普通数据报文比节点 1 要多。同时, 可以看到各个节点和交换机一旦同步后, 时间触发通信的延迟就已经基本固定, 这就是时间触发网络确定性的基本特性。

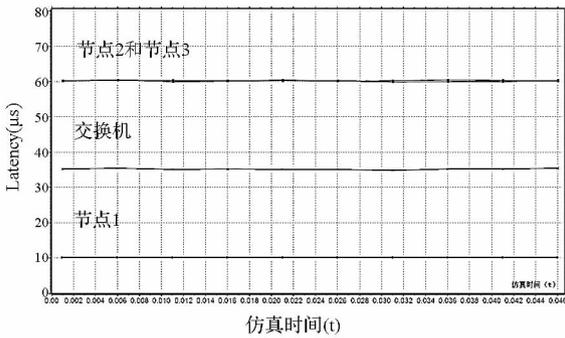


图 12 交换机和 3 个节点进行时间触发数据包通信的延迟  
Fig. 12 Latency of time-triggered communication in switch and nodes

对该网络的利用率进行分析, 如图 13 所示。

节点 3 可达 71%, 节点 1 为 67.8%, 整个网络的平均利用率 69.5%, 高于标准以太网的常用利用率范围。而实际空间应用有效载荷通信中, 普通数据量通常较少, 网络带宽利用率可进一步提高。

## 6 结论

针对普通以太网在空间应用有效载荷通信中的缺陷, 可在标准以太网的协议中增加时间触发

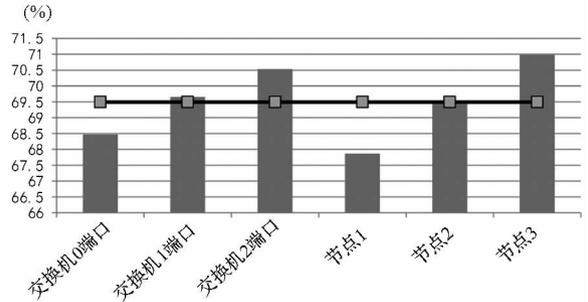


图 13 网络利用率分析

Fig. 13 Analysis of network utilization

协议来解决普通以太网空间应用的可靠性、实时性和确定性问题, 设计了采用时间触发协议的空间应用以太网的网络拓扑结构和协议栈, 并介绍了系统的通信过程, 研究了时间触发以太网的协议, 并进行了同步性能和网络性能仿真实验。

面向空间应用的时间触发以太网可以满足对时间和负载要求完全可预测的通信, 时间延迟的抖动是确定的。系统架构能有效隔离网络资源的关键和非关键功能, 支持从系统级进行所有计算和网络资源的分区, 为硬实时、低实时的数据并行传输提供统一的网络, 其带宽利用率大大高于普通以太网。为强实时、高安全、多数据流的有效载荷领域使用以太网提供了技术解决途径。

## 参考文献 (References)

- [1] Obermaisser R. Time-triggered communication[M]. US: Florida: CRC Press, 2012.
- [2] AS6802, Time-triggered Ethernet[S]. SAE, 2011, 11.
- [3] 罗安心. 基于时间触发以太网的同步算法研究与实现[D]. 成都: 电子科技大学, 2013.  
LUO Anxin. Research and implementation synchronization algorithm based on time-trigger Ethernet [D]. Chengdu: University of Electronic Science and Technology of China, 2013. (in Chinese)
- [4] 郭爱英. 时间触发以太网同步技术的研究与实现[D]. 成都: 电子科技大学, 2013.  
GUO Aiying. Research and implementation of TTE synchronization technology [D]. Chengdu: University of Electronic Science and Technology of China, 2013. (in Chinese)
- [5] Steinbach T, Kenfack H D, Korf F, et al. An extension of the OMNeT++ INET framework for simulating real-time Ethernet with high accuracy [C]. Proceedings of the 4th International ICST Conference on Simulation Tools and Techniques, 2011.
- [6] 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.  
LIU Wanchun, LI Qiao, HE Feng, et al. Research on time-triggered Ethernet synchronization and scheduling mechanism[J]. Aeronautical Computing Technique, 2011, 41(4): 122 - 127. (in Chinese)