一种反射内存网航电软件测试环境实时通讯协议

王立泽，刘 斌，周礼亮

（北京航空航天大学 工程系统工程系，100191 北京，wanglize@dse.buaa.edu.cn）

摘 要：为解决航电软件仿真测试环境中的通用异步和周期实时通讯问题，设计了一种反射内存网实时通讯协议（MSTP）. 基于反射内存网 SCRAMNet 设计协议，介绍了协议内存布局，描述并实现周期和异步数据的发送和接收过程，对协议进行测试以表明设计可行性. 最后，给出了较为完善的系统开销计算模型.

关键词：反射内存网；周期通讯；异步通讯；通讯协议；仿真测试环境；航电软件

中图分类号：TP 311.52 **文献标志码**：A **文章编号**：0367-6234(2011)03-0122-06

A kind of real-time communication protocol based on reflective memory network for the simulation testing environment of avionic software

WANG Li-ze, LIU Bin, ZHOU Li-liang

(Dept. of System Engineering of Engineering Technology, Beijing University of Aeronautics and Astronautics, 100191 Beijing, China, wanglize@dse.buaa.edu.cn)

Abstract: A kind of real-time communication protocol named Message-Stream Test Protocol (MSTP) and based on reflective memory network, is designed for the Simulation Testing Environment of Avionic Software to solve the common communication problem of periodic and asynchronous communications. The MSTP is designed on SCRRAMNet; the layout of memory is introduced; and the sending and receiving process are described. The testing result demonstrates that the design is feasible. At last, a more precise model of system cost is presented.

Key words: reflective memory network; period communication; asynchronous communication; communication protocol; simulation testing environment; avionic software

在国内外航空航天等领域，仿真测试环境多采用 Hardware-In-Loop(HIL)方式和分布式结构来分散繁重的计算任务，测试实时嵌入式软件，进而验证软件可靠性[1]. 而如何保证分布式测试环境各节点间的实时通信则是一个难题[2].

近几年高速网络技术有 Fibre Channel(FC)、Gigabit Ethernet(GigE)、反射内存网等，而反射内存网因用硬件实现通信、传输延迟确定、移植性强等优点，成为解决分布式测试环境节点间的实时通信问题的一种行之有效的方法. 关于反射内存网的研究主要集中于协议设计[3-8]及理论研究[9-10]两方面. 协议设计在功能实现上可分为两类：一类实现了支持一定程度异步、周期通讯模式的通讯协议，但它们都与应用绑定，不能灵活支持系统中不同种类的复杂通信，离开应用，协议需重新实现，缺乏通用性[3-6]，如文献[4]为解决周期同步和通信问题进行了设计，协议建立连接后，通讯双方在约定好的位置传输某种确定的数据如同步数据，传输模式不能配置并伴随整个系统运行过程；另一类虽然实现了一定的通用性，但只支持异步通信，缺乏特殊的优先级机制，也不支持周期通讯[7-8]. 理论研究主要基于一定的计算模型计算实时网络的开销[9-10]. 通过分析，现有基于反射内存网的相关协议设计，都不能满足本文分布式测试系统的通用通信要求.

本文研究并利用反射内存网络 SCRAMNet 设

收稿日期：2009-08-27.
基金项目：航空科学基金资助项目(20095551025).
作者简介：王立泽(1979—)，男，博士研究生；
刘 斌(1967—)，男，教授，博士生导师.

计一种能实现周期和异步通讯的、通用性较强的实时通讯协议，并给出了较为完善的系统开销算法，以解决分布式测试环境节点间的实时通信问题.

1　测试环境通讯需求分析

在分布式测试环境中，测试环境结构分为两层：测试开发层和测试执行层[1]，见图 1. 测试执行层存在原子事务、时钟同步、任务同步、通信等问题. 原子事务的解决主要靠在各节点设置事务协调者，使之维护本地日志，并负责启动、分派、结束事务的执行，各事务协调者遵循一定协议通信[2,11]，如各节点对测试开始、结束的一致响应. 其它问题与原子事务类似，靠节点按即定算法周期或异步协作. 节点间任何通信都要借助网络完成，通讯类型多，实时性高，且处理上有缓急之分. 有的任务如仿真任务随待测系统不同而变化，因此还要求通讯有一定的配置灵活性. 这就需要一个专门的实时通信层来适应节点间的通讯要求.

综合以上分析，为解决分布式测试系统的通信问题，需要基于实时网络为实时通信层设计一种通用的实时通讯协议，满足诸多任务的协作、通信.

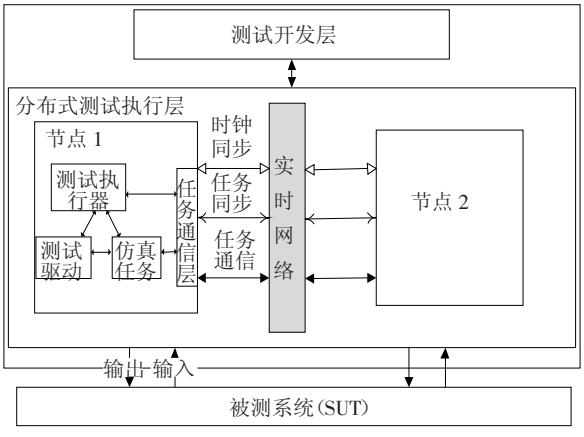


图 1　分布嵌入式仿真软件测试环境结构

2　实时通讯协议的设计

2.1　网络原理

反射内存网是实时性要求较高的专用通讯网络，它的工作原理[12]是各实时节点有反射内存卡，主机内存映射到各卡的独立局部内存，局部内存映射到全局网络内存，使各卡逻辑上共用一段地址. 各节点将数据写入主机内存同时也写入所有其他各卡内存. 全局内存对所有节点都可见，从而实现节点间数据通讯.

2.2　设计思路

经需求分析知，分布式测试系统中既有较多周期通讯，又有灵活异步通讯需求，对异步通讯，不但强调实时性，某些类型消息还应优先处理，如测试指令消息应先于数据消息接收. 本文基于 SCRAMNet 实现了一种支持周期和异步数据传输的协议，异步传输基于消息和优先级机制，周期传输基于数据流机制，称实现为 Message – Stream Test Protocol(MSTP).

本协议的设计目的及特点主要有：

1）能够实现测试系统任务间的异步通讯传输原语，发送消息，查询消息，接收消息；

2）允许两任务间异步通讯有一定持久性，即通过合理分配内存，允许发送多条消息，接收与发送顺序一致；

3）允许异步通讯有优先级机制，使某类通信能够被优先处理；

4）能够实现任务间的周期性通讯，且不同的协议配置，传输不同的周期数据.

2.3　内存布局

内存布局解决的问题是如何为 MSTP 进行内存分配，使协议对用户透明.

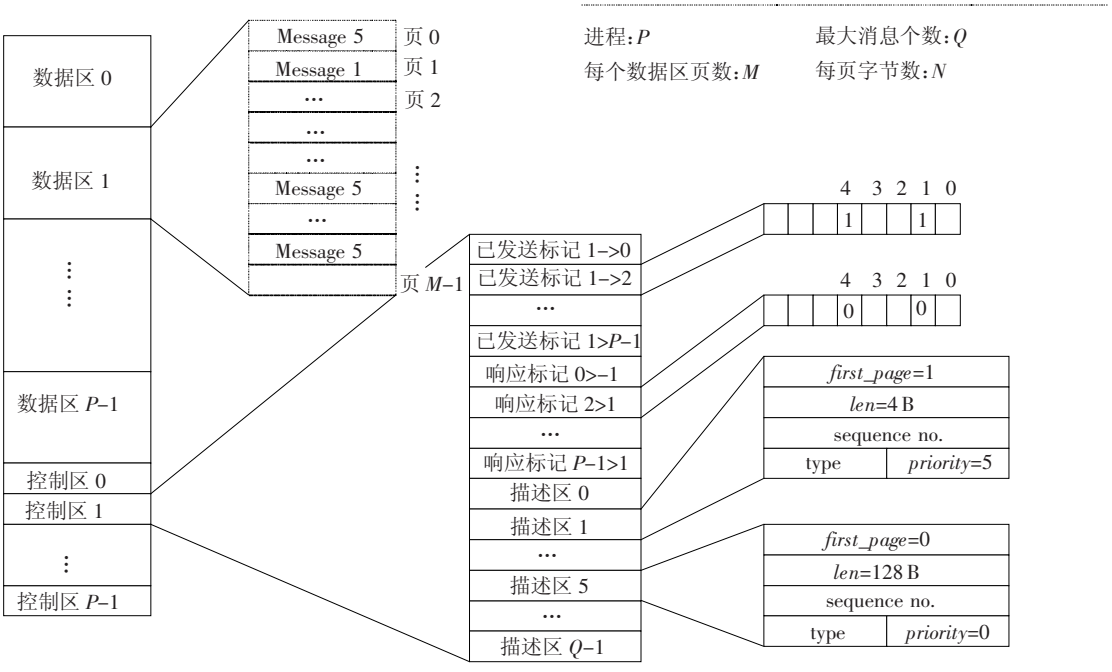
根据反射内存网的广播特点，同一内存区不能被多个用户同时写操作，如同时写则必须实现加锁机制，而加锁会增加开销，所以，协议的核心设计思想是为每个进程开辟一块写操作区域，全局只有此进程拥有写操作权，即一块内存区有且只能有 1 个写操作者. 设参加通讯的进程数为 *proc_ num.*

为实现上述思路，对内存进行访问控制，将其分为 2 个区域：数据区和控制区. 为实现异步通讯，为每个进程分配各自的数据区和控制区，数据区存放消息，控制区存放各消息对应的控制信息，见图 2. 为实现任意两进程间的周期通讯，分配 1 个全局的数据区和 1 个通讯登记区，见图 3.

异步通讯数据区由所属进程管理并执行写操作，当有异步消息发送时，发送进程申请并将消息写入数据区内存，发送后，目的进程只能读取消息. 为实现优先级机制，消息数据区按页(page)分配. 当优先级不同的消息共存时，高优先级消息先被取走而导致在数据区产生外部内存碎片，为每个消息分配一定数量页面可避免碎片产生. 每条消息可分配多页，呈链表结构，每页头部的页指针指向下一页. 布局示意见图 2.

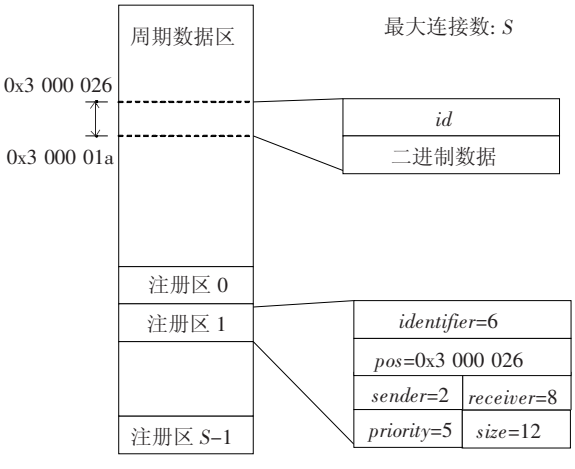
与消息数据区同属发送进程的控制区划分为 3 个子区域：消息描述区，已发送标记区(send)，响应区(ack). 每个进程最多发送某一固定数量的消息，记为 *msg_ num.* 每个消息对应 1 个消息

描述区,分别表示所属进程的从第 1 到第 *msg_ num* 个消息的位置和属性信息,包括首页 (*first_ page*), 长度,发送序号,类型,优先级等. 首页记录起始页号. 长度表示消息字节数. 另外,控制区还各含有 *proc_ num* − 1 个 send 标记和 ack 标记,每个标记有 *msg_ num* 个位. 每个 send 标记代表源进程向目的进程发送的消息序列,自左至右的位序号对应了消息序号,代表源进程是否向目的进程发送消息,同理,ack 标记则代表消息是否已被目的进程接收. 消息描述区、send 区、ack 区,同消息数据区一样,在协议初始化后,大小位置都固定不变.



说明:进程 1 向 2 发送 6 个消息时的状态. 消息 1,5 还未被接收,因 1 优先级低,5 将先于 1 被接收.

图 2 MSTP 异步通讯内存布局



说明:进程 2 与 8 建立 12 B 周期通讯内存布局

图 3 MSTP 周期通讯内存布局

由于测试配置完成后,频繁通讯的周期任务间数据关系确定且测试中不改变,控制约束少于异步通讯,故采用共享内存方案,全局分配 1 个通讯登记表和 1 个数据区. 通讯登记表用于通讯双方确立通讯关系. 假定最大通讯连接数为 *perd_ num*,则登记表包含 *perd_ num* 项登记信息, 包括通讯标识,源,目的,周期,数据区位置,大小等. 每次登记由协议在数据区分配 1 块内存区域. 示意见图 3.

2.4 协议描述

协议描述规定了用户如何使用 MSTP 及其实现的原理. 对异步通信,MSTP 要求明确所有参与通信的进程数量. 每个进程对应 1 个逻辑地址,即在进程集中的编号.

API 定义如下:

1) mstp_ asyn_ init(int *proc_ logic*, int *proc_ num*), 异步通讯初始化. 参数分别为异步进程逻辑地址和异步进程数. 返回指针 struct mstp *,以后进程 *proc_ logic* 将用此指针收发消息. 初始化指针包括了进程逻辑地址,消息数据区和控制区地址及通讯状态等信息;

mstp_ perd_ init( ),周期通讯协议初始化;

mstp_ perd _regist(int *id*, struct reg * *info*), 登记 1 个通讯区, *info* 包括通讯标识、周期等信息.

2) mstp_ asyn_ send(struct mstp *, char * *buf*, int *len*, int *dest*, short *msg_ type*, short *priority*); 向 dest 进程发送缓冲区 *buf* 内的 *len* 字

节的数据，*msg_type* 是用户定义的消息类型. 默认为最低优先级；

mstp_perd_send (int *id*, char * *buf*, int *size*)；写数据流 *buf* 到通讯标识 *id*.

3) mstp_asyn_inquiry(struct mstp * *this*, int *source*)；查询是否有从 *source* 发送至 *this* 的消息，若有，返回进程逻辑地址，反之 -1；若 *source* 为 -1，查询是否有从所有任务发送至 *this* 的消息.

4) mstp_asyn_recv(struct mstp * *this_proc*, char * *buf*, int *max_len*, int *source*, short &*msg_type*)；从 *source* 接收 *max_len* 字节消息至 *buf*. 先取得高优先级的消息，同优先级消息按发送顺序取得.

mstp_perd_recv(int *id*, char * *buf*)；读数据流.

源进程向目的进程发送异步消息时，协议内部动作：1) 按消息大小申请空闲内存区，得到 1 个或多个内存页，根据页号为每页的指针头赋值；2) 将数据拷贝至内存数据区；3) 根据目的进程的 send 标记和 ack 标记，计算出消息缓冲区号，根据通讯状态得出消息发送序号；4) 将新的 send 标记和消息描述信息写入控制区. 发送端协议自动计算消息序号. 发送端节点维护其对应控制区信息.

进程接收源进程发来的异步消息时，协议动作：1) 根据源进程的 send 标记和 ack 标记，得到未接收消息列表；2) 读取消息列表相应描述区信息，根据优先级和发送序号，得出应读消息号；3) 将消息读出；4) 置 ack 标记.

由于本地内存操作比读写反射内存快，所以协议内部的实现应尽可能减少反射内存读写. 可利用状态信息实现.

周期通讯的关键在于通讯登记表的建立. 双方经过通讯标识确立通讯关系，协议负责为通讯双方在数据区维护内存资源. 在通讯配置建立时，发送方协议通过分布式互斥算法向登记区增加记录，然后通过异步消息通知目的方获取记录项. 通讯过程中，为保证实时性能，协议不允许改变登记表.

2.5　协议实现

实现 MSTP 需确定如下参数：异步通讯进程个数 *proc_num*，每个进程发送最大消息数 *msg_num*，每个进程消息数据区长度 *data_area_len*，消息数据区最小内存单元 *page_size*，最小内存单元指针头 page_head，消息描述区 data_desc_size，周期通讯连接数 *perd_num*，登记表项大小 *reg_size*. 优先级数、消息类型、周期通讯数据区大小 *perd_data_size* 根据应用确定. 确定参数后，协议实现异步消息通讯需要的实时网络内存为

$$proc_num \times \left(data_area_len + \frac{2 \times (proc_num - 1)}{8} + data_desc_size \times msg_num\right).$$

实现周期通讯登记表需要的内存为

$$perd_mum \times reg_size + perd_data_size.$$

本文在测试环境中实现了 MSTP：异步通讯进程数 32，每进程数据区 1 M，最大消息数 64，最小内存单元 48 B，指针头 4 B，消息描述区 16 B，优先级别 10 级，最高 0 级. 周期通讯最大连接数 300，登记表项 24 B，数据区 2 MB，计算得实际使用实时网络内存 35 691 800 B.

协议初始化后，为异步通讯每个进程分配的指针维护了指向数据区和控制区、状态信息数据结构的指针. 为避免频繁访问反射内存，状态信息将最近一次操作反射内存的数据缓存至其中，包括消息数据区内存缓存、发送标记、序号、已探测到但还未接收的消息等. 发送和接收进程根据 send 标记和相应 ack 标记是否相等判断此消息号是否在已发送队列中，若相等，说明其空闲.

2.6　测试

为检验 MSTP 设计效果，在分布式测试环境下设计了周期和异步通讯性能测试实验. 分别位于 2 个实时节点的多对仿真任务并发运行，对异步和周期通讯，依次对每种负载持续测试 1 s 以验证是否有丢包或碰撞现象. 异步消息优先级为 0 或 5. 周期通讯周期为 1 ms. 测试负载为 4 B，64 B，…，16 kB，32 kB. 为每种测试负载分配 1 对异步、周期任务，对比发送与接收的数据以判断通讯正确性. 测试设备为 Pentium4 2.8GHz PC 机，OS VxWorks5.5，PCI 64 SCRAMNet，示波器. 实验数据如图 4 所示.

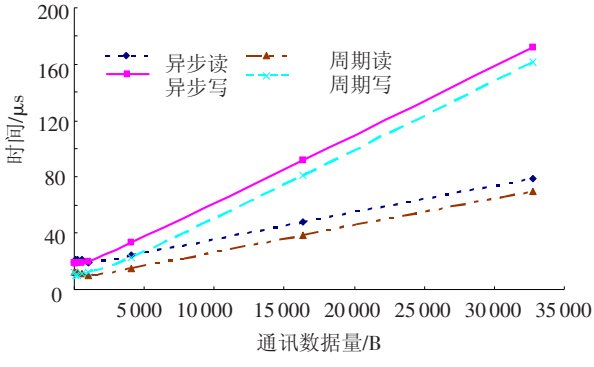


图 4　MSTP 测试数据

根据实验作以下分析:1)MSTP 的异步和周期模式在并发运行、不同数据量、多次测试情况下,无丢包现象,均正常工作,且未知所携带的数据含义,与应用分离;2)文献[4]未给出性能测试数据,但应用系统的周期和异步协议工作在 25 ~ 40 Hz 间;文献[8]使用异步协议实现仿真应用的周期和异步数据传输,测得 3 节点网络系统的性能约为 300 μs/800 B, 320 μs/1600 B;文献[5]实现用周期和异步方式传输 HLA 联邦成员,性能在 70 μs ~ 30 ms 之间. 而本文提出的 MSTP,在具备通用性的前提下,从图 4 实验数据可看出也具有强实时特性,满足大多数应用的实时性要求;3)异步通讯中优先级高的消息总是先于低的被接收;4)传输延迟大小受数据包大小影响很大,因此要获得强实时性,需尽可能减小每次通讯的数据量.

实验结论:MSTP 通用性设计可行,较已有设计有更广适用范围,且实时性强,可靠性高,能够基于其解决分布式测试环境的各种协议实现问题.

3 理论分析

文献[10]采用估算法设计了 1 个专用于反射内存网络的根据各节点的访问次数开销计算系统总开销的计算模型,本文进一步完善文献[10]中关键量的算法,得到更为精确的系统总开销算式.

文献[10]假设节点 i 各进程内存区大小未知,且各进程读/写内存次数比率同为 $\beta(i)$,所以在计算总开销时采用估算法,设 i 所有进程每次读写操作时访问 i 的全部内存区 $S(i)$. 而在本文及实际应用中[3-6],文献[10]的假设会带来网络访问开销的增加和效率的降低,相反,节点 i 的各进程 j 一般不共享使用 $S(i)$,而是使用各自内存 $s(j)$,内存相互独立、大小不同,且读/写次数比率不同,记为 $\beta(j)$,以提高网络内存访问效率. 因而该估算法会因缺乏准确性而带来误差,本文综合以上因素,给出更为符合实际的计算模型改进算法. 计算原理及关键量的改进算法如下.

规定一个时间段 T_{lcm},在 T_{lcm} 内所有访问反射内存的进程必须至少执行 1 次,且假定有 R 个节点,节点 i 上有 j 个进程,各进程执行时间 $\tau(j)$ 已知. 这样,

$$c(i) = N(i,j) \times j \times S(i) = \sum s(j).$$

式中:$c(i)$ 为 i 节点的内存区访问开销;$N(i,j)$ 为平均每个进程 j 对单位数据的访问次数;$S(i)$ 为节点 i 各进程内进程内存和. 各节点内存访问开销和为系统总开销.

根据节点 i 上各进程的执行时间 $\tau(j)$ 等量,求得 i 的平均单元数据访问时间 $\Delta(i)$:

$$\Delta i = \left[\sum_{j=0}^{m(i)-1} \frac{\varphi(j)\tau(j)}{(1+\beta(j))s(j)}\right].$$

其中 $\varphi(j)$ 为进程 j 在 T_{lcm} 内执行次数;$\tau(j)$ 为 j 执行时间;$s(j)$ 为 j 内存数量;$\beta(j)$ 为 j 读/写次数比率.

加入节点 i 各进程 j 执行时间权重,计算 j 在 T_{lcm} 间单元数据访问次数 $N(i,j)$ 为

$$N(i,j) = \frac{\delta(j)\tau(j)}{\Delta(i)} \times \frac{m(i)\tau(j)}{\sum_{s=0}^{m(i)-1}\varphi(s)\tau(s)} = m(i)\frac{\phi(j)^2\tau(j)^2}{\sum_{t=0}^{m(i)-1}\left[\frac{\phi(t)\tau(t)}{[1+\beta(t)]s(t)}\right]\sum_{s=0}^{m(i)-1}\phi(s)\tau(s)}.$$

节点 i 的单元数据访问次数总开销 $C(i)$ 为各进程 j 对单元数据访问开销的平均值,即

$$C(i) = \frac{\sum_{j=0}^{m(i)-1}\frac{N(i,j)[1+\beta(j)]}{s(j)}}{m(i)} = \sum_{t=0}^{m(i)-1}\frac{[1+\beta(j)]\varphi(t)^2\tau(t)^2 s(t)}{\sum_{j=0}^{m(i)-1}\left[\frac{\varphi(j)\tau(j)}{[1+\beta(j)]s(j)}\right]\sum_{s=0}^{m(i)-1}\varphi(s)\tau(s)}.$$

求得 $C(i)$ 后,将 $C(i)$ 组成矩阵 $\boldsymbol{C}$,各节点 i 进程数组成 $\boldsymbol{K}$,各节点进程实际内存分配组成 $\boldsymbol{X}$,则

$$\text{系统总开销} = [\boldsymbol{KC}]^{\mathrm{T}}\boldsymbol{X}.$$

另外,系数矩阵 $\boldsymbol{A}$、$\boldsymbol{X}$、内存限制矩阵 $\boldsymbol{b}$ 组成限制条件 $\boldsymbol{AX} \leqslant \boldsymbol{b}$. 各矩阵表示如下.

$$\boldsymbol{K} = \begin{bmatrix} m(0) & & & \\ & m(1) & & \\ & & \cdots & \\ & & & m(R-1) \end{bmatrix},$$

$$\boldsymbol{X} = \begin{bmatrix} 1^{st}\text{ 节点内存 } S(0) \\ 2^{st}\text{ 节点内存 } S(1) \\ \cdots \\ R^{st}\text{ 节点内存 } S(R-1) \end{bmatrix},$$

$$\boldsymbol{A} = \begin{bmatrix} 1 & \cdots & 0 \\ -1 & \cdots & 0 \\ & \cdots & \\ 0 & \cdots & 1 \\ 0 & \cdots & -1 \end{bmatrix},$$

$$\boldsymbol{C} = \begin{bmatrix} C(0) \\ \cdots \\ C(R-1) \end{bmatrix},$$

$$\boldsymbol{b} = \begin{bmatrix} S(0)\ 上限 \\ S(0)\ 下限 \\ \cdots \\ S(R-1)\ 上限 \\ S(R-1)\ 下限 \end{bmatrix}.$$

可从统计角度将改进后的系统总访问开销的算式用于判断各节点上各任务的内存分配合理与否的量化评估. 可使用 MATLAB“LINPROG”等工具对其进行分析，以最小化总访问开销[10]. 关于计算模型的说明及非关键量的计算参见文献[10].

4　结　论

本文设计了一种基于反射内存网的支持异步、周期性数据通讯的实时传输协议 MSTP，解决分布式测试环境的复杂通讯问题，较已有设计通用性强. 并基于理论计算模型，给出了计算系统开销的更为完善的算法. 实验表明本 STP 能够满足分布式测试平台实时通信、时钟同步、任务同步等需要. 体现了较好的通用性、可配置性和集成性，对基于反射内存网的应用协议设计人员具有重要的参考价值. 下一步将基于 MSTP 设计分布式测试平台的相应算法开展工作.

参考文献：

[1] 刘斌，高小鹏，陆民燕，等. 嵌入式软件可靠性仿真测试系统研究[J]. 北京航空航天大学学报，2000，26(4)：490－493.

[2] ANDREW S T. 分布式系统原理与泛型[M]. 杨剑锋，译. 北京：清华大学出版社，2004：9.

[3] HAVLICSEK S H，ZANA L. Improving real-time communication between host and motion system in a HWIL simulation[C]//Procedings of SPIE. Pittsburgh PA：Acutronic USA，2004：33－43.

[4] TAKAHASHI H. Synchronized data distribution and acquisition system using reflective memory for J-PARC 3GeV RCS[C]//Processings of the 11th European Particle Accelerator Conference. Genoa：[s. n.]，2008：TUPPO13.

[5] 李海，吴嗣亮. 基于 HLA 和反射内存网的半实物卫星对抗仿真系统[J]. 系统仿真学报，2008，18(6)：1520－1523.

[6] 蒲源，王仕成. 多网络半实物实时仿真平台研究[J]. 系统仿真学报，2008，20(10)：2572－2574.

[7] JACUNSKI M G，MOORTHY V. Low latency message-passing for reflective memory networks[C]//In Proceedings of CANPC. Orlando：[s. n.]，1999：211－224.

[8] LAWTON J V，BROSNAN J J. Building a high performance message-passing system for memory chaninel clusters[J]. Digital Technical Journal，1996，8(2)：96－116.

[9] STOCKINGER H，STOCKINGER K. Towards a cost model for distributed and replicated data stores[C]//In Proceeding of Ninth Euromicro Workshop on Parallel and Distributed Processing. Italy：[s. n.]，2001：461－467.

[10] RYOU M S. Optimization of data accesses in reflective memory systems[C]//TENCON 2006，IEEE Region 10 Conference. HongKong：[s. n.]，2006：1－4.

[11] ABRAHAM S. 操作系统概念[M]. 第六版. 郑扣根，译. 北京：高等教育出版社，2005：4.

[12] SCRAMNet G T. A new technology for shared-memory communication in high-throughput networks [EB/OL]. [2009]. http://www.cwcembedded.com/documents/contentdocuments/White-Paper-SCRAMNet-GT.pdf.

（编辑　张　宏）