TIPC 透明进程间通信协议研究和应用①

冀映辉　蔡　炜　蔡惠智 (中国科学院 声学研究所 北京 100190)

摘　要： 针对雷达、声呐领域高性能信号处理平台的研发需求,提出并实现了在并行信号处理机系统中利用 TIPC 透明进程间通信协议实现通信接口的方法。分析了 TIPC 相比其它通信协议的优点，阐述了 TIPC 的基本实现原理，对 TIPC 的实现难点和重点进行了详细讨论，将 TIPC 和 TCP 传输协议的通信性能进行了实际测试比较。最后，提出了基于 TIPC 增强并行信号处理系统通信可靠性的方法。

关键词： TIPC；进程间通信；传输媒介；流量控制

Research and Analysis of Transparent Inter Process Communication Protocol

JI Ying-Hui, CAI Wei, CAI Hui-Zhi

(Institute of Acoustics, Chinese Academy of Sciences, Beijing 100190, China)

Abstract: To gain high signal processing capability in radar and sonar fields, this paper proposes and realizes communication interface among parallel signal processing machines with TIPC. It lists the main features of TIPC compared with other communication protocols. It also analyzes the basic principles of TIPC and its implementation in details. At the same time, it compares the communication performance between TIPC and TCP. At last, it puts forward a new method to enhance communication reliability among parallel signal processing machines based on TIPC.

Keywords: TIPC; inter process communication; transport media; flow control

近年来，在雷达和声呐系统中对信号处理的实时性要求越来越高，迫切需要性能更高的信号处理机做支持。随着计算机技术的高速发展，大规模并行信号处理机已经成为当前信号处理的首要解决方案。并行处理遇到的挑战主要有两点，一个是程序的并行度有限，另一个是处理器间通信的相对高额开销，其原因都可以用著名的 Amdahl 定律解释[1]。Amdahl 定律是:

$$\text{加速比}=\frac{1}{\frac{\text{改进部分所占比例}}{\text{改进部分的加速比}}+(1-\text{改进部分所占比例})}$$

目前各大 CPU 厂商推出的处理器间互连方式多种多样，有以太网、PCI、PCIe、RapidIO、Infiniband 等等。它们的通信方式各异、给用户的使用接口也千差万别，向上层用户提供一套统一、高效的通信接口在未来将是必然趋势。TIPC[2]正是基于这一需求而开发的，目前 TIPC 由风河公司维护。

1　现有通信协议的不足和TIPC的优点

1.1 现有通信协议的不足

现今，通信协议很多，但是还没有一款是为高性能信号处理系统量身定做的。以 TCP 通信协议为例，TCP 具有性能稳定、使用广泛的优点，但是在高性能信号处理系统中它存在以下显著缺点。

(1) TCP 通信节点地址不透明。在机器的 IP 地址频繁变化的情况下或者进程需要频繁迁移到其它机器的情况下，这给用户编程和使用带来很大不便。DNS 等机制虽然实现了 IP 地址的透明和动态查找 IP 的功能，但是在实时系统中使用它们得不偿失。

(2) TCP 缺少很多优化措施，特别是机器内部的通信。同一台机器的进程间通信，可以有内存映射、内存换页、内存共享等多种优化措施，使用这些可以

① 收稿时间:2009-06-29

极大的减少通信延迟和降低 CPU 资源占用。尽管用户可以编程区分这一情况，但是在通信协议内部实现自动区分将是更好的选择。

(3) TCP 面向定点的实现机制决定了它不可能实现真正的多播。

1.2 TIPC 优于一般通信协议的特点

(1) 利用功能性地址，在整个集群系统中实现了通信地址透明。

(2) 提供网络拓扑服务，并可对网络拓扑的改变及时做出响应。对于连接的错误或者目的端的不可达均可以在 1 秒之内做出响应。

(3) 提供通信冗余机制，当两个处理器节点之间有多个通信链路工作时，可选择某一链路进行通信并在该链路连接故障或者拥塞时自动通过别的链路发送消息。

(4) 用户可以设置发送消息的重要等级，当通信链路阻塞时，消息的重要等级决定 TIPC 内部对消息做何处理。

(5) 不但可以提供进程-进程间的通信，还可以实现进程-内核，内核-内核之间的透明通信。

(6) TIPC 只实现了 OSI 网络参考模型中数据链路层及其以上的部分，其对物理层的实现不做任何假设。理论上来讲，这就保证了 TIPC 物理层的实现可以是任何底层通信协议。

2 TIPC的软件架构

2.1 TIPC 中一些关键术语定义

(1) 通信端口(port)：用户实现通信的每一个端点，通常用户进程每创建一个 TIPC 类型的套接字就创建了一个通信端口。通信端口是 TIPC 中最小的通信单元。通信端口有三种表示方式，端口标识、端口名字和端口名字序列。

(2) 通信链路(link)：每对处理器节点之间实现的通信连接，完成消息的实际传输。每对处理器节点之间可以同时存在多个不同的通信链路。

(3) 传输媒介(bearer)：每一个物理上或者逻辑上传输介质的实现，如以太网，ATM，RapidIO 等。

(4) 网络地址(network address)：每一个处理器可以安排一个唯一的 TIPC 网络地址，类似于以太网络中的 IP 地址。

(5) 名字表(name table)：TIPC 内部保存的端口名字和端口标识的映射关系，不仅保存本机的，还保存与其建立连接的其它机器上的端口映射信息。

2.2 TIPC 内部结构

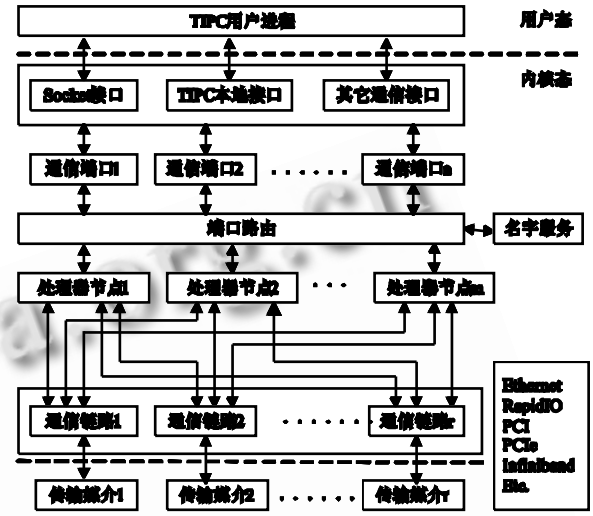

图 1 TIPC 内部结构视图

如图 1 所示，在最上层是 TIPC 提供给用户的接口，TIPC 默认实现了广泛应用的 socket 接口和 TIPC 内部自定义的一套通信接口，同时也允许用户定义自己的一套通信接口。中间部分实现了通信链路的建立，地址翻译，数据的传输控制，网络拓扑，多播等一系列服务，是 TIPC 的核心实现层。在最底层 TIPC 可以桥接实现多种通信协议，TIPC 的这一设计思想方便了很多互联协议驱动程序的开发， Linux 下的 TIPC 默认只实现了以太网传输媒介。

2.3 TIPC 通信链路的建立、维护和删除

通过底层传输媒介提供的多播或者广播接口，TIPC 能够自动发现其它处理器节点的存在并在接口配置允许的情况下自动在两个节点之间建立一个通信链路。一旦 TIPC 发现一个新的节点建立，它将周期性的从该节点向网络内的其它成员节点发送连接请求消息，告诉它们自己节点的存在。如果一个收到请求连接消息的节点决定和该请求连接节点建立一个通信链路，它将建立一个连接端点并向该请求连接节点发送一个连接回应消息。请求连接节点收到此回应消息将建立一个对应的连接端点，这样在这两个处理器节点之间就建立了一个活跃的通信链路。两个节点之间的通信链路状态转移关系如下图 2

所示。

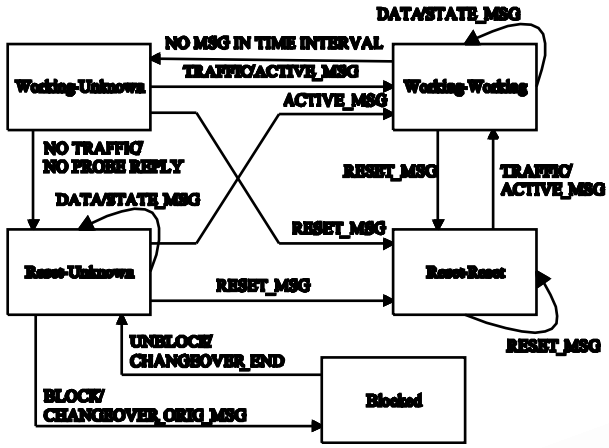

图 2　TIPC 通信链路状态转换关系图

TIPC 通信链路的状态由其两个端点的状态共同决定，通信链路的两个端点之间通过周期性的互通消息实现通信链路状态的转换，每一个新建立的通信链路都处于 Reset-Unknown 起始状态。

2.4 TIPC 阻塞机制的实现

2.4.1 通信链路阻塞

TIPC 将用户态消息按照消息的重要等级分成四种，DATA_LOW、DATA_NORMAL、DATA_HIGH 和 DATA_NONREJECTABLE。在发送端，TIPC 先将用户要发送的消息拷贝到内核态，内核将消息通过某一链路发送出去以后，将其按序加入该链路的发送队列之中。在接收端，当 TIPC 收到通过这一链路发送来的未回应消息个数超过一定量时，将向发送端发送一个回应消息。发送端收到这一回应消息，将该链路的发送队列中序号小于某一值的消息逐个弹出并将该消息所占内存释放。当链路的发送队列长度大于设定值时，TIPC 将该链路暂时标志为阻塞。此时，TIPC 将停止通过此链路发送消息，但是用户态进程是否还允许通过该链路发送消息依据消息的重要等级而定。DATA_LOW 消息将被拒绝发送，DATA_NORMAL 消息将被加入发送等待队列之中，DATA_HIGH 消息和 DATA_NONREJECTABLE 消息将被立即发送。此后一旦链路的发送队列长度小于设定值，链路将被重新标志为不阻塞从而继续正常发送和接收消息。

2.4.2 传输媒介阻塞

当本地传输媒介过载，例如，传输媒介的发送缓冲区满，这将导致传输媒介阻塞。传输媒介阻塞将导致所有建立在此传输媒介基础之上的相应链路阻塞。此时，TIPC 将停止通过此传输媒介发送任何数据包直到传输媒介阻塞解除。在传输媒介阻塞的过程中，用户态进程依然可以发送消息，数据包将被加入链路的发送等待队列之中，但是此时所有的实际发送工作都将停止直到该传输媒介再次标志为非阻塞状态。

3　TIPC性能评价

3.1 TIPC 和 TCP 性能测试比较

测试环境：处理器 MPC8548E[3]、1Ghz 主频、384MB 内存，操作系统 linux-2.6.13，千兆以太网，发送端和接收端通过千兆路由器直接相连，TIPC 底层桥接以太网传输媒介。

3.2 结果分析

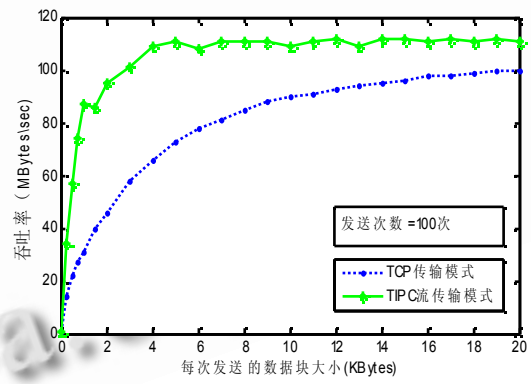

图 3　TIPC 流模式和 TCP 吞吐量曲线

如图 3 所示，就吞吐量而言，TIPC 比 TCP 最高提高了约 11%。随着每次传输数据块大小的增长，TIPC 的吞吐量增长很快，约在 4Kbytes 大小时就达到了峰值。相反，TCP 协议的吞吐量增长缓慢，直到 10Kbytes 大小时才达到其峰值的 90%，直到 19Kbytes 时才能达到吞吐量的峰值。图 4 对通信延迟的对比表明，当每次传输的数据块小于 10Kbytes 时，TIPC 和 TCP 的通信延迟相近，随着数据块大小的继续增大，TIPC 的通信延迟比 TCP 就小很多，特别是在数据块大小等于 20Kbytes 时，TIPC 的通信延迟仅仅是 TCP 通信延迟的约 63%。

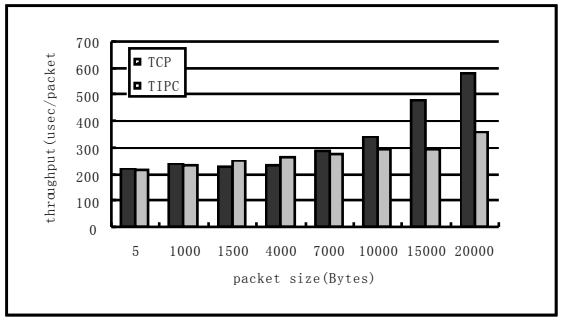

图 4 TIPC 流模式和 TCP 通信延迟比较

4 利用TIPC构建高可靠性通信网络

雷达和声呐领域的数字信号处理系统对系统运行的可靠性要求非常高，如何构建高可靠性的并行信号处理通信网络，一直是人们研究的一个重点。本文将 TIPC 提供的透明地址通信特性和 TIPC 同时支持多种传输媒介通信的特点结合起来，提出了利用 TIPC 构建高可靠性并行信号处理通信网络的方法[4,5]。作为系统的一个应用实例，我们实现的高可靠性并行信号处理通信网络如图 5 所示。

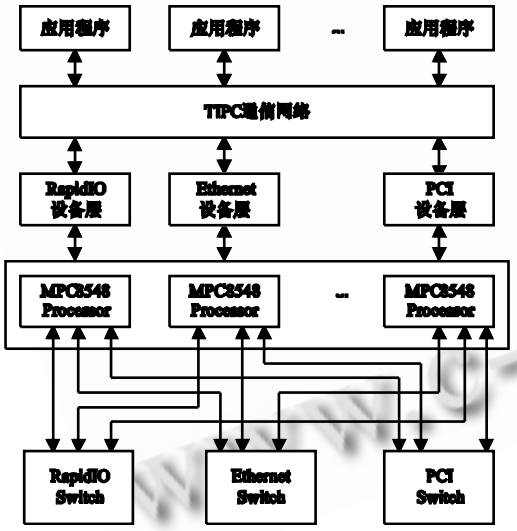

图 5 高可靠性并行信号处理系统通信网络架构

Freescale 公司的 MPC8548E 处理器内部同时集成了 RapidIO[6]互联协议、千兆以太网互联协议以及 PCI 互联协议，硬件通过它们各自的路由器与 MPC8548E 处理器互联组成三种独立的互联通信网络。设备层的主要功能是操作互联协议的硬件设备，完成最基本的数据传输任务。TIPC 通过与每种互联协议设备层的交互实现对每种传输媒介通信的支持。用户通过 TIPC 提供的配置工具 tipc-config[7]配置本节点支持的传输媒介类型并设置好每种传输媒介的通信优先级。应用程序通过TIPC提供的socket通信接口实现数据的发送和接收，TIPC 底层通过那种传输媒介传输的数据用户无需关心。当用户设置的最高优先级的传输媒介故障或者阻塞时，TIPC 将自动通过次优先级的传输媒介传输数据，而这一切对用户是透明的。通过 TIPC 提供的透明地址功能，当某一处理器节点故障时，只要将程序在别的节点重新运行即可，用户编写的应用程序不需做任何改动。这种设计有效的提高了 TIPC 向用户提供的通信带宽、在软件上使得系统发生通信故障的概率大大减小、极大的节省了硬件发生故障时的处理时间。

5 总结

本文指出了 TIPC 的优点和应用范围，对 TIPC 的内部实现进行了详细的分析。通过与 TCP 通信协议的比较说明，TIPC 无论是在上层用户接口方面还是在通信性能上都比 TCP 要优异。同时，TIPC 支持多个传输媒介同时工作的特点，有力的增强了我们并行信号处理系统通信的可靠性。

目前，TIPC 还存在以下问题。

(1) 归咎于 TCP 的 Nagle 算法，对于特别小的消息，TIPC 的吞吐量不如 TCP。

(2) 目前，TIPC 仅支持 Linux、VxWorks 和 Solaris 操作系统，对于使用其它操作系统的用户来说，就不能使用 TIPC。

(3) TIPC 使用传输媒介的 MTU 大小来拆包传输数据，从而导致其通信性能在一些固定点附近出现波动。

参考文献

1 Hennessy JL, Patterson DA. 郑纬民,汤志忠, 汪东升, 译.计算机体系结构-量化研究方法(第三版). 北京:电子工业出版社, 2004.359－361.

2 Maloy J. TIPC: Transparent Inter Process Communication Protocol(Version 2.0) 2006-05. http://tipc.sourceforge.net/doc/draft-spec-tipc-02.html

3 Freescale Semiconductor. MPC8548E PowerQUICC IIITM Integrated Host Processor Family Reference Manual (Version 1.0). [2005-07] http://www.free-

(下转第 11 页)

(上接第 79 页)

scale.com

4 Carew M, Merabti M, Whiteley K, et al. Multimedia support in distributed systems. Proc. of IEEE International Conference America, 1995.821 – 825.

5 Mirtaheri SL, Khaneghah EM, Sharifi M, et al. A Case for Kernel Level Implementation of Inter Process Communication Mechanisms. Proc. of the 3rd International Conference on Information & Communication Technologies: from Theory to Applications (ICTTA 2008). Syria, 2008.1 – 7.

6 RapidIO Trade Association. RapidIO Interconnect Specifications V1.3. 2005 http://www.rapidio.org

7 Maloy J. Linux TIPC 1.7 User's Guide. 2008-07 http://tipc.so- urceforge.net/doc/tipc_1.7_users_guide.txt