

基于消息的 Parlay X 网关过载控制算法

张载龙¹, 王亚宁¹, 李炳林², 刘世栋²

(1. 南京邮电大学 信息技术研究所, 江苏 南京 210003;

2. 中国电力科学研究院 信息通信研究所, 江苏 南京 211106)

摘要:为了解决目前 Parlay X 网关过载算法灵活性低、不能有效地使会话量最大化的问题,给出了基于合约的 Parlay X 网关系统架构。根据 Parlay X 消息类型特点设置消息优先级,根据消息优先级选择丢弃或者缓存消息,之后根据负载情况进行负载均衡以及接入控制,通过四个阶段来实现过载控制过程。实验结果表明,相比较于传统过载控制,该机制能在保证新建会话消息接收数量的同时提高其他类型消息的接收数量,减少平均服务时间,在满足合约限制的情况下,保证最大会话数量。

关键词:Parlay X; 过载控制; 接入控制; 负载均衡

中图分类号: TP301.6

文献标识码: A

文章编号: 1673-629X(2014)10-0119-05

doi: 10.3969/j.issn.1673-629X.2014.10.028

Overload Control Algorithm of Parlay X Gateway Based on Message

ZHANG Zai-long¹, WANG Ya-ning¹, LI Bing-lin², LIU Shi-dong²

(1. Institute of Information and Network Technology, Nanjing University of Posts and Telecommunications, Nanjing 210003, China;

2. Research Institute of Information Communication, China Electric Power Research Institute, Nanjing 211106, China)

Abstract: In order to solve the problems that the Parlay X gateway overload algorithm has low flexibility and cannot make the messages maximized, propose a Parlay X gateway system architecture based on contract. Set the priority of the message according to the feature of the Parlay X message, drop or cache the messages according to the message priority, after that conduct the load balance and admission control in accordance with load, the overload control process has been realized by four steps. The experimental results show that compared with traditional overload control, the proposed algorithm can reduce the average service time, increase the received number of calls, while guaranteeing the number of MakeCall message, make sure the maximal messages number when meeting the contract constrains.

Key words: Parlay X; overload control; admission control; load balance

0 引言

Parlay X^[1]对电信能力进行更进一步的抽象,在NGN(下一代网络)^[2]构架中,通过应用服务器(Application Server, AS)使用一个请求消息就能获取电信网络资源。但是当有太多消息发送到一个Parlay X网关时,网关就可能过载,导致服务时间过长,用户可能会放弃当前的请求而重新发起一个请求。但是已经放弃的请求同样会消耗珍贵的处理时间,在最坏的情况下,一个过载的网关可能只处理已经被丢弃的服务请求。这些都可以引入过载控制机制来解决,在过

载情况下,尽可能早地拒绝一些请求,那么使得已经接收的请求消息可以得到一个好的服务。此外还需要一个机制能够检测Parlay X网关什么时候过载。

1 国内外研究现状

近年来,电信网络中通信过载控制已经成为一个热点话题。在早前的一篇文章中,Forys^[3]讨论了电话交换中控制处理器的过载保护。对于IN(智能网)的过载控制,Pham等^[4]比较了窗口算法和闭环算法,从过载响应速度和公平性而言,窗口算法更优。Kihl

收稿日期: 2013-11-16

修回日期: 2014-02-20

网络出版时间: 2014-07-17

基金项目: 南京邮电大学攀登计划(NY210012); 中国电力科学研究院南京分院(2013外21); 江苏省科技科技成果转化项目(BA2012024)

作者简介: 张载龙(1966-),男,江苏淮阴人,研究员,研究方向为基于IP的下一代电信网络关键技术; 王亚宁(1990-),男,安徽蚌埠人,硕士研究生,研究方向为基于IP下一代网络通信。

网络出版地址: <http://www.cnki.net/kcms/detail/61.1450.TP.20140717.1226.003.html>

等^[5]研究了带票缓存的分散和集中两种机制的算法,在健壮性和公平性方面,集中控制机制更好。Melen等^[6]对 OSA 网关和 Parlay 网关,作出了一般性能的分析。Dahlin^[7]讨论了已经丢弃的消息对负载状态的影响。Berger^[8]则给出了不同接入控制的方法。文献^[9]给出 Parlay X 网关的负载控制框架以及过载控制和负载均衡算法的实现,对消息设置高低优先级,但是没有具体消息类型和会话的特点,只是在过载时控制接收消息速率。Lee 等^[10]在 SCP 中提出高优先级和低优先级队列,其中高优先级中的任务比低优先级的先处理,给出了两种改进的 ACG 算法。

此前的文章大多是在公平看待用户的基础上,如何拒绝一个呼叫。从用户的角度来看这是最公平的方法,但对以盈利为目的的运营商而言显然不是一个好的办法。运营商需要的是根据应用产生的利润与优先级相对应的过载控制方案,此外还必须根据其他的一些变量来保证用户的服务质量。文中给出 Parlay X 网关系统结构,并结合了 Parlay X 消息的特征,提出一种基于消息特点的过载控制算法。

2 Parlay X 过载控制方案

Parlay X 架构是合约驱动的架构^[11],在 AS 和 Parlay X 网关之间有着若干合约和限制,文中只关注关于服务器性能的变量。其中的一个合约是每个时间单元内应保证至少接收多少个来自同一个 AS 的请求消息。

2.1 系统架构

此方案系统架构如图 1 所示。

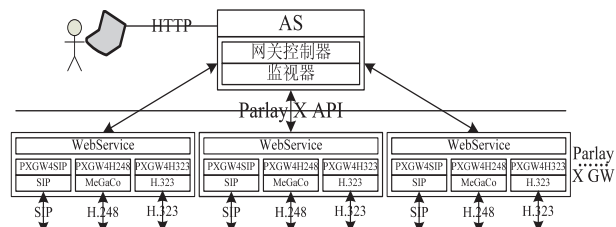


图 1 Parlay X 系统架构

(1) Parlay X 网关: 由网络服务提供商提供,屏蔽底层网络细节,抽象底层网络能力,以 Web Service 服务方式提供业务能力接口,使得服务提供商更容易使用底层网络。

(2) 应用服务器 (AS): 当应用开发者使用网络资源时,通过 HTTP 发送一个 SOAP (Simple Object Access Protocol, 简单对象访问协议) 消息给 AS 就能获取网络资源。

其中过载控制器是 AS 中的一个模块,包括一个网关控制器和一个负载监视器。每次都启动一个新的线程把消息从 AS 发送到 Parlay X 网关中,然后用此线

程接收响应消息以便于监视器从这个线程获取到当前负载的估计值。网关控制器根据从监视器中获取的信息,决定是否接受或者丢弃某一类型的消息。一个过载控制器可以对若干个 Parlay X 网关进行过载控制,其通信过程如图 2 所示。

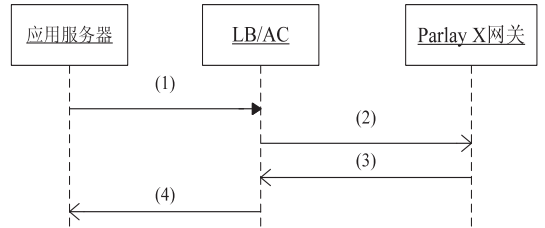


图 2 系统通信过程

消息 (1) 和 (4) 是 AS 和过载控制器进行通信的内部消息。当应用服务器收到一个 SOAP 消息进行相应处理后,会把消息 (1) 发送到过载控制器模块,过载控制模块进行接入控制和负载均衡处理之后,会把此 SOAP 消息 (2) 转发到相应的 Parlay X 网关。在网关处理完请求之后,会发响应消息 (3) 发送给过载控制模块,过载控制模块会根据响应消息收集对应网关的负载信息,之后会发送消息 (4) 给应用服务器。

为了区分不同的消息,每一条消息都会有一个会话 ID,封装在 SOAP 中,过载控制器和 Web Service 控制器都有 SOAP 消息解封装的功能,以获取会话 ID。任何一个网络资源都以 SOAP 消息来指定应用想要请求的网络资源,SOAP 消息经过过载控制器被发送到相应的 Parlay X 网关中,同时过载控制器会对 SOAP 消息进行解析,记录下会话 ID。Web Service 控制器接收 SOAP 消息后,把 SOAP 消息请求转化为对相应的 SCF (Service Control Function, 业务控制功能) 调用。

2.2 Web Service 控制器

Web Service 控制器向 Parlay X 应用提供统一调用接口,处理标准的 Web 请求,包括接收和发送 Web 消息,将收到的请求缓存下来,分配给相应的 SCF。在 Parlay X 中,Web Service 控制器主要支持四种类型的消息,并且每种消息都有相对应的优先级。

EndCall: 优先级为 1, 结束当前正在进行的一个会话。

GetCallInfo: 优先级为 2, 获取指定会话的相关信息。

SetCallInfo: 优先级为 2, 设定指定会话的相关信息。

MakeCall: 优先级为 3, 请求创建一个新的会话。

其中 MakeCall 消息的优先级最低,因为这里只关注 MakeCall 消息的接收速率,同时也是为了保证已经建立会话的消息的通过率和通信质量。由于应用呼叫都是基于会话的,所有有着同样会话 ID 的消息都属于

同一个会话,同一个会话的 EndCall、GetCallInfo、Set-CallInfo 消息都必须发送到同一个 Parlay X 网关,否则就必然造成不必要的开销,在 MakeCall 请求消息到达 Web Service 控制器时,Web Service 控制器就保存下会话 ID。

2.3 过载控制

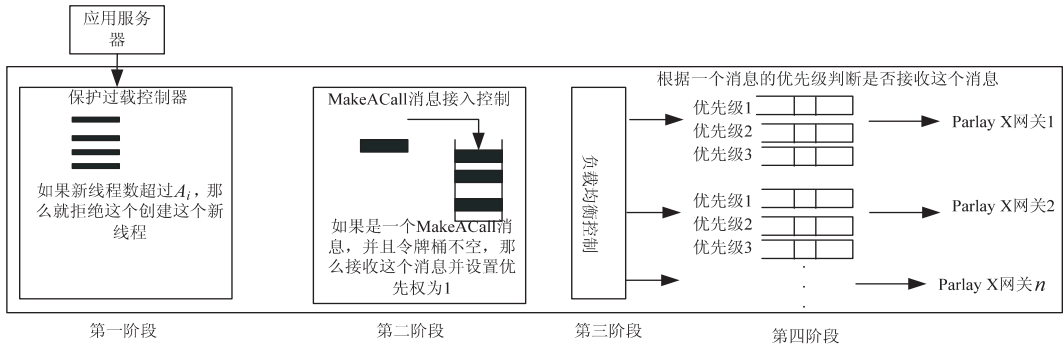


图3 过载控制不同阶段

2.3.1 粗略接入控制

第一阶段使用粗略接入控制保护过载控制器节点免受过载情况,如果消息到达速率太高,就直接丢弃这些消息。第 i 个过载控制器只允许有 A_i 个线程同时为 AS_i 服务,当 AS_i 想创新额外的线程进行服务,就拒绝这个请求。假定 AS_i 每秒发送的消息数量不大于 r_i ,当超过 r_i 时,就直接丢弃超出的消息。一个消息的时间限制是 τ ,也是一个线程在过载控制器和 AS_i 之间进行一次通信的最大时间。那么 A_i 可由式(1)得到

$$A_i = \tau \cdot r_i \quad (1)$$

2.3.2 MakeCall 消息控制

当过载控制器已经实现了消息分类功能时,要保证第 k 个应用服务器 AS_k 每个时间单元内的新建会话数量最多为(即 MakeCall 消息数量) d_k , d_k 的值由合约进行设置。这里使用大小为 d_k ,令牌速率为 d_k 的令牌桶。如果令牌桶中有令牌,则给到达的 MakeCall 消息分配的优先级为 1, EndCall 消息的优先级也是 1, 这些消息都能在负载均衡阶段被接收。每个过载控制器控制 MakeCall 消息的速率都只用一个令牌桶。如果没有令牌,当一个 MakeCall 消息到达时,只是转发到下一个阶段,并且分配的优先级是 3。GetCallInfo 和 End-Call 消息在转发的时候所分配的优先级一直是 2 和 1。

2.3.3 负载均衡

由于应用是基于会话的,其他消息都会发送到指定的 Parlay X 网关,只有 MakeCall 消息是要进行调度的。这里用动态加权轮转调度算法 (Weight Round Robin)^[12],可以根据网关实时的负载情况,分配消息,使网关负载更加均衡。

过载控制器中设置了带有优先级的消息队列,分别与 Parlay X 网关相对应,一个消息经过负载均衡器

一个应用的时间限制是 τ ,消息(1)和(4)之间的等待时间为 $\Delta t_{\text{measured}}$,过载定义为用户等待时间过长,也就是 $\Delta t_{\text{measured}} > \tau$ 。同时假定第 k 个应用服务器 AS_k 每个单元时间内的新会话数量(即 MakeCall 消息数量)最多为 d_k 。其中过载控制过程可分为四个阶段如图 3 所示。

后,就发送到指定优先级队列的末端。过载控制器也会记录下每个优先级为 1 的消息数量 N_i 。设每个网关能处理的消息数量为 Num^k ,那么给每个队列设置的权值,就是 $W_1 = \frac{\text{Num}^1}{N_1}$, $W_2 = \frac{\text{Num}^2}{N_2}$, $W_3 = \frac{\text{Num}^3}{N_3}$ 。假设有队列 $Q_1 \sim Q_4$,且 $W_1 : W_2 : W_3 : W_4 = 6 : 3 : 2 : 3$ 。根据每个队列的权值分配消息,分配过程就是 $Q_1 Q_2 Q_3 Q_4 Q_1 Q_2 Q_3 Q_4 Q_1 Q_2 Q_4 Q_1 Q_1 Q_1$ 。给 Q_1 分配了 6 个消息,3 个给 Q_2 ,2 个给 Q_3 ,3 个给 Q_4 。在完成上述过程即一次轮转之后,负载监视器在一个次轮转之前会根据当前的消息数量重新设置队列权值,从而达到动态分配的目的。

GetCallInfo、SetCallInfo 和 EndCall 消息只能发送到指定的 Parlay X 网关进行相关的服务。因此,最好不要把一个消息发送到错误的 Parlay X 网关中,这将导致额外的处理开销,浪费处理器的能力。在解封装的情况下,在 Web Service 控制器有一个表,表中内容是要服务的会话 ID。

2.4 接入控制

接入控制是用来选择哪个消息发送到 Parlay X 网关,哪个消息是被丢弃的^[5]。过载定义为消息不能在指定的时间限制内得到服务,如果一个消息在时间限制内不能得到服务,那么这个消息就是超时消息。在过载控制器中,一个消息在过载控制器中从发出请求到得到响应之间的时间间隔为 $\Delta t_{\text{measured}}$ 。一个消息在 AS 中从发出到响应的时间间隔为 Δt_{user} ,当 Δt_{user} 大于 τ 时,这个消息就是超时消息,而控制器的目标就是

$$\Delta t_{\text{user}} < \tau \quad (2)$$

这个算法是对 Parlay X 网关的瞬时的负载敏感,在 Parlay X 网关过载时,为了避免继续缓存消息,过载

时只允许有一个线程用于过载控制器和 Parlay X 网关进行通信。当一个消息经过负载均衡阶段,会被发送到相应优先级队列中。这里为每一个 Parlay X 网关设置了三个优先级队列,优先级分别是 1、2、3。过载控制器会记录下每个优先级队列的消息数量,优先级为 j 的队列的消息数量为 $N(j)$ 。假设负载情况不会改变的过快,在每个消息进入到优先级队列时,都会为这个消息分配一个时间戳,那么计算出这个消息在队列当中停留的时间 Δt_{queue} ,用以计算消息发送到 Parlay X 网关时候的负载情况。

但是,要满足公式(2),就有可能导致不能处理队列中的一些消息。 Δt_{user} 包括 $\Delta t_{\text{measured}}$ 和 Δt_{queue} ,所以消息就不能在队列中等待过长时间。假设 $\Delta t_{\text{measured}}$ 是处理一个消息所需时间,当不满足公式(3)时,除了优先级为 1 的消息,其他不满足条件的消息都被丢弃。

$$\Delta t_{\text{queue}} + \Delta t_{\text{measured}} < \tau \cdot f \quad (3)$$

其中, f 是小于 1 的系数。如果在 Parlay X 网关中, $\Delta t_{\text{measured}}$ 保持不变,那么使用公式(4)来估计有哪些优先级队列的消息可以发送给 Parlay X 网关。

$$\sum_{j=1}^m N(j) \times \Delta t_{\text{measured}} < \tau \times f \quad (4)$$

当满足式(4)时,就能得到最大优先级 j 。那么就从优先级值小于等于 j 的队列中选择排队时间最长的消息发送到 Parlay X 网关。

3 仿真结果和分析

为了降低仿真难度,采用 4 个应用服务器,4 个过

表 1 不同情况下的结果

100 秒仿真	MMPP 机制	稳态泊松 10/s	稳态泊松 25/s	稳态泊松 37/s	稳态泊松 55/s	稳态泊松 70/s
MakeCall 消息全部接收数量	3 200	1 100	2 500	3 200	3 500	3 700
优先级 2 消息丢弃率/%	5	0.1	0.6	0.8	4	10
优先级 3 消息丢弃率/%	16	0	0	5	14	33
GetCallInfo 消息全部接收数量	7 700	3 200	6 800	8 300	9 200	7 100
SetCallInfo 消息全部接收数量	7 800	3 200	6 700	8 400	9 100	7 000
EndCallInfo 消息全部接收数量	3 300	1 100	2 300	3 200	3 400	3 400
平均服务时间/ms	64	10	18	18	68	131
超时消息率/%	5	0.1	1	1	7	15

对于传统过载控制机制^[14],在一个消息到达时,立即判断是否丢弃或者接收这个消息,虽然考虑到消息的优先级,但是并不能保证接收 MakeCall 消息的数量,对于是否丢弃一个消息,是根据此前的时间段中已经发送消息数量来决定的。当传统的过载机制使用 MMPP 模型来进行仿真时,得到的结果是:6% 的超时消息,平均服务时间 72 ms,3 050 个 MakeCall,6 900 个 GetCallInfo,6 700 个 SetCallInfo 和 2 950 个 EndCall 消息被接收。相比较于表 1,由于能够缓存消息,这些数

据表明了文中提出的过载控制机制的优势。

据表明了文中提出的过载控制机制的优势。

据表明了文中提出的过载控制机制的优势。

据表明了文中提出的过载控制机制的优势。

据表明了文中提出的过载控制机制的优势。

表 1 不同情况下的结果

100 秒仿真	MMPP 机制	稳态泊松 10/s	稳态泊松 25/s	稳态泊松 37/s	稳态泊松 55/s	稳态泊松 70/s
MakeCall 消息全部接收数量	3 200	1 100	2 500	3 200	3 500	3 700
优先级 2 消息丢弃率/%	5	0.1	0.6	0.8	4	10
优先级 3 消息丢弃率/%	16	0	0	5	14	33
GetCallInfo 消息全部接收数量	7 700	3 200	6 800	8 300	9 200	7 100
SetCallInfo 消息全部接收数量	7 800	3 200	6 700	8 400	9 100	7 000
EndCallInfo 消息全部接收数量	3 300	1 100	2 300	3 200	3 400	3 400
平均服务时间/ms	64	10	18	18	68	131
超时消息率/%	5	0.1	1	1	7	15

据表明了文中提出的过载控制机制的优势。

4 结束语

文中介绍了 Parlay X 系统环境和架构,提出并且评估了一种 Parlay X 的过载控制算法。该算法可以在保障应用呼叫的情况下,最大化通信量,并且对于不同优先级的消息,能有效控制请求和响应之间的最大时延。与传统过载控制机制相比较时,超时消息率和平均服务时间相差无几,但是由于能够缓存消息,在 100

s 内接收消息总数明显超过传统过载机制。

在未来,Web Service 控制器将会是一个分布式架构,充当 Parlay X 和 Parlay 适配器的角色。而在分布式情况下,服务器的负载瞬间变化可能会很快,此前所测负载值和现在负载情况相差甚大。那么未来将进一步研究分布式 Parlay X 过载,避免出现负载变化过快导致资源浪费和服务器过载的情况。

参考文献:

- [1] 基于互联网服务(Web Service)的开放业务接入应用程序接口(Parlay X)技术要求[S]. YDT_1661-2007,2007.
- [2] 曾 蒸. 下一代网络 NGN 技术探析[J]. 重庆工学院学报, 2007,21(17):149-153.
- [3] Forsy L J. Performance analysis of a new overload strategy [C]//Proceedings of the tenth international teletraffic congress. Montreal; [s. n.], 1983.
- [4] Pham X H, Betts R. Congestion control for intelligent networks [C]//Proc of international Zurich seminar on digital communications. Zurich; IEEE, 1992.
- [5] Kihl M, Nyberg C. Investigation of overload control algorithms for SCPs in the intelligent network [J]. IEE Proceedings of Communications, 1997, 144(6):419-423.
- [6] Melen R, Moiso C, Tognon S. Performance evaluation of a Par-

lay gateway [C]//Proc of 7th international conference on intelligence in next generation networks. [s. l.]: [s. n.], 2001:181-184.

- [7] Dahlin M. Interpreting stale load information[J]. IEEE Transactions on Parallel And Distributed Systems, 2000, 11(10): 1033-1047.
- [8] Berger A W. Overload control using rate control throttle; selecting token bank capacity for robustness to arrival rates[J]. IEEE Transactions on Automatic Control, 1991, 36(2):216-219.
- [9] 徐莉娜. 基于 NGN 的第三方多业务融合的研究[D]. 上海:上海大学,2007.
- [10] Lee Y, Song J S. Overload control of SCP in advanced intelligent network with fairness and priority[J]. Computer Communication, 1999, 22(2):137-143.
- [11] 郑劲松. Parlay X 技术研究[J]. 现代情报, 2006, 26(2): 193-195.
- [12] 陈华平, 计永昶, 陈国良. 分布式动态负载平衡调度的一个通用模型[J]. 软件学报, 1998, 9(1):25-29.
- [13] 李玉福, 李胜利. 一种基于 CORBA 的可适应的负载平衡模型[J]. 华中科技大学学报, 2001, 29(12):84-86.
- [14] 李永平, 邹 华. 应用服务器的负载平衡技术和实现方案[J]. 电信科学, 2003, 19(10):15-18.

(上接第 118 页)

通过仿真验证表明,仿真结果与设计预期所要求的结果一致,满足设计要求。

4 结束语

文中为了解决在 VESA 视频数据通过 RapidIO 接口向主机或 DSP 传输中,传统方法的时序转换电路直接进行时序转换所带来的设计复杂、可移植性差等问题,提出用 DPRAM 将 VESA 接口与 RapidIO 的 DMA 接口隔离的体系架构,在此基础上,详细论述 VESA 转 RapidIO 接口的设计与实现。并且通过仿真验证工具以及板级验证对该设计实现进行了功能仿真及测试。

功能仿真和工程实践表明,该转接桥工作稳定,性能良好,较好地满足了 VESA 接口与 RapidIO 接口之间的通信需求,并且对于其他类似的接口转换设计具有一定的借鉴意义。

参考文献:

- [1] VESA and industry standards and guidelines for computer Display Monitor Timing (DMT) [S]. 2004.
- [2] 陈跃军,姜文博. 基于 DOS 操作系统下的高分辨率图形显示[J]. 黑龙江科技信息, 2011(21):101-101.
- [3] 赵艳军,何其锐,施锡涛. 基于 DVI 和 FPGA 的视频叠加器

设计[J]. 电子技术应用, 2011, 37(6):31-34.

- [4] RapidIO Trade Association. RapidIO interconnect specification (Rev. 2.1) [S]. [s. l.]:RapidIO Trade Association, 2009.
- [5] RapidIO Trade Association. RapidIO interconnect specification (Rev. 1.3) [S]. [s. l.]:RapidIO Trade Association, 2005.
- [6] 马春江,牛文生,孙靖国. 几种串行总线互连技术分析[J]. 航空计算技术, 2007, 37(5):127-130.
- [7] 聂新义,孙柯柯,马克杰. PCI 转 RapidIO 桥的设计与实现[J]. 计算机工程, 2010, 36(3):246-248.
- [8] 林国欢,胡封林,吴虎成. RapidIO 到 AXI 接口转接桥的设计与实现[C]//第十七届计算机工程与工艺年会论文集. 西安:国防科技大学出版社, 2013.
- [9] Hennessy J L, David A. Patterson computer architecture: a quantitative approach[M]. [s. l.]:Elsevier Science, 2003.
- [10] 陈世健,张 铁. 基于 DPRAM 的数据实时采集系统的研究[J]. 微计算机信息, 2009(10):84-85.
- [11] ACEX 1K data sheet [EB/OL]. 2003. <http://www.altera.com>.
- [12] 方湘艳,袁由光,韩 威. 基于 FPGA 的 PCI 到 ISA 总线转换芯片的设计与实现[J]. 计算机工程和应用, 2006, 42(33):83-87.
- [13] 杨海波,田 泽,蔡叶芳,等. 基于 FPGA 的多功能 FC 协议分析仪设计[J]. 计算机技术与发展, 2013, 23(7):214-217.

基于消息的Parlay X网关过载控制算法

作者: [张载龙](#), [王亚宁](#), [李炳林](#), [刘世栋](#), [ZHANG Zai-long](#), [WANG Ya-ning](#), [LI Bing-lin](#),
[LIU Shi-dong](#)

作者单位: [张载龙, 王亚宁, ZHANG Zai-long, WANG Ya-ning \(南京邮电大学 信息网络技术研究所, 江苏南京, 210003\)](#), [李炳林, 刘世栋, LI Bing-lin, LIU Shi-dong \(中国电力科学研究院 信息通信研究所, 江苏 南京, 211106\)](#)

刊名: [计算机技术与发展](#) 

英文刊名: [Computer Technology and Development](#)

年, 卷(期): 2014(10)

引用本文格式: [张载龙](#). [王亚宁](#). [李炳林](#). [刘世栋](#). [ZHANG Zai-long](#). [WANG Ya-ning](#). [LI Bing-lin](#). [LIU Shi-dong](#) [基于消息的Parlay X网关过载控制算法](#)[期刊论文]-[计算机技术与发展](#) 2014(10)