

# 动态计算网络:一种面向物联网的普适计算框架

卢 军 黄 健 许源平 郭本俊 岳 希

(成都信息工程学院软件工程学院 成都 610225)

**摘 要** 随着物联网的发展,大量不同的设备通过各种不同的方式连接到物联网中,使普适计算环境更加复杂和多样化,因此要求在物联网环境下的普适计算框架能适应新的计算条件的变化。提出了一种新的普适计算框架,以便在物联网环境中普适计算系统能够自动适应复杂多样的软硬件运行环境,使软件无需修改就可以在整个物联网环境中任意地执行,实现普适计算的服务发现、上下文感知与服务迁移功能。提出物联网环境下普适计算的基本运算单元为“动态计算网络”,使用“设备动态匹配”作为普适计算软件自动适应环境的解决方案。提出的动态计算网络增强了普适计算软件对物联网环境的适应性,为面向物联网的通用软件系统设计提供了一种有效的解决方法。

**关键词** 框架,普适计算,物联网

中图分类号 TP391 文献标识码 A

## Dynamic Computing Network: Architecture of Pervasive Computing for the Internet of Things

LU Jun HUANG Jian XU Yuan-ping GUO Ben-jun YUE Xi

(Software Engineering College, Chengdu University of Information Technology, Chengdu 610225, China)

**Abstract** With the development of Internet of Things, a lot of different devices connect to the Internet of Things through a variety of ways. This trend makes the pervasive computing environment more complex and requires the architecture of pervasive computing change to adapt the new environment of the Internet of Things. This paper provided architecture of pervasive computing for Internet of Things, which aims to how to make the pervasive computing systems to automatically adapt to the complex hardware and software operating environment in the internet of things. This architecture of pervasive computing can make the software run everywhere without modification, and implement the service discovery, context-aware service and service migration easily. This paper used the computing area network as the base infrastructure, and used the "devices dynamically match" as the solution of software self-adaptive. The research of this paper enhances the adaptability of pervasive computing software on the environment of the Internet of Things, and provides an effective solution of pervasive computing software system design for the Internet of Things.

**Keywords** Architecture, Pervasive computing, Internet of Things

## 1 引言

物联网的出现,使普适计算的计算基础环境越来越成熟,普适计算已经开始走向实际的应用<sup>[1,14]</sup>,这是因为:(1)由于物体普遍联网的需求以及物联网技术的发展,推进了计算设备的小型化、微型化与泛在化。从而,促使普适计算的物理环境越发成熟。(2)传感器技术的快速发展,使传感器能更好地感知人以及与人相关的各种设备的状态,从而为普适计算中的上下文感知奠定了物理实现基础。(3)物联网技术的发展,需要数据的传输、存储、处理等方面技术的快速发展,这推动了云计算技术、移动通信技术的快速发展,为普适计算的服务迁移提供了有效的物理实现基础。(4)物联网技术的发展,促使了更多的设备IP化,并连接到IP网络中,也促使了很多的小型非IP传感器通过设备代理的方式连接到物联网中。物

联网中的各种传感器、IP设备与传统互联网中的计算机系统相互混合,形成了物联网环境下普适计算的物理计算环境。

总之,物联网的出现和快速发展,深刻地促进了普适计算物理计算环境的发展。物联网,是一种传统计算机、新型计算设备和新型信息设备连接在一起的新的网络。

在物联网这种新型的网络环境下,为普适计算创建怎样的体系框架,才能更好地适应物联网的发展,才能很好地解决在物联网环境中的普适计算需求?这是一个需要重点研究的问题。

本课题研究的核心内容就是在物联网环境下,面向普适计算应用,研究普适计算的体系框架,使其能有效地支撑普适计算的服务发现、服务迁移与上下文感知。

大多数学者认为普适计算是极其复杂的,是由多个计算设备节点之间的协作完成的<sup>[2,3]</sup>。为了解决这一问题,普适

到稿日期:2011-09-01 返修日期:2011-12-20 本文受国家自然科学基金(61102076)资助。

卢 军(1974—),男,副教授,主要研究方向为普适计算、网络协议、嵌入式等,E-mail:jluc@cuit.edu.cn;黄 健(1976—),男,讲师,主要研究方向为移动计算;许源平(1981—),男,副教授,主要研究方向为嵌入式与移动计算、图像处理等;郭本俊(1975—),男,讲师,主要研究方向为移动计算;岳 希(1964—),女,副教授,主要研究方向为嵌入式与移动计算。

计算中间件是一个重要的研究方向。

文献[4]提出一种由服务发现协议 PDP 和服务描述语言 GSDL 组成的服务发现中间件。PDP 是一个安全分布式协议,综合了拉和推的特征;GSDL 是一个基于 XML 的语言,使用层次服务描述并使用 WSDL(Web Service Description Language)描述接口;SAFE-RD<sup>[5]</sup> 基于簇结构(tree structure of the cluster)对服务进行描述;UbiCOSM<sup>[6]</sup> 采用了抽象的原数据(metadata)描述机制。

2002 年,美国 Carnegie Mellon University 计算机科学学院的学者 Shang-Wen Cheng 和 David Garlan 在文献[7]中提出了一种软件体系模型,该体系模型可以监控一个软件系统的执行,并能动态地改变其执行状态。该体系模型将软件系统划分为 3 层:Runtime Layer 负责测量系统的运行属性和执行底层的操作;Model Layer 负责对系统的行为从高层进行分析,并最终决定是否启动系统修复;Task Layer 负责服务的质量是否满足应用的要求。这种软件体系模型的核心思想是使用体系模型化方法来使整个软件系统可自适应,侧重于自身资源的动态调配,使用模型化的方法来调配。Shang-Wen Cheng 和 David Garlan 提出的体系,将普适计算的核心分解为:输入设备、输出设备与控制设备。不过,在 Shang-Wen Cheng 和 David Garlan 的研究中,这些设备还是一台计算机上的不同软件层,不具备本项目中提出的动态计算网络的概念。

本文针对物联网环境下的普适计算软件框架进行研究,针对物联网真实的网络、设备环境,尽量避免做出过多的简化假设。因此,在本项目的研究中,网络环境是复杂的,计算设备是多样的。本文提出:物联网环境下普适计算的基本运算单元为“动态计算网络”,使用“设备动态匹配”作为普适计算软件自动适应环境的解决方案,增强了普适计算软件对物联网环境的适应性,为面向物联网的通用软件系统设计提供了一种有效的解决方法。

本文第 2 节介绍动态计算网络的基本构成与工作原理;第 3 节介绍基于动态计算网络的普适计算软件工作过程;第 4 节介绍基于动态计算网络的普适计算功能实现;第 5 节介绍了本文所提方案的验证实验;最后是总结和对未来工作的展望。

## 2 动态计算网络

在物联网中,网络环境是复杂的,传感器、I/O 设备是多样的。本文针对物联网应用环境,提出了一种新型的物联网普适计算框架:动态计算网络。

动态计算网络是由一系列联网的设备元件组成的一个网络,这些设备元件包括:控制元件、运算元件、存储元件、输入元件、输出元件。每个元件都是 IP 化的智能设备,通过网络互联,共同形成动态计算网络。

动态计算网络的系统示意图如图 1 所示。

构成动态计算网络的主要元件的功能如下:(1)控制元件:完成动态计算网络的运行控制功能,包括元件管理与通

信;(2)运算元件:负责执行软件代码,可以由多种语言的执行环境组成(例如 Java、C/C++ 等),以保证动态计算网络与具体的语言无关;(3)存储元件:存储软件代码与数据,以及存储服务的状态;(4)输入、输出设备:执行普适计算软件的输入、输出功能。

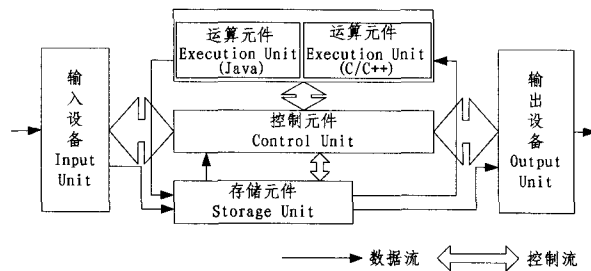


图 1 动态计算网络的系统示意图

普适计算软件在动态计算网络中的执行过程如下:

(1)代码与数据通过网络传输到一个动态计算网络中,并存储在存储元件中。

(2)控制元件分析软件,确定其语言类型,依据其语言类型确定该软件对应语言类型的运算元件。

(3)在控制单元的控制下,存储单元中的代码被读入到运算元件中执行,数据在存储元件与运算元件之间传输。

(4)运算元件在执行代码的过程中,如果软件代码执行了 I/O 操作,将触发设备动态匹配过程。设备动态匹配过程将依据软件自身的 I/O 设备描述表中描述的 I/O 设备请求类型,与动态计算网络的 I/O 设备描述表中描述的设备列表,通过设备能力匹配来确定软件 I/O 操作的输入/输出设备。

如果软件 I/O 设备能力匹配失败(没有找到能力一致的设备),则由控制单元启动降级执行过程,寻找不匹配、但最适合上下文环境的 I/O 设备,从而完成软件的 I/O 操作执行。

如果降级执行失败,那么软件执行失败退出。

(5)软件执行结束后,控制单元完成程序执行的评估,并向用户反馈评估信息。

在物联网中,每一个动态计算网都有一个唯一的域名。例如,域名 se.cuit.cn 表示中国成都信息工程学院软件工程学院所属的动态计算网。动态计算网络中每个元件都有一个独立的域名,例如 display.se.cuit.cn,表明该元件是动态计算网络:中国→成都信息工程学院→软件工程学院中的 display 显示设备。所属在一个动态计算网的用户,也有一个唯一的域名,例如 jlu.se.cuit.cn 表示动态计算网 se.cuit.cn 中的用户 jlu。

在动态计算网中,维护着一张 I/O 设备描述表,动态计算网中的每个设备在表中占一行。动态计算网的 I/O 设备描述如表 1 所列。

表 1 动态计算网的 I/O 设备描述表

设备功能描述(DPD)	上下文特征(CPD)	元件域名(UDN)
Device Property	Context Property	Unit
Description	Description	Domain Name

在动态计算网的 I/O 设备描述表中,DPD 代表设备的功能特征,CPD 代表该设备适合的上下文场景,DN 代表该设备

在动态计算网中的域名。DPD 与 CPD 都使用 SDL (Specification and Description Language) 语言<sup>[8,9]</sup> 进行描述。本文描述的动态计算网络具有如下的特点:

(1) 动态计算网络自身是一个由若干元件构成的分布式虚拟计算机。“小的”动态计算网络,其各个设备元件可以在一个局域网中;“大的”动态计算网络,其各个设备元件可能在一个广域网中。

(2) 普适计算软件是运行在动态计算网络基础上的。对普适计算软件而言,其下层的运行环境是一台抽象的计算机(包括硬件和支持软件):动态计算网络。分布在局域网的动态计算网络和分布在广域网的动态计算网络对其上层运行的普适计算软件而言是透明的。因此,基于动态计算网络的普适计算软件具有天生的分布式特点。

### 3 基于动态计算网络的普适计算软件

动态计算网络是物联网普适计算软件的运行基础。基于动态计算网的普适计算软件代码分为:非 I/O 部分与 I/O 部分,非 I/O 代码和 I/O 代码都在动态计算网的运算元件中执行。I/O 代码的执行过程需要访问动态计算网的 I/O 设备,遵循如下的过程:

(1) 在访问 I/O 设备前,使用 `get_device(dpd, cpd)` 来获得设备,其中参数 `dpd` 是设备功能描述,参数 `cpd` 是上下文特征。`get_device()` 将触发动态计算网的设备动态匹配过程,设备动态匹配过程将依据普适计算软件 `get_device(dpd, cpd)` 传入的参数 `dpd` 和 `cpd` 在动态计算网的 I/O 设备描述表中查找匹配。如果查找到匹配的设备,就返回该设备在动态计算网中的域名 `udn`。

(2) 在上述设备动态匹配过程中, `dpd` 与 `dcp` 都必须匹配,这表明在上下文环境中,设备的能力是符合要求的,例如:在只能发声的上下文环境下,找到了正确的发声设备。如果无法实现 `dpd` 与 `dcp` 完全匹配,则将依据 `cpd` 和 `dcp` 的要求,寻找替代设备,并返回 `udn`。依据 `cpd` 和 `dcp`,寻找替代设备的过程,就是动态计算网普适计算软件 I/O 过程进行降级执行的过程。寻找替代设备的方法可以依据预先设备的标准,或者采用人工智能的方法。

I/O 操作的降级执行是一个复杂的处理过程,既要考虑 I/O 操作的特征,也要考虑用户所处的上下文情景。许多学者对基于上下文情景的服务降级执行方法进行了研究,提出了许多智能化的方法,例如:InterPlay 中间件<sup>[10]</sup> 基于简单的上下文可以推断用户的意图并给出服务执行的建议,使用类似 SDL 的语言来描述设备与任务,并且完成设备与任务的匹配。Middle-Where 中间件<sup>[11]</sup> 引入了概率推理方法,支持基于位置感知基础上的概率推理上下文感知,在合适的位置上,用合适的服务执行方式给用户提供服务。Middle-Where 中间件提供了一种基于位置的服务降级执行的方法。Ranganathan 等人在文献<sup>[12]</sup> 中使用一阶谓词逻辑、线性时序逻辑推理机制来实现服务与场景的匹配。MICA 中间件<sup>[13]</sup> 提供了一种根据用户的行为学习用户的偏好的方法,其使服务逐渐地适应用户的行为习惯,从而让服务以合适的方式执行,以满足用户的需求。

## 4 基于动态计算网络的普适计算功能实现

### 4.1 服务发现方法与上下文感知

在基于动态计算网络的普适计算软件中,服务发现与上下文感知,主要是由两个重要的输入数据来触发的:第一个是用户的域名,它标记了需要服务的用户及此用户所属的动态计算网络 `P_DCN`。第二个是读取此用户信息的读取设备,这台读取设备读取了此用户的信息,因此是最靠近此用户的(靠近此用户指物理空间距离靠近),代表了此用户当时所处的上下文环境。从这台设备的域名上,可以解析到该设备所属的动态计算网络 `D_DCN`,从而获得此用户附近的动态计算网络。

从获得的 `P_DCN` 中,可以查找到用户所需要的服务,动态计算网可以依据 `P_DCN` 和 `D_DCN` 的差异(包括物理位置、网络差异、计算性能等),决定是否进行服务迁移,或者决定服务迁移的方法。

例如,如果 `P_DCN` 和 `D_DCN` 之间的带宽很小,而服务所需的数据传输量又很大,那么服务可能会从 `P_DCN` 迁移到 `D_DCN`,从而解决服务带宽的矛盾。

### 4.2 动态计算网络服务迁移方法

基于动态计算网络的普适计算软件,由于动态计算网络的天生分布式特征,服务迁移实现非常简单,具体实现方法有两种形式:

(1) 服务与数据都不迁移:使用远距离 I/O 方式

动态计算网络是由各种元件通过网络组成的,具有天生的分布式特征。在实现服务迁移的时候,最简单的方式是使用远距离 I/O 的方式,也就是动态计算网络通过 I/O 设备的动态组成,使用在用户物理位置附近的 I/O 设备来完成 I/O 操作,从而实现服务“围绕人”的服务迁移效果。

例如,一个用户在校的动态计算网中阅读一份文档,文字阅读服务的服务进程、数据和状态都在学校动态计算网中。当此用户回到家中的时候,可以把家中的显示器与输入设备动态接入到学校的动态计算网络中,方法是修改动态计算网络 `se.cuit.cn` 中的显示设备域名指向家中的显示器与输入设备。这样,学校动态计算网中的文字阅读服务,就可以“远程”地呈现在此用户的家中,实现基于远距离 I/O 方式的服务迁移,如图 2 所示。

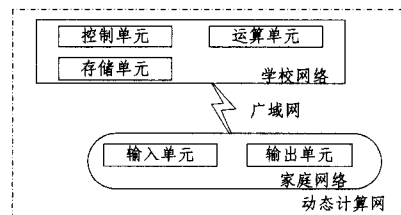


图 2 基于远距离 I/O 的服务迁移

(2) 从一个动态计算网络到另外一个动态计算网络:服务与数据的迁移

基于远距离 I/O 方式的服务迁移,其优点是实现简单,缺点是当距离很远时,由于网络带宽等因素,可能导致其性能低下,一种改进性能的方法是使用服务与数据全迁移的方式。

如图 3 所示,在前述的用户在学校阅读文档的例子中,如

果用户回到家后,在家中通过一个传感器获得了此用户的域名 jlu. se. cuit. cn,又通过此用户的域名 jlu. se. cuit. cn 知道了用户属于动态计算网 se. cuit. cn,然后查找动态计算网 se. cuit. cn 就可以获得此用户的服务。

如果通过服务迁移模块的评估后,决定应该进行服务迁移,就启动服务迁移过程,将服务代码、数据和服务状态,全部迁移到目标执行的动态计算网络 myhome. cuit. cn 中。

本方法的一种变化是服务代码迁移,数据不迁移,或者是数据迁移,而服务代码不迁移。

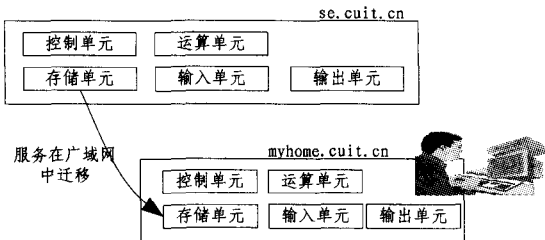


图3 服务在动态计算网络之间迁移

## 5 验证实验

本文所提基于动态计算网络的普适计算框架,通过了验证实验。在本实验中,构建一个物联网普适计算实验环境,整个环境由两个动态计算网络构成,分别为 se. cuit. cn 和 myhome. cuit. cn。每个动态计算网络包括:运行单元、控制单元、高级输出单元(高分辨率屏幕等设备)、低级输出单元(LED点阵显示屏等设备)、高级输入单元(计算机键盘)、低级输入单元(3按钮键盘)、RFID感应单元、存储单元、高级发声单元(计算机声卡)、低级发声单元(计算机 speaker 喇叭)。

在上述动态计算网络实验环境中,本文进行了如下的验证实验:

(1)通过由计算机模拟而成的各个单元在无线网络中的自动发现与控制,验证了动态计算网络的发现与控制协议。

(2)在程序中引入高级单元与低级单元(高级显示单元、低级显示单元),针对高级单元的普适计算软件,在运行过程中,高级显示单元失效,则由低级显示单元替换后,程序继续运行。

(3)通过平板电脑、手机、笔记本电脑等设备单元的协作,实现一个人在学校办公室笔记本电脑上阅读文档,然后人移动到家中,通过基于 RFID 的上下文感应,自动将服务迁移到家中的动态计算网络 myhome. cuit. cn,实现服务迁移。当家中仅有低级发声设备的时候,可以自动使用低级发声设备发声,实现服务降级执行。

通过上述实验,验证了本文提出的基于动态计算网的普适计算框架在模拟物联网环境中可以实现普适计算的服务发现、服务迁移与上下文感知功能。

**结束语** 本文提出的基于动态计算网络的普适计算框架具有如下特点:

(1)降低物联网普适计算运行环境“粒度”。本文根据物联网的特点,将物联网环境中的普适计算硬件环境抽象为动态计算网络。动态计算网络在硬件粒度上突破了物联网软件环境的传统影响,从而将软件的自适应推向了更低的粒度,能

实现更好的软件自适应性。

(2)动态计算网络天生的分布式特征,简化了普适计算软件实现的复杂度。动态计算网络自身是由多个设备单元通过网络动态组成的,可以在一个局域网中,也可以在一个广域网中。在普适计算软件设计的时候,仅仅需要在一个抽象的动态计算网络中进行程序设计。在具体运行的时候,动态计算网络会自动地处理服务迁移、上下文感知等问题,极大地简化了普适计算软件实现的复杂度。

(3)为 I/O 设备提供基于能力描述的降级执行方法。本文提出的动态计算网络可以实现基于 I/O 设备能力描述的降级执行方法。软件采用统一的接口去访问硬件,当硬件发生变化时,由动态计算网络自动选择不同的 I/O 设备来提供服务,实现了软件 I/O 操作的降级执行,特别适用于设备动态变化频繁的物联网中的普适计算软件的实现。

在后续工作中,还有若干重要的问题需要进一步研究,主要包括:(1)是在代码加载的时候进行 I/O 设备匹配工作(运行前匹配,特点是高效,缺点是运行时可能失效),还是在代码运行到 I/O 语句的时候进行设备匹配工作(运行时匹配,特点是动态适应性强,缺点是效率低下),各自具有不同的特点。在本项目的后期执行中,将对这个问题进行重点研究。(2)I/O 操作的降级执行,如何根据上下文特征选择最适合的 I/O 降级执行设备,在后续工作中需要进一步研究。(3)需要进一步研究如何确保动态计算网络的安全性问题。

## 参考文献

- [1] Want R, Pering T, Tennenhouse D. Comparing Autonomic and Proactive Computing[J]. IBM Systems Journal, 2003, 42(1): 129-135
- [2] Weissman J B, Zhang Zhi-li, Environments S. Middleware Building Blocks for Pervasive Network Computing (A Position Paper)[C]// the NSF Workshop on Developing an Infrastructure for Mobile and Wireless Systems, October 2001; 90-105
- [3] Roussaki I G, Strimpakou M A, Anagnostou M E. Designing next generation middleware for context-aware ubiquitous and pervasive computing[J]. International Journal of Ad Hoc and Ubiquitous Computing, 2007, 2(3): 197-206
- [4] Campo C, Muoz M, Perea J C. PDP and GSDL: A new service discovery middleware to support spontaneous interactions in pervasive systems[C]// Proceedings of the 3rd IEEE Conference on Pervasive Computing and Communications Workshops, 2005: 178-182
- [5] Sharmin M, Ahmed S, Ahamed S I. SAFE-RD (Secure, Adaptive, Fault Tolerant, and Efficient Resource Discovery) in pervasive computing environments[C]// Proceeding of the IEEE International on Information Technology, 2005; 271-276
- [6] Corradi A, Montanari R, Tibaldi D. A context-centric security middleware for service provisioning in pervasive computing[C]// Proceedings of the symposium on Applications and the Internet, IEEE Computer Society, 2005; 421-429
- [7] Cheng Shang-wen, Garlan D, Schmerl B R, et al. Software Architecture-Based Adaptation for Pervasive Systems [C] // ARCS 2002. 2002; 67-82

- [8] Sousa R M, Putnik G D. Formal description technique SDL for manufacturing systems specification and description[C]//APMS 1999, 1999, 449-456
- [9] Ellsberger J, Hogrefe D, Sarma A. SDL[M]. Formal Object-oriented Language for Communicating Systems, Prentice Hall, 1998
- [10] Messer A, Kunjithapatham A, Sheshagiri M. Interplay: A middleware for seamless device integration and task orchestration in a networked home[C]//Proceeding of the Fourth Annual IEEE International Conference on Pervasive Computing and Communications. 2006;296-307
- [11] Ranganathan A, Al-Muhtadi J, Chetan S. MiddleWhere: A mid-

dleware for location awareness in ubiquitous computing applications[C]//Proceeding of the 5th ACM International Conference on Middleware. 2004;397-416

- [12] Ranganathan A, Campbell R H. A middleware for context-aware Agents in ubiquitous computing environments[C]//Proceeding of the ACM International Middleware Conference. 2003;143-161
- [13] Kadous M W, Sammut C. MICA: Pervasive middleware for learning, sharing and talking[C]// Proceeding of the Second IEEE Conference on Pervasive Computing and Communications Workshop. 2004;176-180
- [14] Lyytinen K, Yoo Y. Issues and challenges in ubiquitous computing[J]. Communications of the ACM, 2002, 45(12): 62-65

(上接第 38 页)

平均时间  $T_{avg\_ack}$ 。求解方法是让发布者同样发送 4 字节的数据,因而可以采用往返时间的一半作为  $T_{avg\_ack}$ 。从而:

$$T_{avg\_delay} = \frac{\sum_{i=1}^n (T_i + T_i')}{n} - T_{avg\_ack} \quad (2)$$

为了模拟动态适应性和容错性,为每个节点(包括发布者和订阅者)都配备一个失效备份节点。不失一般性,为了降低测试复杂度,试验中将备份节点采用静态配置的方式,为了模拟节点动态变化情形,采用周期性更新节点路由表的方式,心跳检测的临界值设为 0.5ms,利用失效节点率(即失效的节点数占总节点数的比率)来评估网络的动态变化情况。

实验环境搭建了 28 个节点,每 14 个节点在一个网段内(备份节点在相同的网段内),每个节点都是普通的 PC,网络采用 1000Mbps 交换机和 10M 集线器两种连接方式。根据上述测试方法对中间件反复测试,取其平均值,结果如图 3 所示,其中数据大小的单位为 1024 字节(kB),时延的单位为微秒( $\mu$ s)。

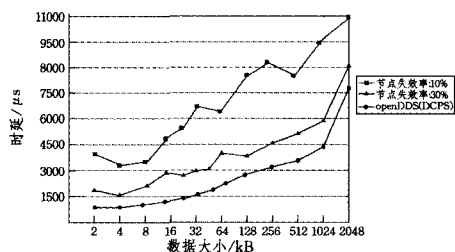


图 3 测试结果

由图 3 可知,时延总体随着数据量的增加而增加,而在数据大小相等的情况下,openDDS 具有最小的时延,这也验证了在网络拓扑固定的情况下,P2P 路由相比集中式路由,花费的时间更多;在存在节点失效率小于 10%的情况下,时延能够控制在 10ms 内,可以满足协同作战环境的实时性要求(毫秒级);而在失效率为 30%时,可以看到时延明显增大,并且随着传输数据的增多出现跳跃现象,说明路由表的更新严重影响了时延的稳定性。从上面的试验结果可以看出,实现的自适应中间件在动态变化不是很频繁的情况下能够满足实时要求。

**结束语** 本文通过引入 P2P 架构和 Chord 协议,对现有 DDS 中间件从动态服务发现和容错两个角度进行了改进;并在传统心跳检测模型的基础上,提出了加速推拉模型,以提高系统的实时容错性。实验结果验证了其具有实时、容错、自适应的特点。在一个主题多个订阅的情况下,利用组播技术可以大大提高传输效率,因此下一步的研究方向就是结合组播协议和 Chord 协议来优化现有的路由查找协议。

## 参考文献

- [1] <http://www.nsa.gov/ia/industry/gig.cfm>
- [2] Object Management Group. Data Distribution Service for Real-time Systems[R]. Version1. 2, Jan. 2007
- [3] Object Management Group. High-performance CORBA Specification[R]. Version1. 2, Jan. 2005
- [4] 翟立东,刘元安,马晓雷,等.发布/订阅通信机制在移动 Ad Hoc 网络中的应用[J].北京邮电大学学报,2008,31(2):30-33
- [5] 韩松,张晓林,占巍,等.基于空中指挥节点的信息分发模型及时延分析[J].系统工程与电子技术,2009,31(11):2677-2681
- [6] Shirky C. Wat is P2P and what isn't[C]//O'Reilly's Emerging Technology Conference. 2002
- [7] An Introduction of openSlice DDS. pdf [EB/OL]. <http://www.openslice.org>
- [8] RTI DDS Qos and Features. pdf. [EB/OL] <http://www.rti.com>.
- [9] Steven Stallion. Using Reliable Multicast for Data Distribution with OpenDDS [EB/OL]. <http://mnb.ociweb.com/mnb/MiddlewareNewsBrief-201002.html>
- [10] Clarke L. A distributed decentralized information storage and retrieval system[D]. Master's thesis. University of Edinburgh, 1999
- [11] Stoica, Morris R, Karger D, et al. Chord: A scalable peer-to-peer lookup service for internet application[C]//Proceedings ACM SIGCOMM. 2001
- [12] Joung Y J, Wang J C. Chord: A two-layer Chord for reducing maintenance overhead via heterogeneity [J]. Computer Networks, 2007, 51(3): 712-731
- [13] 贺建立,陈榕,顾伟楠.一个事件驱动的中间件平台[J].计算机科学,2010,37(5):107-111