

## 一种面向服务的可靠多媒体传输算法<sup>\*</sup>

张 胜<sup>1,2</sup>, 钱柱中<sup>1,2+</sup>, 陆桑璐<sup>1,2</sup>

<sup>1</sup>(南京大学 计算机软件新技术国家重点实验室,江苏 南京 210093)

<sup>2</sup>(南京大学 计算机科学与技术系,江苏 南京 210093)

### Service-Oriented Reliable Multimedia Delivery Algorithm

ZHANG Sheng<sup>1,2</sup>, QIAN Zhu-Zhong<sup>1,2+</sup>, LU Sang-Lu<sup>1,2</sup>

<sup>1</sup>(State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210093, China)

<sup>2</sup>(Department of Computer Science and Technology, Nanjing University, Nanjing 210093, China)

+ Corresponding author: E-mail: qzz@nju.edu.cn, http://www.nju.edu.cn

Zhang S, Qian ZZ, Lu SL. Service-Oriented reliable multimedia delivery algorithm. *Journal of Software*, 2010,21(2):277-286. <http://www.jos.org.cn/1000-9825/3788.htm>

**Abstract:** This paper proposes an efficient algorithm LD/RPath (lowest delay/reliability path) for service-oriented reliable multimedia delivery in a pervasive environment. LD/RPath estimates the dynamic data volume on service nodes and links through reasonable data volume approximation. And the data splitting technique is imported to convert the node delay into the edge delay. In the mean time, the reliability of nodes is considered as a coefficient of delay, so that the multimedia delivery problem is transformed into a conventional shortest path problem. Simulation results prove that LD/RPath achieves good path selection performance while imposing low overhead to the system.

**Key words:** pervasive space; service-oriented architecture; multimedia delivery; delay; reliability

**摘 要:** 针对普适环境中可靠的多媒体传输问题,提出了 LD/RPath(lowest delay/reliability path)算法.LD/RPath 算法通过合理的数据量近似来拟合服务节点和传输边上动态变化的数据量,引入节点分裂技术将节点时延转化为边时延,同时考虑了节点可靠性因素,并作为时延的系数.通过这样的处理,该传输问题可以转化为传统的最短路径问题来解决.实验结果表明,LD/Rpath 能够达到较好的路径选择效果,且系统开销较低.

**关键词:** 普适环境;面向服务架构;多媒体传输;时延;可靠性

中图法分类号: TP311 文献标识码: A

实时多媒体传输系统具有重要的应用价值,广泛应用于实时监视系统、视频会议等应用中.然而随着移动设备能力的提升和无线通信技术的发展,传统的单一网络环境逐渐变为异构设备和网络构成的普适计算环境.

在普适环境中,不同链路的传输能力差异巨大,特别是无线链路带宽较小且不稳定;各个节点的处理能力差异巨大,尤其是移动节点的处理能力往往较弱.这些特点使得实时多媒体传输系统的可靠性无法得到保证.

\* Supported by the National Natural Science Foundation of China under Grant Nos.90718031, 60721002 (国家自然科学基金); the National Basic Research Program of China under Grant No.2009CB320705 (国家重点基础研究发展计划(973)); the Jiangsu Provincial Natural Science Foundation of China under Grant No.BK2008264 (江苏省自然科学基金)

Received 2009-06-15; Revised 2009-09-11; Accepted 2009-12-07

面向服务的构架(service oriented architecture,简称SOA)<sup>[1]</sup>利用可用服务可以快速构建松耦合的应用.作为SOA的关键技术,服务组合(service composition)<sup>[2]</sup>可以把独立、分布、可用的基本服务组合起来,满足用户复杂的业务需求,使其能够适应于普适计算环境.这些特点使得SOA适于利用现有的分布的服务资源,动态地构建松耦合的普适环境中的多媒体传输系统.

在面向服务的实时多媒体传输系统中,若干多媒体处理服务预先部署于网络中的不同节点.这些服务可以分为功能服务和非功能服务.功能服务包括字幕嵌入、添加标签和编码转化等对源信息进行必要的处理,满足用户的功能需求;非功能服务包括媒体压缩、采样等,往往能够降低数据传输量,从而减少数据传输的时延.尤其对于采用无线接入的移动用户而言,这些非功能性服务能够有效改善用户所获得服务的可靠性,但其本身有处理时延,并会降低多媒体的影音质量<sup>[3]</sup>.系统的功能是:当接收到用户的服务请求时,系统从数据源开始,经过若干功能服务和非功能服务处理节点,构建一条多媒体信息处理传送链,将多媒体信息以较低的时延和较高的可靠性传递给终端用户.在本文中,我们假设用户的服务请求是逐个到达的,相互之间没有影响.

在面向服务的实时多媒体传输系统中,具有相同功能需求的服务请求往往有多种服务组合方式能够完成.而且在运行时,每一个服务功能组件又可以选择不同的服务副本来实现.在实时多媒体系统中,用户对时延较为敏感,如果多媒体信息延迟过长,有频繁的抖动,用户是无法接受的.此外,服务使用者无法控制服务,且在普适环境中,服务还受到设备、环境等因素的影响,服务个体之间具有较大的可靠性差异,而服务个体的可靠性最终也决定了组合服务预期的性能.因此,如何在普适环境中选择一系列合适的服务副本,构成一个低时延且具有高可信的服务路径是一个具有挑战性的问题.

Zeng等人<sup>[4]</sup>认为,服务副本的选择应该在运行时刻而不是设计时刻决定,因为很多基本服务的质量参数在运行之前是无法获得的.他将其转化为一个质量驱动的线性优化问题来解决.Gu等人<sup>[5,6]</sup>提出了一个集成的P2P服务组合框架SpiderNet,通过启发式的方法逐跳地选择服务副本以满足多个质量要求;此外,SpiderNet还对负载均衡有一定的保证.Roman等人<sup>[7]</sup>考虑了服务副本之间的负载均衡,提出了LIAC(least-inverse-available-capacity)算法,使得选择路径时保证服务副本之间的负载达到均衡,并且采用了捎带机制,使得LIAC更加高效.Roman等人<sup>[8]</sup>还提出了广域网中基于服务覆盖网的服务组合容错系统.Kalasapur等人<sup>[9]</sup>提出了一个动态服务组合框架,通过层次式服务覆盖网来处理普适环境中的移动性和动态性.

本文从实时多媒体传输系统出发,从两个角度来考察可靠多媒体传输的服务路径的选择标准:一是实时性.用户希望得到的是实时的数据,如果数据传输有很大的延迟,那么对用户而言就不是可信赖的.二是可靠性.由于环境的动态性和异构性,每个服务副本都有可能出现差错.如果系统选择的服务路径出现差错,系统就要重新选择一条服务路径来完成服务请求,从而导致系统的负荷增加.而且路径切换使得时延也进一步地增加,那么对用户而言,系统提供的服务就不再可靠.本文的研究目标是,在给定服务请求时,选择合适的服务副本构建具有低时延和高可靠性的多媒体传输路径.

本文第1节建立普适环境中多媒体传输系统模型.第2节介绍LD/RPath算法.第3节给出实验结果和分析.第4节是对本文工作的总结和下一步的工作.

## 1 普适环境中多媒体传输系统模型

在一个普适多媒体传输环境中,多媒体服务节点和网络链路的参数直接影响算法的路径选择效果.本节主要说明模型的参数,并给出问题定义.

### 1.1 参数定义

结合研究的问题,我们定义下面的几个相关参数:带宽(bandwidth)、单位处理时间(unit processing time)、压缩比(IORatio)和可靠性(reliability).为简单起见,我们将不区分传输和传播.在下面的描述中,我们用传输来表示数据从一个节点开始传输,到下一个节点完全接受的过程.

带宽:带宽指的是在多媒体传输环境中,两个服务节点之间的当前链路传输能力,我们用  $B$  表示.

单位处理时间:单位处理时间指的是在某一个服务节点上处理单位数据所需要的时间,用  $o$  表示,这是根据

多媒体信息特征提出的.多媒体系统是数据密集型应用,数据量的大小对处理时间有很大的影响.此外,由于服务节点的异构性,不同服务节点的计算能力有很大的差异.因此,我们采用处理单位数据的时间来衡量节点的处理能力.

压缩比:数据经过某一种服务处理后,我们用这种服务处理之后和处理之前的数据量之比来定义压缩比,用  $r$  表示.这也是与多媒体系统的特性密切相关的,不同功能的服务往往会对数据量造成不同的影响.比如,字幕嵌入对数据量可能几乎不产生影响,但是媒体压缩往往会将数据量变小很多.请注意,不同功能服务的压缩比一般是不同的,但是同一种服务不同副本的压缩比是相同的.

可靠性:可靠性表示一个服务处理节点的可靠程度,用  $e$  表示,由历史数据得来.本文延用了文献[4]中的定义:考察在过去的一段时间内系统对某个服务节点的  $K$  次调用,若成功调用的次数为  $C$ ,那么有  $e=C/K$ .由于可靠性是乘积型的参数<sup>[10]</sup>,一条服务路径的可靠性指的是这条路径上的所有服务节点的可靠性的乘积(这里,我们认为所有的链路都是可靠的).

时延:时延包括传输时延和数据处理时延,用  $d$  表示.由于时延是累加性的参数<sup>[10]</sup>,一条服务路径的时延指的是这条服务路径上所有节点的处理时延和链路的传输时延的总和.

### 1.2 功能图和服务图

由于对于用户同一类型的服务,一般有多种功能服务的组合均能满足需求,为此,我们采用功能图(functional graph,简称FG)来表示系统中所有可能的功能服务的组合关系.假设系统中有 5 种基本的多媒体处理服务,如图 1 所示,那么从  $f_0$  出发到  $f_4$  一共有 4 种可能的服务组合方式,分别是  $(f_0f_1f_4, f_0f_1f_3f_4, f_0f_1f_2f_4, f_0f_1f_2f_3f_4)$ .

功能图描述的是系统功能上的抽象.进一步地,我们考虑每一种服务有多个服务副本,把每一种服务的全部副本放到功能图中进行扩展就形成了如图 2 所示的服务图(service graph,简称SG).服务图描述的是所有服务副本之间的依赖关系,概括了系统的全部信息.在图 2 中,假设服务  $f_0$  有 1 个副本,  $f_1$  有 2 个副本,  $f_2$  有 1 个副本,  $f_3$  有 2 个副本,  $f_4$  有 2 个副本,并且增加一个  $S_i$  作为多媒体数据最终传送到达的目的节点.我们可以看出,对应于 FG 中的功能路径  $f_0f_1f_2f_4$ ,如果选择  $S_1^2$  作为  $f_1$  的服务实例,  $S_4^2$  作为  $f_4$  的服务实例,那么在 SG 中具体的服务路径为  $S_0S_1^2S_2S_4^2$ .同时定义,如果存在有向边从  $S_i$  指向  $S_j$ ,则称  $S_i$  为  $S_j$  的前驱节点,  $S_j$  为  $S_i$  的后继节点.

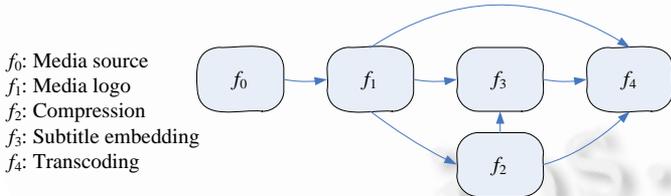


Fig.1 Functional graph  
图 1 功能图

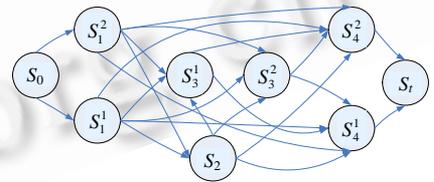


Fig.2 Service graph  
图 2 服务图

### 1.3 问题定义

每个服务副本可以表示为一个四元组  $(f, r, o, e)$ ,依次表示该服务副本的服务种类、压缩率、单位处理时间和可靠性.例如,  $S_1^2$  表示为  $(f_1, r_1, o_{1,2}, e_{1,2})$ .系统的目标是在从  $S_0$  到  $S_i$  的所有路径中选择一条时延尽可能小,同时可靠性尽可能大的路径.在仅仅考虑时延的情况下,由于数据量的变化,这实际上相当于是一个多约束路径(multi-constrained path,简称MCP)问题.Wang等人<sup>[10]</sup>证明了这个问题是NP-完全问题.指数时间复杂度的算法由于耗时较多,在实时系统中不适用.因此,我们需要时间复杂度较小同时效果较好的算法.

一条服务路径的时延  $\delta_p$  是数据在链路上的传输时延和数据在节点上处理时延的总和.我们设  $m$  为原始的数据量,则  $m \cdot o_0$  表示数据在  $S_0$  上面的处理时延.  $\delta_p$  可以表示为



```

6.      if  $index(j) \leq index(i)$  then      {保证有向边的尾节点序号大于首节点序号}
7.           $index(j) \leftarrow index(i)+1$       {节点序号调整}
8.           $flag \leftarrow true$ 
9.      end if
10.     end for
11. end while
12. for each  $index[i]$  do
13.      $c \leftarrow 0, sum \leftarrow 0$ 
14.     for each node  $v \in SG$  do          {找到每个节点的所有前驱节点}
15.         if  $v$  has a service link to  $index[i]$  then
16.              $c \leftarrow c+1, sum \leftarrow sum+m[v]$       {累加其所有前驱节点传出的数据量,并计数}
17.         end if
18.     end for
19.      $m[index[i]] = sum/c$           {取所有的前驱节点传出的数据量的算术平均}
20. end for
21. return  $m[0, N_n-1]$ 

```

## 2.2 节点分裂

最短路径算法<sup>[11]</sup>解决的是权值在边上的情况下的单源最短路径问题,然而我们的服务图上的节点上也有代价(权值),于是我们需要对服务图作一些转化,使其能够适用最短路径算法.

Choi等人<sup>[12]</sup>提出节点分裂的方法来转化某些网络问题.我们借鉴了这一方法.例如在图2中,我们将 $S_2$ 分裂为两个节点 $S_{2,1}$ 和 $S_{2,2}$ ,然后让 $S_2$ 的所有前驱节点都连接到 $S_{2,1}$ ,让 $S_{2,2}$ 连接到 $S_2$ 的所有后继节点.原来 $S_2$ 节点上面的数据处理时延,现在用 $S_{2,1}$ 和 $S_{2,2}$ 之间的边的代价来表示,我们称像 $S_{2,1}$ 和 $S_{2,2}$ 之间这样的边为内部边(inner-link).

我们对服务图中每一个节点都使用分裂过程,增加了新的节点,消除了顶点上面的代价,使得转化后的服务图中代价只存在边上.

## 2.3 可靠性转换

由于普适环境的动态性和移动性,服务副本节点可能会失效.系统的要求是要找出一条时延尽可能小,同时可靠性尽可能高的服务路径.我们的想法是,让可靠性这个参数体现到节点的处理时延上.

我们借鉴了以往的QoS相关的研究工作<sup>[13,14]</sup>,给服务图中每条内部边增加一个系数,这个系数是这条内部边代表的服务节点的可靠性 $e$ 的倒数.这样处理之后,可靠性较高的服务副本节点分配的系数比较小,可靠性较低的服务副本节点分配的系数比较大.这些内部边新的时延代价是原来的代价乘以各自的系数,作为内部边新的处理时延.

这一过程的想法很直观:对于最短路径算法选择的单源最短路径,可靠性转换使得可靠性较低的服务副本所在的内部边的代价变得比较高,使得可靠性较高的服务副本所在的内部边的代价变得比较低,从而最短路径算法在选择最短路径时偏向于选择可靠性较高的服务副本.也就是说,可靠性较高的服务副本节点更容易被算法选择.这使得LD/Rpath算法选择出的路径的时延较小,同时可靠性较高,在两者之间达到一个平衡.

## 2.4 代价标注

服务图经过以上3步的转换后,我们现在可以对其中的每条边来标注时延代价.

如果某条边是一条内部边,那么它的代价是如下的形式:

$$\frac{1}{e_i} \cdot m_i \cdot o_i \quad (4)$$

其中, $e_i$ 和 $o_i$ 是这条内部边所代表的服务副本节点的可靠性和单位处理时间, $m_i$ 是近似数据量;否则,这条边的代

价是如下的形式:

$$\frac{m_i}{B_i} \tag{5}$$

其中,  $B_i$  是这条服务边的带宽,  $m_i$  是这条服务边上传输的近似数据量.

图 3 展示了一个简单的转换例子. 图 3(a) 是某个服务图, 假设图中每个服务只有 1 个服务副本(多个副本的情况类似处理), 节点分裂之后, 转换为图 3(b) 所示的结构.  $S_0$  和  $S_i$  是数据的源节点和目的节点.

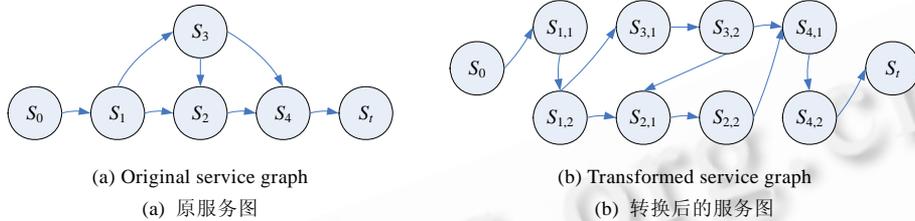


Fig.3 An example of transforming

图 3 一个转换例子

综合考虑数据量近似和可靠性转换, 假设在  $S_0$  处的初始数据量为  $m$ , 那么图 3(b) 中  $S_{1,1}$  和  $S_{1,2}$  之间的内部边的代价为  $\frac{1}{e_1} \cdot m \cdot r_0 \cdot \alpha_1$ , 表示节点  $S_1$  的数据处理时延乘以  $S_1$  的可靠性的倒数;  $S_{3,2}$  和  $S_{2,1}$  之间的边的代价为  $\frac{m \cdot r_0 \cdot r_1 \cdot r_3}{B_{3,2}}$ , 由链路上的传输时延构成;  $S_{2,2}$  和  $S_{4,1}$  之间的边上的代价为  $\frac{m \cdot r_0 \cdot r_1 \cdot r_2 + m \cdot r_0 \cdot r_1 \cdot r_3 \cdot r_2}{2B_{2,2-4,1}}$ , 由于这条边上的数据量有两个可能的值(一是从  $S_0$  经  $S_1$  和  $S_3$  传输到  $S_2$ , 二是从  $S_0$  经  $S_1$  传输到  $S_2$ ), 故代价中有数据量近似部分;  $S_{4,1}$  和  $S_{4,2}$  之间的内部边的代价为  $\frac{1}{e_4} \cdot \frac{m \cdot r_0 \cdot r_1 \cdot r_3 + \frac{m \cdot r_0 \cdot r_1 \cdot r_2 + m \cdot r_0 \cdot r_1 \cdot r_3 \cdot r_2}{2}}{2} \cdot \alpha_4$ , 这条内部边所代表的服务节点有两个前驱节点, 而且其中一个前驱节点仍有两个前驱节点. 所以, 式中有两处数据量近似部分.

### 2.5 时间复杂度分析

对服务图进行代价标注之后, 就可以用最短路径算法来解决问题. 求出最短路径之后, 我们将产生的节点序列还原到转换之前的服务图中去, 可以得到 LD/RPath 产生的路径. 我们让  $N_n$  和  $N_e$  分别来表示原服务图中的节点个数和边的条数. 在数据量近似算法中的第 1 部分, 由于每个服务节点最多与  $N_n$  个节点连接, 所以遍历所有边的过程最多进行  $N_n$  次, 此部分的复杂度是  $O(N_n N_e)$ ; 第 2 部分中同样每个节点被访问一次, 同时找出其所有可能的前驱节点, 此部分的复杂度也是  $O(N_n N_e)$ . 节点分裂与数据量近似的过程很相似, 需要处理每一个节点, 然后判断前驱节点和后继节点时需要访问  $N_e$  条边, 因此, 这部分的复杂度为  $O(N_n N_e)$ . 在可靠性转换中, 由于图中只有  $N_n$  个内部边, 故复杂度为  $O(N_n)$ . 代价标注需要处理每一条边, 复杂度为  $O(N_e)$ . 最短路径算法我们使用 Dijkstra 算法, 其复杂度为  $O(n^2)$ , 其中,  $n$  表示图中顶点的个数. 根据节点的分裂过程有  $n < 2N_n$ , 同时, 根据完全图的规律有  $N_e \leq N_n^2$ , 因此 LD/RPath 算法的总的时间复杂度为

$$O(N_n N_e + N_n N_e + N_n + N_e + n^2) = O(N_n^3).$$

### 3 仿真实验

本节通过仿真实验的方法评估 LD/RPath 算法的运行效率和路径选择效果. 仿真实验在 Inter(R) Core(TM)2 Duo CPU E8200 @ 2.66GHz, 2GB RAM 的机器上执行.

### 3.1 评价方法

仿真实验的目的主要有两个:一是评估 LD/RPath 算法的运行效率,二是评估算法的选择效果。

我们实现了两种其他算法来对比运行效率:一是随机(random)算法,Random 算法在选择服务路径时,随机挑选当前节点的某一个后继节点作为下一步的处理节点;二是最优化(optimal)算法,Optimal 算法遍历所有可能的路径,然后选择出最优的路径。

关于算法的选择结果对比,我们在不同的网络规模下将 LD/RPath 算法、Random 算法和 Optimal 算法分别执行多次,将它们产生的路径作比较,分别比较这 3 条路径的时延、可靠性和时延/可靠比。时延越小越好,可靠性越大越好,时延/可靠比越小越好。

Optimal 算法的巨大运算量使其非常耗时。在实时多媒体传输系统中,用户可以容忍的时延是比较小的,故 Optimal 算法在服务图中节点规模达到一定程度后就不再实用。因此在我们的模拟实验中,当 Optimal 算法的时延超过一个较大的值之后,我们只比较 Random 算法和 LD/RPath 算法的运行效率和选择效果。

### 3.2 实验设计

仿真实验用 Java 语言实现,主要包括服务图中各种参数的生成和 3 种算法的实现。

服务图中的参数都是设计为可调的,如我们设置了  $MaxNode=2000$  个服务节点。这些服务节点上,初始时没有多媒体处理服务副本,然后按照不同的网络规模来产生功能图。比如,某种设置中有  $MaxService=7$  种服务,每种服务最少有  $MinInstance=3$  个副本,最多有  $MaxInstance=7$  个副本,然后我们随机产生一个连通的功能图,对每种服务在  $MinInstance$  和  $MaxInstance$  之间随机产生一个实际副本数,再在  $MaxNode$  个节点中随机挑选一些节点来部署服务副本。服务的压缩率、服务节点的单位处理时间和可靠性、链路的带宽均按照一定的正态分布产生,这样就产生了服务图。我们可以通过  $MaxNode$ ,  $MaxService$ ,  $MinInstance$  和  $MaxInstance$  等参数来调节服务图的拓扑规模,进行不同网络拓扑规模上的对比实验。

在 3 种算法的设计中,Random 算法较容易实现,Optimal 算法动态地申请内存以保存所有可能的路径,LD/RPath 算法按照前面一节中所述的 4 个步骤进行处理。

### 3.3 实验结果分析

我们首先对 3 种算法的运行时间在不同的网络拓扑规模下进行对比。分别运行 3 种算法 100 次,然后取 100 次运行的平均执行时间来进行对比,如图 4 所示。我们可以看出,LD/RPath 和 Random 算法的运行时间都远远少于 Optimal 算法。当节点数量在 50 左右时,LD/RPath 和 Random 算法运行时间分别在 1s 和 60ms 左右,但此时,Optimal 算法已达 20s。当网络规模变大时,LD/RPath 算法的运行时间几乎与 Random 算法相同,可见,LD/Rpath 算法的运行效率较高。除了图中所示的数据以外,我们还对比了更大规模网络拓扑中各算法的运行时间,在节点数量达到 70 时,Optimal 算法执行时间已超过 80s。在一个实时多媒体传输系统中,80s 的时延显然是不能接受的。故在下面的对比实验中,当节点数目超过 70 时,我们不再考虑 Optimal 算法。

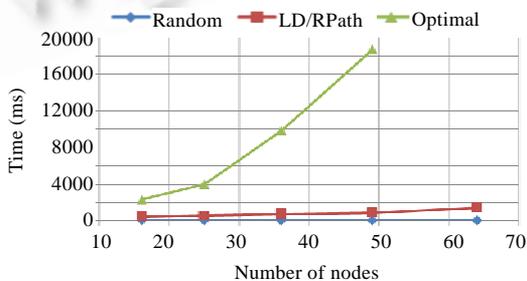


Fig.4 Comparison of execution time of three algorithms

图 4 3 种算法的运行时间对比

然后,我们在不同的网络规模下对 3 种算法的选择效果进行对比.图 5 所示的实验数据是在  $MaxService=7$ ,  $MinInstance=2$ ,  $MaxInstance=7$  的配置下产生的.在此配置下,我们运行 50 组实验,每组实验中网络的拓扑结构都重新随机生成,每组实验包括 1 次 LD/RPath 算法、1 次 Optimal 算法和 20 次 Random 算法,其中,Random 算法的结果取 20 次中最好的结果(Random-Best),时延、可靠性和时延/可靠比的对比如图 5 所示.

图 5(a)表示的是 50 组实验中,3 种算法产生的路径的时延对比.可以看出,LD/RPath 算法很接近 Optimal 算法,都好于 Random-Best.统计信息发现.在 50 组实验中,有 44 次 LD/RPath 算法产生的路径的时延比 Random-Best 算法要小,有 24 次产生的路径与 Optimal 算法相同.图 5(b)是可靠性对比,在 50 组实验中,LD/RPath 算法产生的路径的可靠性有 26 次超过 Random-Best 算法,1 有 6 次超过 Optimal 算法.图 5(c)是时延/可靠比的对比,观察到 LD/RPath 和 Optimal 算法产生的路径的时延/可靠比非常接近,在大多数情况下都优于 Random-Best 算法.统计结果表明,在 50 组实验中,有 42 次 LD/RPath 算法的结果优于 Random-Best 算法,有 9 次优于 Optimal 算法.

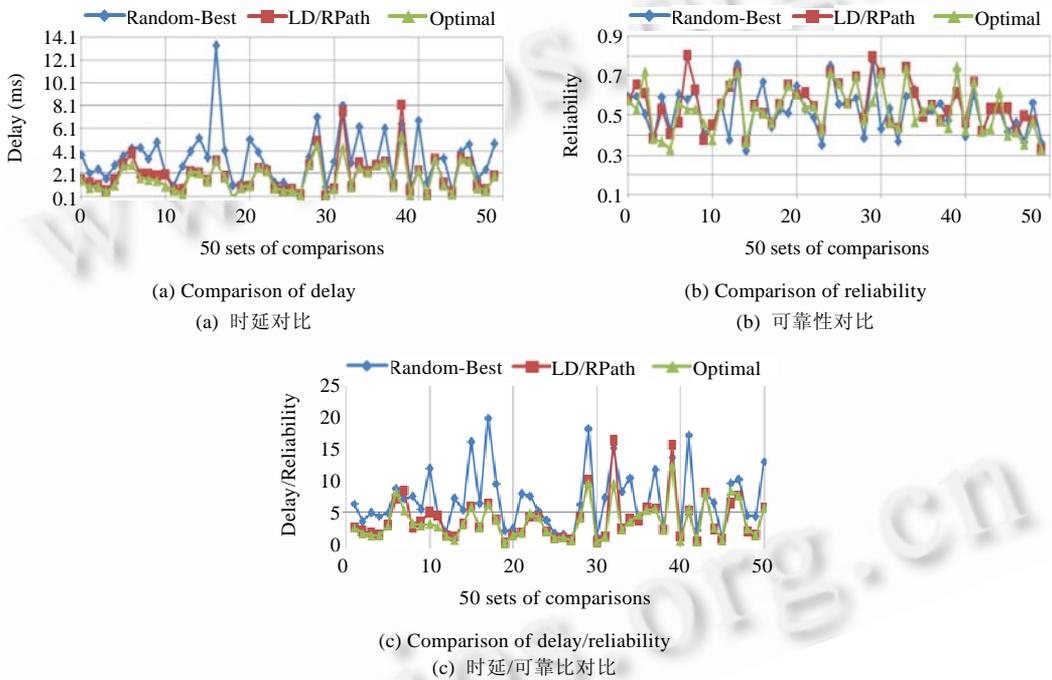


Fig.5 Comparison of delay, reliability and delay/reliability of three algorithms

图 5 3 种算法的时延、可靠性和时延/可靠比对比

此外,我们还比较了 LD/RPath 和 Random 算法在节点规模为 30,60,100,400,900 和 1 600 时的选择效果(这里没有对比 Optimal 算法,因为 Optimal 算法在节点规模稍大时不能保证实时性).在每一种节点规模下,我们都运行 100 组实验,每组实验包括 1 次 LD/RPath 算法和 20 次 Random 算法,Random 算法结果取 Random-Best,对比结果如图 6 所示.

图 6(a)表示在 6 种不同的服务网络规模下,100 组实验中 LD/RPath 算法与 Random 算法相比产生的路径的时延较小的次数.我们看到,在 6 种情况下,LD/RPath 算法产生的路径的时延均有 90 多次,比 Random-Best 算法要小.我们也观察到,随着网络规模的增大,LD/RPath 算法的性能略微下降,在网络规模为 30 的时候,LD/RPath 算法产生的路径的时延有 94 次,比 Random-Best 算法要小;当网络规模达 1 600 时,LD/RPath 算法产生的路径的时延比 Random-Best 算法小的次数下降到 82 次.图 6(b)和图 6(c)分别是两者的可靠性和时延/可靠比的比较.与图 6(a)的情况类似,我们看到 LD/RPath 算法的选择效果均远远要好于 Random 算法.

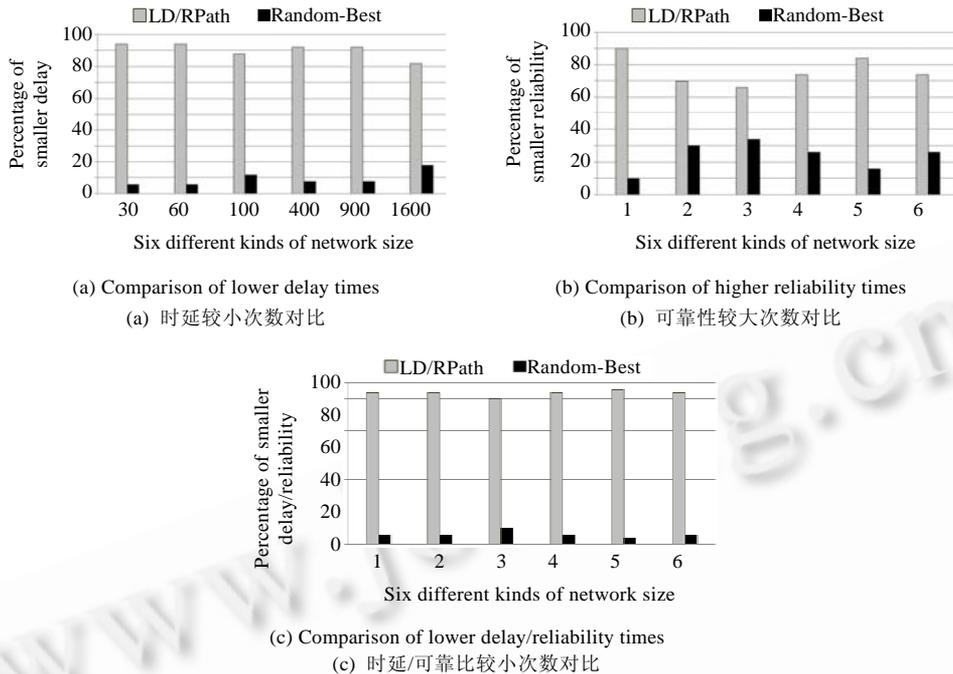


Fig.6 Comparison of algorithms LD/RPath and Random-Best at different network sizes

图6 LD/RPath 算法和 Random-Best 算法在不同网络规模下路径选择效果对比

综合运行效率和选择效果两方面的实验结果可以看出,LD/RPath 算法在网络拓扑规模较小时选择效果接近 Optimal 算法,同时运行时间几乎与 Random 算法相同.当网络拓扑规模逐渐变大时,LD/RPath 算法选择效果远远要好于 Random 算法,而 Optimal 算法则逐渐不能保证实时性从而变得不适用.综合来看,LD/RPath 算法以较小的时间复杂度获得了较好的路径选择效果,较适合于面向服务的实时多媒体传输.

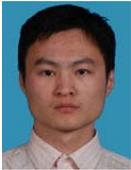
#### 4 总 结

在开放的普适环境中,如何构建高可靠软件系统是一个重点研究问题.本文研究在面向服务的实时多媒体传输系统中如何选择具有低时延和高可靠性的路径的问题,提出了一种较有效的近似算法 LD/RPath.该算法的创新之处在于:(1) 把可靠性合理地融入到边权值之中,使得算法在选择最短路径的同时考虑了可靠性;(2) 数据量的合理近似很好地降低了问题的复杂度;(3) 节点分裂使节点权值统一至边权值之中.仿真实验结果表明,LD/RPath 算法是一种高效、可靠的算法,以较小的时间复杂度获得了较好的路径选择效果.然而,LD/RPath 算法还有很大的改进空间,我们准备在负载均衡、实时网络状态信息更新等方面做进一步的研究工作.

#### References:

- [1] Nahrstedt K, Balke WT. Towards building large scale multimedia systems and applications: Challenges and status. In: Proc. of the 1st ACM Int'l Workshop on Multimedia Service Composition. New York: ACM Press, 2005. 3-10.
- [2] Benatallah B, Dumas M, Sheng QZ, Ngu A. Declarative composition and peer-to-peer provisioning of dynamic Web services. In: Proc. of the ICDE 2002. San Jose: IEEE Computer Society Press, 2002. 297-308.
- [3] Qian ZZ, Guo MY, Zhang S, Lu SL. Service-Oriented multimedia delivery in pervasive space. In: Proc. of the Wireless Communications and Networking Conf. (WCNC 2009). IEEE Computer Society Press, 2009. 1-6.
- [4] Zeng L, Benatallah B, Dumas M, Kalagnanam J, Sheng QZ. Quality driven Web services composition. In: Proc. of the 12th Int'l Conf. on World Wide Web (WWW 2003). New York: ACM Press, 2003. 411-421.

- [5] Gu XH, Nahrstedt K, Yu B. SpiderNet: An integrated peer-to-peer service composition framework. In: Proc. of the 13th IEEE Int'l Symp. on High Performance Distributed Computing (HPDC-13 2004). 2004. 110–119.
- [6] Gu XH, Nahrstedt K. Distributed multimedia service composition with statistical QoS assurances. IEEE Trans. on Multimedia, 2006,8(1):141–151.
- [7] Raman B, Katz RH. Load balancing and stability issues in algorithms for service composition. In: Proc. of the 22nd Annual Joint Conf. of the IEEE Computer and Communications Societies (INFOCOM 2003). IEEE Computer Society Press, 2003. 1477–1487.
- [8] Raman B, Katz RH. Emulation-Based evaluation of an architecture for wide-area service composition. In: Proc. of Int'l Symp. on Performance Evaluation of Computer and Telecommunication Systems (SPECTS 2002). 2002.
- [9] Kalasapur S, Kumar M, Shirazi B. Dynamic service composition in pervasive computing. IEEE Trans. on Parallel and Distributed Systems, 2007,18(7):907–918.
- [10] Wang Z, Crowcroft J. Quality-of-Service routing for supporting multimedia applications. IEEE Journal on Selected Areas in Communications, 1996,14(7):1228–1234.
- [11] Cormen TH, Leiserson CE, Rivest RL, Stein C. Introduction to Algorithms. Cambridge/New York: MIT Press/McGraw-Hill, 2003.
- [12] Choi S, Turner J, Wolf T. Configuring sessions in programmable networks. In: Proc. of the 20th Annual Joint Conf. of the IEEE Computer and Communications Societies (INFOCOM 2001). IEEE Computer Society Press, 2001. 60–66.
- [13] Korkmaz T, Krunz M, Tragoudas S. An efficient algorithm for finding a path subject to two additive constraint. In: Proc. of the Int'l Conf. on Measurement and Modeling of Computer Systems (SIGMETRICS 2000). New York: ACM Press, 2000. 318–327.
- [14] Ma QM, Steenkiste P. On path selection for traffic with bandwidth guarantees. In: Proc. of the '97 Int'l Conf. on Network Protocols. 1997. 191–202.



张胜(1986—),男,江苏句容人,硕士生,主要研究领域为服务组合,无线网络,分布式计算.



陆桑璐(1970—),女,博士,教授,博士生导师,CCF 高级会员,主要研究领域为分布并行处理,高性能计算.



钱柱中(1980—),男,博士,讲师,CCF 会员,主要研究领域为面向服务的计算,分布式系统.