一种集中调控的分布式服务路径选择算法

李 丹* 兰巨龙 王 鹏 胡宇翔
(国家数字交换系统工程技术研究中心 郑州 450000)

摘 要：在可重构网络多态路由模型中，通常存在多条满足业务需求的服务路径。针对最优服务路径选择问题，该文设计了一种集中调控的分布式服务路径选择算法，各节点根据服务请求中的第 1 个元能力和目的节点生成路由表，控制器实时监控网络，调控代价过高路径并平衡网络的带宽和负载。性能分析和仿真结果表明，分布式路由表能够生成有效的服务路径，表项规模、收敛时间与元能力个数成正比，在 30%的集中调控比例下，路径代价和负载均衡度性能良好，与其他算法相比，对服务请求的响应时延降低约 50%。

关键词：可重构网络；多态路由；服务路径选择；分布式路由表；集中调控
中图分类号： TP393 **文献标识码**： A **文章编号**：1009-5896(2018)04-0785-09
DOI: 10.11999/JEIT170600

Distributed Service Path Selection Algorithm under Central Control

LI Dan LAN Julong WANG Peng HU Yuxiang
(*National Digital Switching System Engineering & Technology Research Center*, *Zhengzhou* 450000, *China*)

Abstract: There are many service paths which can satisfy the business requirements in polymorphic routing model of reconfigurable networks. For the issue of the best service path selection, this paper proposes a distributed service path selection algorithm under central control. Each node generates routing tables based on the first function and destination node in service request. The controller monitors the network in real time, regulates paths with high costs and balances the bandwidths and loads of the network. Performance analysis and simulation results show that, the distributed routing tables can generate efficient service paths and the convergence time is proportional to the number of functions. When the proportion of central control is 30 percent, the algorithm has a good performance on average cost of paths and load balance. The response delay to service request decreases almost 50 percent compared with other algorithms.

Key words: Reconfigurable networks; Polymorphic routing; Service path selection; Distributed routing table; Central control

1 引言

随着互联网的高速发展和广泛应用，简单的存储转发功能已经满足不了新兴应用的需要，不断创新的通信模式、不断变化的应用环境以及不断提高的安全需求对网络中的新功能和新服务提出了更高的要求。然而由于网络体系的僵化，当前的互联网络无法支持大量新的功能和服务，为弥补这些不足，中间盒[1](middlebox)作为一种新的技术引入到网络的数据传输路径中，在一定程度上扩展了网络功能。随后又出现了覆盖网[2](overlay networks)、P2P 网络[3]、面向服务架构[4](Service Oriented Architecture, SOA)等。这些研究工作和成果大大丰富和增强了网络服务能力，但是仅仅对现有网络端系统改造而不改变核心网络的方法对服务能力的提升仍然有限。

面对多样化的业务和通信模式需求，以可重构基础网络[5]为代表的以服务为中心的新型网络应运而生。在可重构网络多态路由[6]模型中，把网络定制服务分解为数量有限的网络功能单元 —— 元能力(如防火墙、入侵检测、服务代理、网络地址转换等)，特定的服务由按顺序排列的一系列元能力组成，且次序不能发生改变。每种元能力可以对应多个具体的元能力实例，根据需要部署于网络的任意节点中，依次连接元能力实例和目的节点的路径称为服务路径，网络控制器需要在所有满足要求的服务路径中选择一个总体代价最小的路径，这一个问题称为最

收稿日期：2017-06-23；改回日期：2017-11-27；网络出版：2018-01-23
*通信作者：李丹 pkulidan@foxmail.com
基金项目：国家 863 计划项目(2015AA016102)，国家自然科学基金(61521003)
Foundation Items: The National 863 Program of China (2015AA016102), The National Natural Science Foundation of China (61521003)

优服务路径选择问题。

与传统网络路由问题不同，在最优服务路径选择过程中，除了目的地址之外，还需要依次经过每个元能力对应的节点，目前大部分研究采用了集中式的控制方式来计算路径。文献[7]将服务路径选择问题约减为一个经典的多约束最优路径问题，提出了一种建立确保 QoS 的端到端跨域通信路径的自动服务组合框架。文献[8]给出了一种网络服务组合算法，通过马尔科夫思想迭代计算最优的服务组合。文献[9]研究了时延和其它 QoS 指标之间的关系，并基于自适应的遗传算法实现了时延和其它 QoS 指标之间的最优化平衡。文献[10]从人工智能和组合优化的角度研究服务组合与路径选择问题，但是算法在服务较多时复杂度较高。文献[11]将网络中的节点分类，分析不同节点在不同时期对各类服务的需求程度，从而动态分配服务节点。文献[12]设计了一种分层服务图算法，一定程度上降低了服务路径的求解复杂度，但是该算法容易引起节点过载。文献[13]分别针对小型网络和大型网络设计了线性规划算法和启发式算法，同时优化路径代价和节点负载。

集中式的控制方式固然可以得到较为精确的计算结果，但是频繁的路径计算负担较大，表项的下发需要大量的控制信息，网络的鲁棒性和扩展性较差。文献[14]和文献[15]选择分布式的控制方式，定义了各自的服务路径代价度量指标，并基于最短路径优先算法计算最优路径。文献[16]和文献[17]在分布式的控制方式基础上增加了集中调控机制，各节点按照 OSPF 协议构建缺省路由，控制器修改相关路由表来调节路径。文献[18~20]提出了一种面向服务的 SpiderNet 框架，部署功能与服务请求相匹配的节点发送服务提供请求，由网络汇聚结点集中比对各服务提供请求并选择服务路径，但是需要大量的网络信息交互进行路径搜索，计算速度较慢。在 SpiderNet 基础上，文献[21]提出了一种可以动态优化 QoS 和负载均衡的算法(QALB)，该算法引入了动态加权代价函数，灵活应对不同的网络需求，在新服务出现时通过蚁群算法进行求解，计算速度更快。

基于以上分析，本文提出一种以分布式控制为主，集中式控制为辅的服务路径(Central-Distributed Service Path, CDSP)算法，每个节点需保存分布和集中两类路由表，路由表中除目的节点外还包含网络中的各类元能力。节点间通过分布式路由算法计算缺省路径，即时处理各类服务请求，当某一条路径需要改动时，由网络控制器进行路径修正，并通知相关节点添加集中路由表项。该算法尽管牺牲了一定的路径代价，但是具有数据包即时处理转发、管控网络负载均衡、快速应对网络结构变化等优点。

2 集中调控的分布式服务路径算法

在可重构网络多态路由模型中，S_a 代表一个服务请求，$S_a=\{S_1,S_2,\cdots,S_n,D\}$，其中 n 表示所分解的元能力数量，S_i 表示 S_a 所分解的第 i 个元能力，$i=1,2,\cdots,n$，D 表示完成服务后数据的发送目的。假设每一个元能力 S_i 所对应的候选元能力实例集为 $\left\{S_i^1,S_i^2,\cdots,S_i^{M_i}\right\}$，$M_i$ 为候选的元能力实例的个数，用 $S_i^{j_i}$ 表示元能力 S_i 对应的第 j_i 个元能力实例。因此，多态路由模型需要分别为每一个元能力选择一个元能力实例，生成一个组合服务响应 $S_b=\left\{S_1^{j_1},S_2^{j_2},\cdots,S_n^{j_n},D\right\}$ 来满足服务请求 S_a。例如，在图 1 中，当源节点收到一条内容为 $S_a=\{S_1,S_2,\cdots,S_4,D\}$ 的服务请求时，可以选择实线表示的服务路径 $S_{b_1}=\left\{S_1^1,S_2^2,S_3^4,S_4^1,D\right\}$，也可以选择虚线表示的服务路径 $S_{b_2}=\left\{S_1^3,S_2^1,S_3^2,S_4^3,D\right\}$，在满足服务请求的同时，应该尽可能在所有符合要求的路径中选择一条代价最小的路径，来降低整个网络的负载，即最优服务路径选择问题。

本文提出的路由算法以分布式控制为主，集中式调节为辅，每个节点分别储存一张分布路由表和一张集中路由表，分布路由表只包含目的节点和服务请求中的第 1 个元能力，而集中路由表包含目的节点和服务请求中的全部元能力，两张表相互独立且集中路由表优先查询。系统结构如图 2 所示，各节点根据相互通告的路由信息建立分布路由表并上传至控制器，由控制器实时监控缺省路径，当路径因为代价过大或负载不均衡等原因需要更改时，控制器计算最优路径，并下发控制信息到路径上的相关节点，指挥其添加集中路由表项，在表项匹配时按照该路由表修正转发目标。该算法具有以下两大

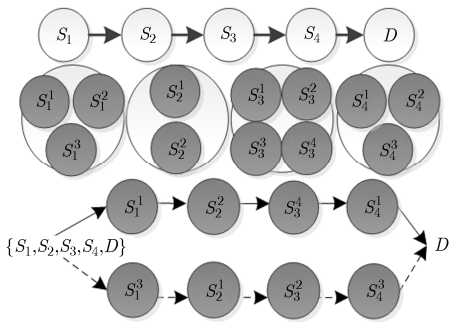

图 1 服务路径选择示意图

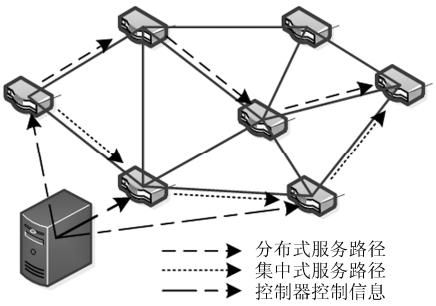

图 2 服务路径算法结构图

优点：

(1)分布式算法只需考虑目的节点和服务请求中的第 1 个元能力，既能按照服务请求的顺序依次经过提供各项元能力实例的节点和目的节点，又能大幅降低路由维护的复杂度。通过合理的参数设计，该算法生成路径和最优路径相比的平均代价误差不大，即使不引入集中调控仍然可以满足大多数服务需要；

(2)集中式算法考虑整个服务请求，可以得到最优路径来替换分布式路径中代价较大的路径，进一步降低网络的平均路径代价。除路径代价外，集中式算法还可以综合考虑网络的负载均衡，链路的带宽限制等。

2.1 分布式路由算法

2.1.1 路由表结构　分布式路由算法需要各节点根据邻居节点间的交互信息建立路由表，与传统地址寻址只需要考虑目的节点不同，服务路径需要依次匹配服务请求中的每一项元能力和目的节点，才能得到准确的最优路径。但是，如果将所有可能的服务请求都添加到路由表中，随着服务请求队列长度的增加，路由表的规模会近似指数级增加，而且路由的生成和维护复杂度极高，需要路由节点具备很强的计算和存储能力，另外路由算法的收敛时间也会很长，容易造成路由回路或数据丢失。考虑到以上问题，在本算法中路由表只包含目的节点和服务请求中的第 1 个元能力 —— 首项元能力，每当一个元能力完成实例化，就将其从服务请求中删除，由下一个元能力递补为首项元能力。路由表的规模和网络中的服务种类数成正比，虽然得到的服务路径很可能不是最优的，但是通过合理的参数设计，其平均路径代价仅仅稍高于最优路径。

2.1.2 路径代价计算　分布式路由算法的核心是路径代价计算，最优服务路径选择问题要求从所有可能的服务路径中选择满足多约束的多目标最优的一条服务路径。多约束包括带宽约束、负载约束、功能相似度约束等等，这些约束由集中式调控算法实现；多目标主要指时延最小、费用最小、功能可用性最大以及节点可靠性最大，这部分则通过代价函数体现。其中时延(De)表示相邻节点间的传输时延，费用(Co)表示下一跳节点能够提供的服务所需开销，功能可用性(Av)表示下一跳节点成功提供相应服务的概率，节点可靠性(Re)表示下一跳节点保持有效的概率。考虑到时延和费用与代价正相关，功能可用性和节点可靠性与代价负相关，为了同时优化以上 4 个目标，本算法参考文献[22]给出的方式，将链路 g 的代价表示如式(1)：

$$Z(g)=\frac{w_1\mathrm{De}(g)+w_2\mathrm{Co}(g)}{w_3\mathrm{Av}(g)+w_4\mathrm{Re}(g)} \tag{1}$$

与经典的最短路径算法(Dijkstra)类似，节点根据链路代价，选择服务路径代价最小的表项填入分布路由表。在本文算法中路径代价的计算分为两类，第 1 类是无服务路径代价，与最短路径算法相同，令 AD 表示节点 A 到节点 D 的所有可能路径，无服务路径代价按式(2)计算，即源节点到目的节点的最短路径。

$$Z\left(A0D\right)=\min Z(AD) \tag{2}$$

第 2 类是有服务路径代价，理论上，源节点经过首项元能力 S_i 到目的节点的服务路径代价应表示为

$$\begin{aligned}Z\left(AS_iD\right)=&\min Z\left(AS_i\right)+\min\Big(\min Z\left(S_iS_{i+1}\right)\\&+\cdots\min\Big(\min Z\left(S_{i+n-1}S_{i+n}\right)\\&+\min Z\left(S_{i+n}D\right)\Big)\Big)\end{aligned} \tag{3}$$

该函数由两部分组成，分别是源节点到首项元能力节点的最短路径，以及首项元能力节点经过其他元能力节点到目的节点的最短路径，根据分布路由表表项组成，本文算法用 $\min Z\left(S_iD\right)$ 替换式(3)的第 2 部分，为了使等号成立，两部分需分配不同的权值，如式(4)所示，其中 $\alpha>0,\beta>0$ 。

$$Z\left(AS_iD\right)=\alpha\min Z\left(AS_i\right)+\beta\min Z\left(S_iD\right) \tag{4}$$

各节点通过比较式(2)和式(4)得到的代价分别生成无服务和有服务路由表。最优权值的确定与服务请求队列中的元能力数以及可选择的元能力实例个数相关，在下一小节详细说明。

2.1.3 最优权值比推导　权值是否合理决定了算法效率的高低，可以通过两个参数逐步推导 α 和 β 的关系，分别是可选择的元能力实例个数和服务请求队列中的剩余元能力个数。

首先分析可选择的元能力实例个数，考虑服务请求中剩余元能力数为 2 的情况。假设网络中任意两个节点间的最短路径代价不超过 d ，即

$$0 \le \min Z(S_iD), \min Z(S_iS_{i+1}), \min Z(S_{i+1}D) \le d \quad (5)$$

根据三角不定式可得

$$\min Z(S_iD) \le \min\left(\min Z(S_iS_{i+1}) + \min Z(S_{i+1}D)\right) \le 2d \quad (6)$$

令 p 为提供元能力 S_{i+1} 的节点个数，h 为节点总数，其中有 h_1 个节点在 S_iD 路径上，为了便于计算，假设其他 $h-h_1$ 个节点平均分布在到 S_i 和 D 的路径代价之和为 $\min Z(S_iD) \sim 2d$ 之间的 2 维平面上，p 个元能力实例在所有节点中随机选取。用 $f(2)$ 表示剩余 2 个元能力时对 $\min\left(\min Z(S_iS_{i+1}) + \min Z(S_{i+1}D)\right)$ 的估计，则估计代价为

$$f(2) \approx \left(\frac{h-h_1}{h}\right)^p \left(\frac{4d - \min Z(S_iD)}{\sqrt{p+1}} + \min Z(S_iD)\right) + \left(1 - \left(\frac{h-h_1}{h}\right)^p\right) \min Z(S_iD) \quad (7)$$

接下来分析剩余元能力个数，当剩余元能力个数为 1 时，$\min Z(S_iD)$ 也是确定值，只需要令 $\alpha = \beta$ 即可得到最优代价函数。当剩余元能力个数为 n 时，用 $f(n)$ 表示相应的估计。假设节点的分布和元能力实例的部署都是随机的，每次引入一个元能力带来的误差为前一次的 q 倍，可以推算出：

$$f(n) \approx \left(\frac{h-h_1}{h}\right)^p \left(\frac{4d - \min Z(S_iD)}{\sqrt{p+1}}\right) \sum_{i=1}^{n-1} (q)^{i-1} + \min Z(S_iD) \quad (8)$$

得到 $f(n)$ 的估计后，可以进一步推导 α 和 β 的参数选取了，把 $f(n)$ 代入式(3)，得

$$Z(AS_iD) \approx \min Z(AS_i) + \left(1 - \left(\frac{h-h_1}{h}\right)^p \cdot \frac{1}{\sqrt{p+1}} \sum_{i=1}^{n-1} (q)^{i-1}\right) \min Z(S_iD) + \left(\frac{h-h_1}{h}\right)^p \frac{4d}{\sqrt{p+1}} \sum_{i=1}^{n-1} (q)^{i-1} \quad (9)$$

节点对各服务路径代价进行比较确定路由表项，式(9)中的最后一项为常量，在比较时消去不计，因此最优权值应满足：

$$\frac{\beta}{\alpha} = 1 - \left(\frac{h-h_1}{h}\right)^p \frac{1}{\sqrt{p+1}} \sum_{i=1}^{n-1} (q)^{i-1} \quad (10)$$

利用 β/α 的非负性推导 q 的取值，当 $n \to \infty$ 时，$\beta/\alpha \to 0$，即

$$\lim_{n\to\infty} \left(1 - \left(\frac{h-h_1}{h}\right)^p \frac{1}{\sqrt{p+1}} \sum_{i=1}^{n-1} (q)^{i-1}\right) = 0 \quad (11)$$

求解式(11)并代入式(10)，得到最优权值的最终表达式。

$$\frac{\beta}{\alpha} = 1 - \left(\frac{h-h_1}{h}\right)^p \frac{1}{\sqrt{p+1}} \cdot \sum_{i=1}^{n-1} \left(1 - \frac{1}{\sqrt{p+1}} \left(\frac{h-h_1}{h}\right)^p\right)^{i-1} \quad (12)$$

各节点按照式(12)计算，可以得到相应参数下的最优权值关系，为了便于计算，可以直接设定 $\alpha = 1$。在一个稳定的网络中，参数 h，h_1，p 都是确定值，最优权值主要由服务请求中的元能力数 n 来决定，系统应依据网络中的平均服务队列长度 $\bar{n}$ 来确定 β/α 值，从而优化网络中服务路径的平均代价。

2.1.4 路由更新和查询 确定了路径代价的计算法则后，要设计邻居节点间的路由共享和更新策略。每个节点需要周期性地向邻居节点共享自己的路由信息，收到共享信息的节点更新路径代价和路由表。和路径代价一样，路由信息同样分成两类，第 1 类是无服务路由信息：节点收到 $(0, D)$ 路由信息后，在原路径代价基础上加上两个节点间的链路代价，更新为本节点的 $(0, D)$ 路径代价；如果该节点可以提供某种元能力 S_i，则在原路径代价基础上加上两个节点间的链路代价，并乘上权值 β，更新为本节点的 (S_i, D) 路径代价。第 2 类是有服务路由：节点收到 (S_i, D) 路由信息后，在原路径代价基础上加上两个节点间链路代价的 α 倍，更新为本节点的 (S_i, D) 路径代价。表 1 给出了节点 B 收到节点 A 共享的路由信息后的具体处理方式，其中节点 B 可以提供元能力 S_1，节点 B 更新路径代价后再和其他相同表项的路径代价作比较，选择代价最低的路径填入路由表。

路由表的生成和维护策略确定后，节点就可以根据服务请求和目的节点来传输和处理数据了。节点接收到一个数据后，首先查看服务请求队列是否为空，若为空则根据目的节点查找无服务路由表并转发数据。对于有服务要求的数据，根据首项元能力和目的节点查询下一跳节点，若下一跳节点为该节点本身，则由该节点提供相应元能力，并在服务队列中删除首项元能力，重新查看队列是否为空；若不是，则把数据和相关信息发送给下一跳节点。以此类推，直至数据发送至目的节点且服务队列为

表 1 节点 B 处理节点 A 共享信息

路由信息	首项元能力	目标节点	路径代价
$A0F$	0	F	$A0F + AB$
	S_1	F	$\beta\,(A0F + AB)$
AS_1F	S_1	F	$AS_1F + \alpha\,AB$
AS_2F	S_2	F	$AS_2F + \alpha\,AB$

空，完成服务路径的分布式实现。图 3 表示节点 A 收到 $\{S_1,S_2,F\}$ 服务请求时的查表转发过程，其中节点 B 和节点 C 由于分别提供了元能力 S_1 和 S_2，需要进行两次查表，最后数据发送至节点 F 且服务队列为空，服务请求完成。

2.1.5 算法说明和复杂度分析　分布式服务路径路由算法的伪代码如表 2 的算法 1 所示。

分布式路由算法的复杂度指路由收敛时间，即从检测到网络拓扑发生变化开始，到所有节点的路由表更新结束的时间，是分布式路由算法的重要指标。如果收敛时间过大，容易造成路由回路，数据丢失等情况，严重影响网络性能。

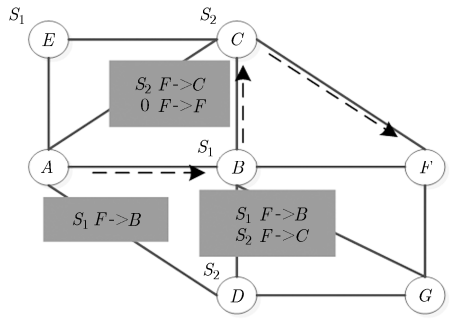

图 3　节点 A 收到 $\{S_1,S_2,F\}$ 服务请求分布式路由示意

对于一个包括 V 个路由节点、n 个元能力的网络，在经典的最短路径算法(OSPF/IS-IS)中，路由表规模为 V，路由收敛时间和路由表项数成正比，即 $T_1=O_1(V)$ [23]，本文算法可以看做在最短路径路由表的基础上增加了 n 个有服务的最短路径路由表，所以路由收敛时间可以表示为 $T_2=O(V)+O(nV)$。由于网络中的元能力种类有限，因此路由表规模和算法复杂度可以满足大多数网络要求。

2.2 控制器集中调控

2.2.1 集中调控机制　通过上面的分布式路由算法，各节点可以协作实现路由表的建立与维护和数据的处理与转发，但是由于该算法得到的不是全局最优解，生成的服务路径可能代价较大，另外网络中经常需要考虑负载均衡和带宽限制等原因调整路径，

表 2　分布式服务路径路由算法的伪代码

算法 1　分布式路由算法

(0) The following procedure runs on each individual node independently.
(1) for each packet arrived
(2)　　if it is routing packet　//接收到路由共享信息
(3)　　set newrout←routing message, linkcost←cost between two nodes
(4)　　if there is no service in the message　//无服务路径信息
(5)　　oldrout with no service ←min(newrout+linkcost, oldrout with no service)
(6)　　if this node can offer service s//可以提供服务的节点修正服务路径信息
(7)　　oldrout with s ←min((newrout+linkcost)* β , oldrout with s)
(8)　　end if
(9)　　else　//服务路径信息
(10)　　oldrout with s ←min(newrout+linkcost* α , oldrout with s)
(11)　　end if
(12)　　send changed oldrout to other neighbor node and controller
(13)　　else //接收到数据信息
(14)　　set s ←the first service, d ←destination
(15)　　if s is null and d is this node //服务请求为空且抵达目的节点，结束传输
(16)　　keep the data
(17)　　else
(18)　　nextskip←find (s,d) in oldrout //通过路由表查找下一跳
(19)　　if nextskip is this node //下一跳为该节点，提供首项元能力，更新服务请求，重新查询
(20)　　this node offer service s
(21)　　delete the first service in service queue
(22)　　goto (14)
(23)　　end if //下一跳不是该节点，将数据传输给下一跳节点。
(24)　　send service message to nextskip
(25)　　end if
(26)　　end if
(27)　end for

这时需要集中式的调控机制发挥作用。

集中调控由网络中的控制器完成，控制器实时获取网络拓扑结构和带宽负载等信息，并监控各节点分布路由表生成的缺省路径，当路径需要修正时，控制器根据当前的网络信息结合带宽负载等限制条件计算最优服务路径，并将计算结果以集中路由表项的形式下发给需要修正路径的节点。每个节点把接收到的路由表项添加到集中路由表中，当数据包到达时，节点首先查看集中路由表中是否存在匹配项，如果有匹配项则按照该表项执行操作，否则继续查看分布路由表。集中路由表与分布路由表互相独立，集中路由表项的增添完全由控制器指挥完成，邻居节点间只共享分布路由信息。集中路由表的查表项包括服务队列中的所有元能力以及目的节点，因此一条集中路由表项只能修正一条服务请求，不会影响其他分布式路径。

图 4 表示路由表修正后节点 A 接收到 $\{S_1,S_2,F\}$ 服务请求时的查表转发过程，在图 3 中节点 A 的分布路由表下一跳为节点 B，而经过修正后该服务请求的下一跳节点根据集中路由表修正为节点 E。另外，如果节点 E 和节点 C 的分布式路径不符合服务器计算的最优路径，服务器也需要向这两个节点下发路由表项。

2.2.2 最优路径计算方法 已经有大量文献给出了最优路径的集中式计算方法，由于本文算法需要计算次数较少，不需要考虑计算时延和代价，应选择计算精度最高的遍历法，即穷举所有满足服务请求和带宽负载限制的路径，选择其中代价最小的为最优路径。具体的算法伪代码如表 3 的算法 2 所示。

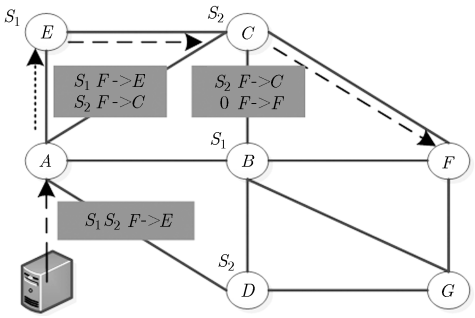

图 4 集中调控后节点 A 收到 $\{S_1,S_2,F\}$ 服务请求路由示意

2.2.3 控制触发机制与时延分析 集中控制的引入有两个目的，一是为了调节网络的带宽和负载，需要综合考虑除路径代价外的各项网络参数；二是为了避免分布式路径传输代价过大，当服务请求出现时，控制器要实时判断分布式路径的传输代价是否合理，以决定是否进行集中控制。

判断分布式路径代价有 3 种方法，分别是计算法，估计法和统计法，由于计算法会大幅增加控制器的负担，因此在系统运行初期应该主要依赖估计法判断代价，运行一段时间后可以采用统计法判断路径代价。

估计法的推导方式与最优权值的分析过程类似，参照 2.1.3 节，服务队列长度为 n 的路径代价估计为

$$Z(AS_1\cdots S_nD)=\left(\frac{h-h_1}{h}\right)^p\left(\frac{2d-Z(AD)}{\sqrt{p+1}}\right)\cdot\sum_{i=1}^{n-1}\left(1-\frac{1}{\sqrt{p+1}}\left(\frac{h-h_1}{h}\right)^p\right)^{i-1}+Z(AD) \tag{13}$$

表 3 最优路径算法伪代码

算法 2 集中式路由算法

(0) The following procedure runs on controller.
(1) for each service message arrived
(2) set a ←source, d ←destination, as ← Dijkstra(a,d), sd ←0, lastsnode ←d
(3) for service queue is not empty//迭代计算引入每一个元能力的最优路径
(4) set s ←the last service, snode ←nodes that can offer s, ss ←Dijkstra(snode, lastsnode)
(5) if $ss(i,j)$ do not satisfy constraint conditions//放弃不满足约束条件的路径
(6) $ss(i,j)$ ←infinite
(7) end if
(8) sd ←min (sd+ss)
(9) as ←Dijkstra(snode, a)
(10) delete the last service in service queue
(11) lastsnode ← snode
(12) end for
(13) ad=min(as+sd)//所有元能力引入后得到最优路径并下发路由信息。
(14) send routing message to the nodes along ad if necessary
(15) end for

综上所述，集中控制的触发步骤和时延如下：第 1 步，当服务请求出现时，源节点将服务请求信息上传给服务器，时延为 T_1；第 2 步，控制器采用估计法或统计法得到路径代价参考值并设定判决门限，依次查看各节点上传的分布式路由表，得到分布式路径经过的链路带宽和节点负载，并将各链路代价相加得到路径代价，时延为 T_2；第 3 步，分别根据带宽、负载和路径代价判断是否需要集中控制，如果不需要调控，直接通知源节点处理该服务请求，时延为 T_3；否则，控制器按照算法 2 计算最优路径，将集中式路由表项下发给路径上的节点，时延为 T_4。其中 T_1, T_2 和 T_3 时延很短，在小型网络中仅为毫秒级；T_4 时延较长，且与网络规模、服务队列长度直接相关，但是由于集中控制触发比例较低，网络的平均响应时延很短。

3　性能仿真与分析

3.1 仿真设计

为了对本文提出的服务路径算法性能进行仿真验证，设计了如下仿真场景：随机网络拓扑按照 Waxman-Salam 模型[24]生成，网络的平均连接度为 4，可保证网络的连通性，任意相邻节点间的链路代价随机生成。网络节点数(h)为 60，网络中可提供的元能力数(n)为 2~10，每种元能力的可选择实例数(p)为 10，各元能力实例随机分布在所有网络节点中，每个节点最多可同时处理 10 个服务请求。

在仿真过程中，网络中任意节点可以产生服务请求 $S_a=\{S_1,S_2,\cdots,S_n,D\}$，其中 $n\le 10$ 且 $S_1,S_2,\cdots,S_n$ 互不相同，作为参照的最优路径由遍历算法生成，来评估本算法的有效性。为了提高算法评估的精确性，采用蒙特卡洛方法，在每个测试场景下都进行了 100 组仿真测试，取 100 组仿真测试的平均值作为测试结果。

3.2 分布式算法性能验证

基于以上仿真场景，对分布式路由算法的性能进行了仿真。路径代价是评价算法性能最主要的指标，也是判断代价函数权值是否合理的标准。实验首先仿真了设定不同权值时的路径代价和最优路径的相对误差，分别仿真了在元能力种类为 2, 6, 10 的情况下代价函数的权值对算法性能的影响。

从图 5 中可以看出，元能力种类数与最佳权值比相关。表 4 给出了在上述条件下不同元能力种类数对应的理论最佳权值和实验最佳权值，结果表明，由式(13)得到的理论值与实验仿真得到的最佳权值接近，结合图 5 可以看出，即使理论值稍稍远离最佳权值范围(如 $n=10$)，相对误差也没有明显的提升。

表 4 理论最佳权值和实验最佳权值

元能力种类	理论权值	实际权值
2	1.49	1.4~1.6
4	2.16	2.2~2.4
6	3.03	3.0~3.2
8	4.11	3.8~4.0
10	5.30	4.4~4.6

图 6 表示在元能力种类为 10 的网络中，进行 100 次仿真得到的分布式路径与最优路径代价相对误差分布直方图。从图 6 中可以看到，多数情况下相对误差小于 0.3，如果把集中控制的代价触发门限设置为最优路径的 1.3 倍，可以严格控制网络中的服务路径代价，调控比例仅为 30%，如果网络中的服务对路径代价要求不高，可以把门限设置为 1.4 倍，调控比例降到 10%，进一步降低控制器的工作负担。

3.3 相关算法性能比较

为了进一步验证引入集中调控后算法在服务请求路径代价、负载均衡度和响应时延等方面的性能，将本算法(CDSP)与遍历算法、SpiderNet 和 QALB 算法进行了对比，其中本文算法的调控门限分别设置为最优路径的 1.3 和 1.4 倍，调控比例约为 30% 和 10%。请求响应时延的仿真结果如图 7 所示，可以看出，本文算法对服务请求的响应时延远远小

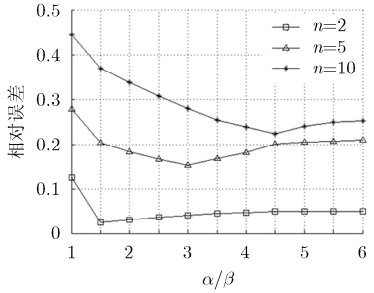

图 5 元能力种类为 2/6/10 条件下不同权值与相对误差的关系

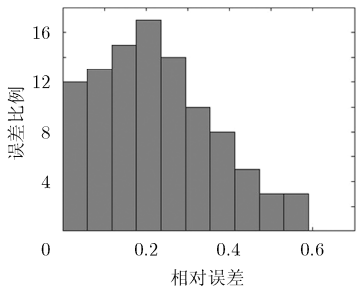

图 6 相对误差分布

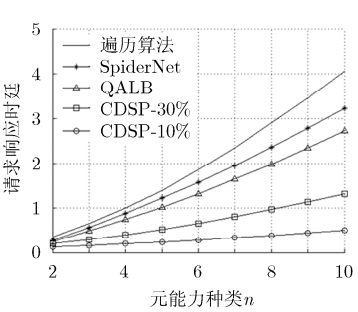

图 7 各算法服务请求响应时延

于其他算法，在 30%调控比例下，与响应时延最小的 QALB 算法相比减少约 50%的计算时间，如果将调控比例降低到 10%，响应时延进一步缩短，可以实现对数据包的快速处理转发。

图 8 和图 9 分别给出了服务路径代价和负载均衡度的仿真结果，其中负载均衡度的对比指标为全网节点中处理服务请求数量的最大值 $\max(\text{load})$，该指标越大则负载均衡性越差。结果表明，在 30%调控比例下，本文算法服务路径代价稍高于 QALB 算法和遍历算法，与 SpiderNet 算法基本持平，负载均衡性能则介于 SpiderNet 和 QALB 算法之间；调控比例为 10%时，路径代价和负载均衡度性能均比其他算法稍差。

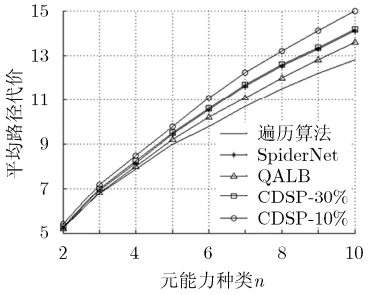

图 8 各算法平均路径代价

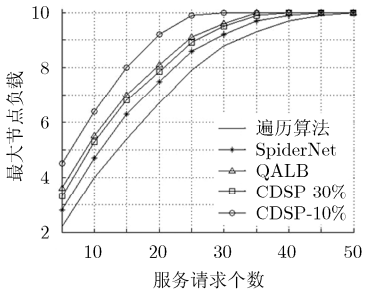

图 9 各算法负载均衡性能

4 结束语

本文针对可重构网络中的服务路径选择问题，提出了一种以分布式控制为主、集中式控制为辅的混合式架构，并分别设计了分布式路由算法和集中调控机制。分布式算法构建缺省路由，保证服务路径有效；集中调控机制计算最优路由，调控代价过大路径。与现有集中式算法相比，本文算法在满足服务属性要求的同时，服务请求的响应时延大幅缩短而路径代价和负载均衡度没有明显降低，分布式的缺省路由具有更强的鲁棒性和扩展性，对于可重构网络多态路由的稳定和扩展具有重要意义。

参 考 文 献

[1] BERNSTEIN P A. Middleware: A model for distributed system services[J]. *Communications of the ACM*, 1996, 39(2): 86–98. doi:10.1145/230798.230809

[2] CHU Y H, RAO S G, SESHAN S, *et al*. A case for end system multicast[J]. *IEEE Journal on Selected Areas in Communications*, 2002, 20(8): 1456–1471. doi: 10.1109/JSAC.2002.803066.

[3] GOLLE P, LEYTON-BROWN K, MIRONOV I, *et al*. Incentives for Sharing in Peer-to-Peer Networks[M]. Electronic Commerce. Springer Berlin Heidelberg, 2001: 75–87. doi: 10.1007/3-540-45598-1_9

[4] PERREY R and LYCETT M. Service-oriented architecture [C]. 2003 IEEE Symposium on Applications and the Internet Workshops, Orlando, USA, 2003: 116–119. doi: 10.1109/SAINTW.2003.1210138

[5] 兰巨龙, 程东年, 胡宇翔. 可重构信息通信基础网络体系研究[J]. 通信学报, 2014, 35(1): 128–139. doi: 10.3969/j.issn.1000-436x.2014.01.015

LAN Julong, CHENG Dongnian, and HU Yuxiang. Research on reconfigurable information communication basal network architecture[J]. *Journal on Communications*, 2014, 35(1): 128–139. doi: 10.3969/j.issn.1000-436x.2014.01.015.

[6] 胡宇翔, 董芳, 王鹏, 等. 面向多样化服务定制的多态路由机制研究[J]. 通信学报, 2015, 36(7): 48–59. doi: 10.11959/j.issn.1000-436x.2015096.

HU Yuxiang, DONG Fang, WANG Peng, *et al*. Research on polymorphic routing mechanism for customized diversified services[J]. *Journal on Communications*, 2015, 36(7): 48–59. doi: 10.11959/j.issn.1000-436x.2015096.

[7] XIAO J and BOUTABA R. QoS-aware service composition and adaptation in autonomic communication[J]. *IEEE Journal on Selected Areas in Communications*, 2005, 23(12): 2344–2360.

[8] WANG P, LAN J, ZHANG X, *et al*. Dynamic function composition for network service chain: Model and optimization[J]. *Computer Networks*, 2015, 92: 408–418. doi: 10.1016/j.comnet.2015.07.020.

[9] LEE K, YOON H, and PARK S. A service path selection and adaptation algorithm in service-oriented network virtualization architecture[C]. 2013 IEEE International Conference on Parallel and Distributed Systems (ICPADS), Seoul, South Korea, 2013: 516–521. doi: 10.1109/ICPADS.2013.93.

[10] PAIK I, CHEN W, and HUHNS M N. A scalable architecture for automatic service composition[J]. *IEEE Transactions on Services Computing*, 2014, 7(1): 82–95. doi: 10.1109/TSC.2012.33.

[11] GHEZZI C, PANZICA LA MANNA V, MOTTA A, *et al*. Performance-driven dynamic service selection[J]. *Concurrency and Computation*: *Practice and Experience*, 2015, 27(3): 633–650. doi: 10.1002/cpe.3259.

[12] CHOI S, TURNER J, and WOLF T. Configuring sessions in programmable networks[J]. *Computer Networks*, 2003, 41(2): 269–284. doi: 10.1016/S1389-1286(02)00396-1.

[13] BARI F, CHOWDHURY S R, AHMED R, *et al*. Orchestrating virtualized network functions[J]. *IEEE Transactions on Network and Service Management*, 2016, 13(4): 725–739. doi:10.1109/TNSM.2016.2569020.

[14] RAMAN B and KATZ R H. Load balancing and stability issues in algorithms for service composition[C]. Twenty-Second IEEE Annual Joint Conference of the IEEE Computer and Communications. San Francisco, USA, 2003, Vol. 2: 1477–1487. doi: 10.1109/INFCOM.2003.1208983.

[15] HUANG X, GANAPATHY S, and WOLF T. A distributed routing algorithm for networks with data-path services[C]. 2008 Proceedings of IEEE 17th International Conference on Computer Communications and Networks. St. Thomas, USA, 2008: 1–7. doi: 10.1109/ICCCN.2008.ECP.32.

[16] VISSICCHIO S, TILMANS O, VANBEVER L, *et al*. Central control over distributed routing[J]. *ACM SIGCOMM Computer Communication Review*, 2015, 45(4): 43–56. doi: 10.1145/2829988.2787497.

[17] CARIA M, JUKAN A, and HOFFMANN M. SDN partitioning: A centralized control plane for distributed routing protocols[J]. *IEEE Transactions on Network and Service Management*, 2016, 13(3): 381–393. doi: 10.1109/TNSM.2016.2585759

[18] GU X, NAHRSTEDT K, and YU B. SpiderNet: An integrated peer-to-peer service composition framework[C]. 13th IEEE International Symposium on High performance Distributed Computing, Honolulu, USA, 2004: 110–119. doi: 10.1109/HPDC.2004.1323507.

[19] GU X and NAHRSTEDT K. Distributed multimedia service composition with statistical QoS assurances[J]. *IEEE Transactions on Multimedia*, 2006, 8(1): 141–151. doi: 10.1109/TMM.2005.861284.

[20] GU X and NAHRSTEDT K. On composing stream applications in peer-to-peer environments[J]. *IEEE Transactions on Parallel and Distributed Systems*, 2006, 17(8): 824–837. doi: 10.1109/TPDS.2006.107.

[21] LEE K and PARK S. A service path selection and adaptation algorithm for QoS assurance and load balancing in context-aware service overlay networks[J]. *International Journal of Web and Grid Services*, 2015, 11(3): 265–282. doi: 10.1504/IJWGS.2015.070963.

[22] CANFORA G, DI PENTA M, ESPOSITO R, *et al*. QoS-aware replanning of composite web services[C]. IEEE International Conference on Web Services (ICWS'05), Orlando, USA, 2005: 121–129. doi: 10.1109/ICWS.2005.96.

[23] FRANCIOS P and BONAVENTURE O. Avoiding transient loops during the convergence of link-state routing protocols [J]. *IEEE/ACM Transactions on Networking* (*TON*), 2007, 15(6): 1280–1292. doi: 10.1109/TNET.2007.902686.

[24] SALAMA H F. Multicast Routing for Real-time Communication of High-speed Networks[M]. Raleigh, USA, North Carolina State University, 1996: 198.

李　丹：　男，1989 年生，博士生，研究方向为新型网络体系结构.

兰巨龙：　男，1962 年生，教授，博士生导师，主要研究方向为网络体系结构、信息安全.

王　鹏：　男，1985 年生，博士，主要研究方向为新型网络体系结构、路由技术.

胡宇翔：　男，1982 年生，博士，主要研究方向为新型网络体系结构、网络安全.