

基于分布式选择探测算法的服务路由机制

梁宁宁¹, 兰巨龙², 张 岩²

(1. 解放军信息工程大学信息工程学院, 河南郑州 450002; 2. 国家数字交换系统工程技术研究中心, 河南郑州 450002)

摘 要: 当前网络中间件服务部署方式僵化单一, 难以支持未来多样化的网络服务需求. 针对此问题, 本文从可重构信息通信基础网络中引入元服务的思想, 提出一种支持在控制层进行定制编排以及在数据面按需部署的服务路由机制. 该机制将服务路径构建问题建模为多约束最优化问题, 并针对性设计了分布式选择探测算法, 通过对探针的分布式选择处理进行最优服务路径的构建. 仿真结果表明, 相较于现有算法, 本文提出的服务路由机制在付出较小探测开销的情况下, 有效提高了路径服务质量以及服务请求成功率.

关键词: 服务路由; 中间件服务; 元服务; 分布式选择探测

中图分类号: TP393.1

文献标识码: A

文章编号: 0372-2112 (2017)07-1545-08

电子学报 URL: <http://www.ejournal.org.cn>

DOI: 10.3969/j.issn.0372-2112.2017.07.001

A Service Routing Mechanism Based on the Distributed Selection Probing Algorithm

LIANG Ning-ning¹, LAN Ju-long², ZHANG Yan²

(1. Institute of Information System Engineering of the PLA Information Engineering University, Zhengzhou, Henan 450002, China;

2. National Digital Switching System Engineering & Technological R&D Center, Zhengzhou, Henan 450002, China)

Abstract: The current way of service deployment using network middleware is too rigid to support the future demand for a variety of network services. Aiming at this problem, this article puts forward a service routing mechanism supporting the customized scheduling in the control layer and the on-demand deployment in the data plane, which is based on the introduction of atomic service from the reconfigurable information communication basal network. According to the operation of this mechanism, this paper models the service routing problem as the multiple target optimization problem under the multi-restrictions, and designs the distributed selection probing algorithm, which constructs the optimal service path through the process of distributed probe selection. Compared with the existing algorithms, the results of simulation show that the service routing mechanism improves the success rate of the service requests and network resource utilization and reduces the service deployment costs.

Key words: service routing; middleware; atomic service; distributed selection probing

1 引言

现有网络大多都采用在网络重点环节部署中间件设备^[1] (Middle Box, MB) 的方式来提供相应服务功能, 如防火墙、入侵检测系统、缓存代理等. 然而伴随着云端应用、电子商务、数据中心等多样化网络服务的迅猛发展, 传统的网络服务部署方式已经难以适应业务的多样化需求^[2], 其主要表现在: 一方面现有服务功能硬件化的实现方式存在建设周期长、成本高、重复性等弊端;

另一方面网络服务的静态部署, 导致其难以适应上层多样化业务需求的动态变化, 易造成网络资源浪费^[3]. 为此, 新型网络服务模式的研究成为当前的一个研究热点.

相关研究中, 研究人员相继提出了 SOA^[4] (Service Oriented Architecture, SOA), SILO^[5] (Architecture for Services Integration, Control, and Optimization, SILO) 以及 SCN^[6] (Service Centric Network, SCN) 网络服务模型. 但是这些只关注于宏观网络服务架构, 没有涉及到架构

收稿日期: 2015-11-24; 修回日期: 2016-08-11; 责任编辑: 孙瑶

基金项目: 国家 973 重点基础研究发展计划 (No. 2012CB315901, No. 2013CB329104); 国家 863 高技术研究发展计划 (No. 2013AA013505); 国家自然科学基金 (No. 61309019, No. 61372121)

实现以及服务组合关键问题. 近年来, 软件定义网络^[7] (Software Defined Network, SDN) 以及网络功能虚拟化^[8] (Network Function Virtualization, NFV) 的兴起加速了网络服务部署方式的革新式发展. 在此方面, Qazi 等人^[9] 提出的 SIMPLE 方案利用 SDN 技术设计了对数据层中网络功能中间件执行策略的管理机制, 并考虑了资源利用效率. Gember 等人^[10] 基于软件定义网络中间件的思想设计实现了带状态 MB 来简化网络功能的管理. 但是它们在控制层面均缺少服务功能编排机制. OpenNF 方案^[11] 提出通过有效合作控制实现流表的再分配, 但是没有提及如何根据特定需求整合不同的流表. Mogul 等人^[12] 设计了一个 Corybantic 控制平面, 支持在单一控制器中进行模块化的编程, 实现优化组合. 但是 Corybantic 只能运行在单一控制器中, 且考虑的是上层控制应用模块的组合. 为了支持服务功能组合, 因特网工程任务组 (IETF) 专门创立服务功能链 (Service Function Chain, SFC) 工作组^[13], 但 SFC 主要针对传统网络 L4-L7 层提出的服务组合策略, 底层服务功能缺乏灵活重构能力, 且没有考虑未来网络的演进. 在服务路由由相关算法方面, Choi 等人^[14] 提出了分层服务图算法, 但是由于采用集中式控制, 而且没有考虑节点过载情况. Wolf 等人^[15] 提出了分布式服务矩阵路由算法, 但是仅使用单一时延指标来度量通信以及处理代价, 算法可扩展性不高.

综上, 如何支持未来不断涌现的新兴服务需求, 增强网络灵活智能的服务能力, 从而支持多样化网络服务定制是一个亟待解决的问题. 针对以上问题, 本文从可重构信息通信基础网络^[16] 中引入元服务的思想, 提出使用元服务对网络服务进行统一资源描述, 进而提出基于元服务的可重构服务模型, 即将传统集中固定僵化的中间件服务拆分为基于元服务描述的多个服务处理单元, 并将其有序部署在数据传输路径之中, 从而增强网路服务能力的灵活性. 针对可重构服务模型中所涉及的服务路径构建问题, 本文将建模为多约束条件下最优化问题, 并针对性地提出了服务路由机制, 该机制采用分布式选择探测算法, 通过在网络释放探针搜索服务路径, 然后在各个服务节点中进行探针优化选择, 以分布式选择方式实现服务路径的按需构建. 最后, 基于 ClickOS 平台对文中所提服务路由机制进行了验证分析.

2 服务路由机制

2.1 元服务基本概念

可重构信息通信基础网络将基础网络服务分解为细粒度的功能单元, 提出了元服务 (Atomic Service, SC) 思想. 元服务作为网络服务划分的原子要素, 为开发者

提供统一的服务描述, 并为后续网络服务组合编排提供基础.

元服务是实现网络服务的最小功能单元. 可重构网络将元服务实例描述为五元组结构 $\langle \text{Type}, \text{ID}, \text{Attributes}, \text{Action}, \text{Provider} \rangle$ (即 \langle 类型, 标识, 属性集, 动作, 提供者 \rangle). 类型代表元服务的类别, 所有具有相同类型的实例也具有相同的功能. 标识是一个数字, 用 “Type.ID” 全网唯一地标识一个元服务实例. 属性集包括处理目标和性能参数, 其中处理目标是指该元服务实例处理的对象, 性能参数是指该元服务实例的参数化的性能指标, 如响应时间、存储代价、可靠性、依赖关系 (提供前、后向组合接口) 等. 动作代表该元服务对处理对象执行的操作. 提供者代表元服务实例的提供商. 元服务可以通过 XML 语言进行语义化描述. 算法 1 是一个关于防火墙元服务实例的语义描述.

算法 1 防火墙元服务 XML 实例

```
< capacity >
  < type > FIREWALL </ type >
  < ID > </ ID >
  < name > FIREWALL_ HTTP </ name >
  < attr >
    < logic : and >
      < property name = " dst _ port " >
        80
      </ property >
      < property name = " dst _ IP " >
        115. 64. 0. 0/16
      </ property >
      < property name = " delay " >
        5
      </ property >
    </ logic : and >
  </ attr >
  < action > drop </ action >
  < provider > NDSC </ provider >
</ capacity >
```

2.2 可重构服务模型

基于元服务可动态组合的特点, 本文所提服务路由机制采用可重构服务模型 (Reconfigure Service Model, Re-Service). 如图 1 所示, 可重构服务模型将传统层次化的协议栈进行基于元服务的分解, 在传统网络层之上构建一个可重构服务层. 该层支持进行元服务的组合, 从而将传统数据流处理过程从端节点迁移至数据流传输路径之中, 增强中间节点处理能力, 从而加速数据处理过程, 能够通过元服务的编排定制以及灵活部署为上层多样化业务提供服务保证. 可重构服务模型

中,只要节点资源满足要求,元服务可以根据需要部署于网络的任意节点之上,不同元服务之间可以组合编排形成服务链来满足复杂业务需求.如图 1 所示,一个端到端服务请求根据特定需求从网络元服务资源池中选择特定的元服务,按次序组合编排形成元服务链完成相应的服务请求.

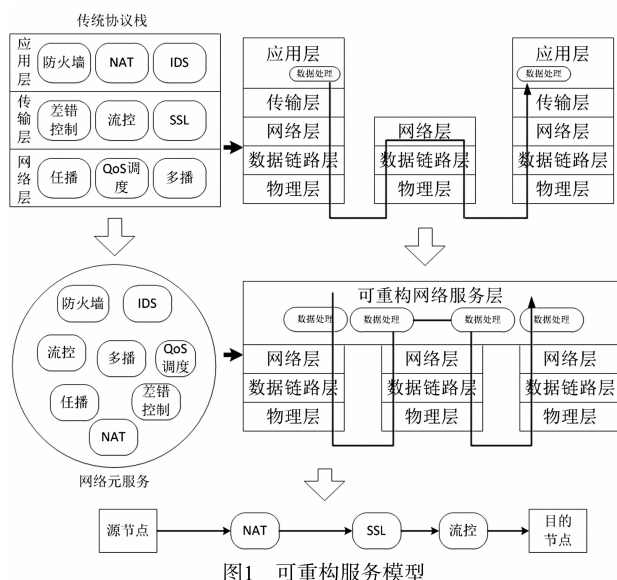


图1 可重构服务模型

可重构服务模型采用 NFV 架构.在底层数据面,服务节点采用统一标准化的硬件设备(如 OpenFlow 交换机)构建网络数据报文的处理和传输通道,主要负责接收管理面下发的元服务部署策略,对数据报文执行具体的数据处理操作,实现相应服务.在管理面,管理节点由软件管理模块分析上层业务需求并结合网络状态组合形成元服务链,接着映射元服务链至搜索路径中相应服务节点之上,进行服务部署.当服务请求结束后,考虑到后续服务请求可能同样需要相应元服务实例,元服务实例会首先保留一段时间,随后可根据特定策略进行移除(如近期最少使用策略),从而保证节点资源的可用性.同一元服务实例在同一时刻只能为一种业务提供服务,但元服务实例支持分时功能复用.

可重构服务模型中,为满足业务的多样化服务需求,管理节点需要根据网络对位于多个服务节点上的元服务进行组合,并选择能够到达目的节点的一条最优路径,且路径上的服务节点必须具备满足业务需求的元服务,这种跨越全网、满足特定业务传送需求的“节点-链路”序列就称为服务路径.可重构服务模型中,如何根据上层用户的服务请求构建一条最优的服务路径并部署相应元服务,便是可重构服务模型运行中最关键的一环.

2.3 服务路由运行场景

服务路径构建问题涉及到服务组合、服务布局以

及约束路由问题,与传统分离式的研究策略不同,本文提出的服务路由机制基于可重构服务模型,创新性地将其进行统一结合考虑,即在考虑实际物理网络约束(如节点处理能力、链路带宽等)、节点的服务能力(能够提供的元服务)以及元服务依赖关系的情况下,在网络中选择一条最优的路径,并将相应元服务组合链映射部署到特定的节点集合中,从而构建得到满足需求的服务路径.图 2 所示为服务路由机制的运行场景.

本文所提服务路由机制充分利用可重构服务模型中元服务可按需部署、灵活组合的特点,在进行约束式路由的同时,考虑服务组合链的构建,最后通过服务映射实现最优服务布局,从而有效降低了分离式研究中各个问题的复杂度.下一小节将对此问题进行建模分析.

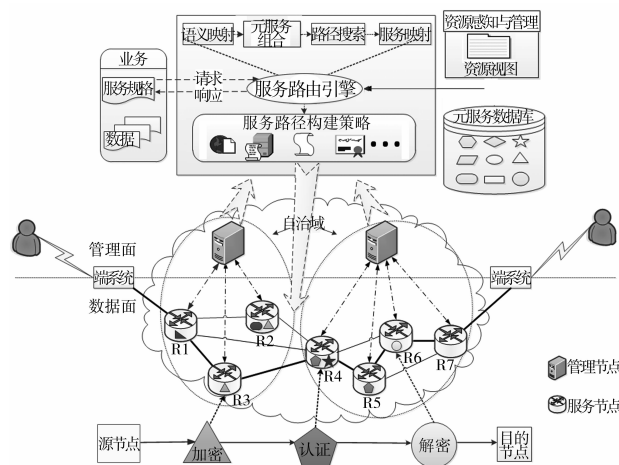


图2 服务器路由运行场景

3 问题建模

根据前述分析,本文将服务路由问题(Service Routing Problem, SRP)建模为多约束最优化问题.文献[17]证明这是一个 NP 难问题.为了解决此问题,本文首先对 SRP 涉及到的多个概念进行定义,具体表述如下.

定义 1 服务请求 R

服务请求表示用户的特定服务请求,是构建服务路径的基础.定义如下:

$$R = \{v_s, v_d, Q_{\text{req}}, [S_{k_1}, S_{k_2}, \dots, (S_{k_i}, \dots, S_{k_j}), \dots, S_{k_n}]\} \quad (1)$$

其中, v_s 表示服务请求源节点, v_d 表示服务请求目标节点, Q_{req} 表示服务请求的质量要求,包括带宽、时延、处理能力等. $[S_{k_1}, S_{k_2}, \dots, S_{k_n}]$ 表示服务请求所需的元服务集合.其中,考虑元服务之间的依赖关系,有序元服务集合定义如下:

$$(S_{k_i}, S_{k_j}) : S_{k_i} \rightarrow S_{k_j} \quad (2)$$

表示元服务 S_{k_i} 依赖于元服务 S_{k_j} ,在服务实现过程中不

能交换相应元服务的先后次序. 无序元服务集合定义如下:

$$[S_{k_1}, S_{k_2}]: S_{k_1} \leftrightarrow S_{k_2} \quad (3)$$

表示元服务 S_{k_1} 不依赖于元服务 S_{k_2} , 在服务实现过程中可以交换相应元服务的先后次序. 因此, 根据服务请求 R 可以构建相应元服务链 SC , 表示如下:

$$SC: S_{k_1} \leftrightarrow S_{k_2} \rightarrow S_{k_2} \rightarrow S_{k_3} \leftrightarrow S_{k_4} \quad (4)$$

定义 2 服务路径 P

服务路径 P 表示一个候选直连序列. 定义如下:

$$P = \{E^P, M^P\} = \{v_0, e_0, v_1, e_1, \dots, v_n, e_n\} \quad (5)$$

$$(n > 0), e_i = (v_{i-1}, v_i)$$

其中, E^P 表示边集合, M^P 表示节点集合, v_0 表示源节点, v_n 表示目的节点, e_i 表示节点 v_i 与 v_{i-1} 之间的连接边. $U_i = \{S_{k_i} | S_{k_i} \rightarrow V_i\}$ 表示服务节点 v_i 能够提供的特定元服务集合.

定义 3 服务质量 Q_{path}

服务质量 Q_{path} 是服务路径端与端之间的评价标准. 评价标准是由域内管理节点进行定义, 可根据用户需求进行灵活扩展. 定义如下:

$$Q_{\text{path}} = \frac{1}{C_{\text{path}}},$$

$$C_{\text{path}} = \omega_p \sum \frac{p_{s_i}}{q_{v_i} + p_{s_i}} + \omega_b \sum \frac{b_{s_i}}{c_{v_i} + b_{s_i}} + \omega_d \frac{D}{D + D_{\text{max}}},$$

$$c_{v_i} = \sum l_{v_i, v_{i+1}} \quad (6)$$

C_{path} 表示服务路径代价函数. 其中, p_{s_i} , b_{s_i} 分别表示元服务 S_i 所需的计算资源与带宽资源. q_{v_i} 表示提供元服务 S_i 的节点的剩余计算资源, c_{v_i} 表示节点 v_i 与所有邻居节点 v_{i+1} 之间直连链路的可用带宽之和, $l_{v_i, v_{i+1}}$ 表示节点 v_i 与邻居节点 v_{i+1} 的直连链路可用带宽, n 表示路径 P 所包含的节点个数. D 表示服务路径的端到端时延, D_{max} 表示服务路径能够允许的最大端到端时延. ω_p , ω_b , ω_d 分别表示针对计算资源、带宽资源以及时延指标的权重因子. 参考相关文献[18, 19], 本文所选指标均属于服务路径构建中关键影响因素. 此外, 式(6)对其进行了无量纲处理, 因此便于后续扩展.

定义 4 等价服务路径

当两条服务路径所经过节点集合提供的元服务序列相同时, 即称之为等价服务路径. 定义如下:

$$P_x = P_y: \forall v_x \in M^P, \forall v_y \in M^P, \cup U_x = \cup U_y \quad (7)$$

定义 5 SRP 问题

服务路由问题 SRP 可描述为一个多约束最优化问题. 服务网络可以使用图表示为 $G = (V, E)$. 其中, 节点集合 V 表示网络中服务节点集合, E 表示节点的连接关系. 通过对图 G 的搜索, 找到一条路径满足下列条件:

①每个节点资源不发生超载; ②每条链路不发生拥塞; ③路径上节点所提供元服务序列满足服务请求中的元服务要求; ④优化目标是最大化服务路径质量. 综上所述, SRP 定义如下:

$$\text{Object max } Q_{\text{path}}$$

$$\text{s. t. } \begin{cases} \text{path} = (E^P, M^P) \\ \forall v_i \in M^P, q_{v_i} \geq 0, c_{v_i} \geq 0 \\ \forall e_i \in E^P, l_{v_i, v_{i+1}} \geq 0 \\ \cup U_i = \{S_i | S_i \in SC, i = 1, \dots, n\} \end{cases} \quad (8)$$

4 分布式选择探测算法

考虑到现有网络以及可重构网络均采用分布式集中管理控制方式, 本文针对性地设计了分布式选择探测算法来进行服务路径构建. 分布式选择探测算法的基本思想是在目的节点生成路径查询探针, 向满足其条件约束的邻居节点进行发送. 其关键点在于分布式选择的实现, 即不像传统探测算法那样仅仅在端节点进行探针选择, 而是在网络中所有节点均进行探针的选择, 只有当探针测得的服务路径比之前探针测得的服务质量高时, 节点才转发此探针, 因此大幅减少了网络探测开销. 在服务路径探测过程中, 当某一节点能够提供服务请求中的相应元服务时, 则在此节点进行资源预留. 最后, 当服务路径构建完成后, 源节点向所有候选服务路径发送路径确认消息, 对最终选择满足要求的服务路径进行资源确认并统一部署相应元服务, 同时对其它未被选择的候选服务路径进行资源释放, 以保证其它服务请求.

分布式选择探测算法主要分为两部分: 探针处理, 探针转发. 探针处理过程依据 SRP 最优化模型中的节点资源约束、链路约束以及服务质量约束条件做出优化决策, 决定将探针丢弃或者加入至转发队列. 探针转发过程依据 SRP 最优化模型中的元服务约束条件筛选合适的邻居节点, 然后转发各节点的探针缓存信息, 并周期性清除探针缓存.

针对服务路径构建这一多约束最优化问题, 考虑到此问题具有最优子结构性质, 即一个最优化策略的子策略总是最优的. 因而, 本文基于动态规划的思想, 把多阶段过程转化为一组单阶段问题, 利用各阶段之间的关系逐个求解, 然后从这些子问题的解得到原问题的解. 具体来说, 本文提出的分布式选择探测算法将最优服务路径分解为满足要求的多个节点, 采用分布式处理, 依据探针信息搜索满足要求的节点, 从而逐步构建最优服务路径.

图 3 是分布式选择探测算法的一个探针转发过程示例.

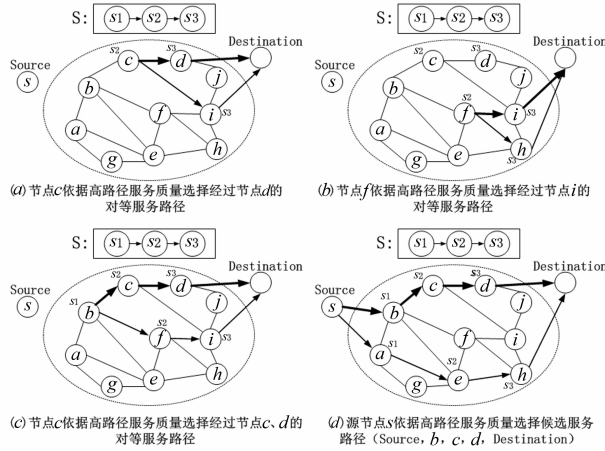


图3 分布式选择探测算法转发过程示例

考虑到元服务间的依赖关系,图4展示了分布式探测机制的探针处理过程.即探针依次将节点 d, c, a, b, a 加入服务路径中,同时根据节点能力在各个节点上依次对元服务 $S5, (S4, S6), S2, S3, S1$ 进行资源预留.由于元服务 $S1$ 与 $S3$ 存在依赖关系,因此节点 a 在服务路径中先后出现两次,先后分别为元服务 $S2$ 和 $S1$ 进行资源预留.

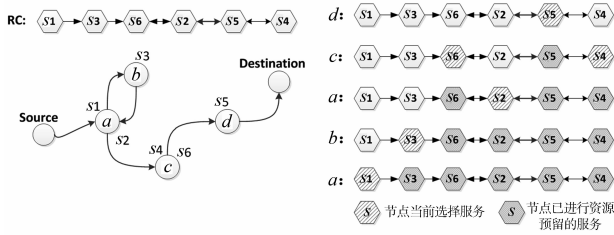


图4 分布式选择探测算法探针处理过程示例

此外,本文还考虑通过定义相关参数来限制探针数量,从而减低算法开销.参数定义如下:

(1) 缓存计时器 T .节点每隔 T 毫秒,清除一次缓存,即 T 毫秒表示一个探测处理周期.

(2) 转发限制参数 \bar{N} .每个节点针对特定服务组合会话最多转发 \bar{N} 个探针.

(3) 扩散限制参数 \bar{M} .每个节点探针随机转发至 \bar{M} 个合格邻居节点.

(4) 边界限制参数 \bar{H} .当一个探针经过 \bar{H} 个节点之后仍然没有找到一条合适的服务路径,则丢弃该探针.

(5) 选择限制参数 Q .当且仅当探针探测到的服务路径超出原有最优服务路径的服务质量达到 Q (百分数)时,节点根据探针信息更新服务路径,转发探针.

分布式选择探测算法的具体过程如算法2所示.

算法2 Algorithm: Distributed Selection Probing

Probe Processing:

1. **On** receiving probe p on node n through edge e
 2. **if** n does not fit in p .serviceChain, **then**
 3. drop p and **return**
 4. **if** p .quality < expectedQuality(p) **or** p .quality < buffer.bestQuality(p), **then**
 5. drop p and **return**
 6. p .path = annex(n, e, p) and reserve resource for p .serviceChain on node n
 7. assign p to buffer
 8. **return**
- Probe Forwarding:**
9. **On** probe assignment to buffer
 10. **if** buffer is empty **then** start timer Timer
 11. buffer.setBufferBestQuality(p)
 12. **return**
 13. **On** buffer Timer timeout
 14. **for** each probe p in the buffer
 15. **if** probe.serviceChain is achieved, **then**
 16. **if** probe.target == n , **then**
 17. map the p .serviceChain along the optimal path, release resource along unselected paths
 18. **else** send probe to probe.target, goto step 16
 19. **else**
 20. **if** numberOfHopsTraversed(probe) > \bar{H} **or** n .numberOfForwards(probe.sessionID) > \bar{M} , **then**
 21. remove probe from buffer and **return**
 22. Forward probe to not more than \bar{N} qualified nodes in n .neighbors
 23. Increase n .numberOfForwards(probe.sessionID)
 24. **end for**
 25. setExpectedQuality(probe, probe.quality * (1 + Q))
 26. remove probe from buffer
 27. **return**

5 实验结果与分析

本文结合伦敦大学学院(University College London, UCL)开发的轻量级网络试验平台^[20](Very Lightweight Network & Service Platform, VLSP)以及ClickOS平台^[21]进行仿真实验.本文所提服务路由机制采用可重构服务模型为基础,基于分布式选择探测算法进行服务路径构建.因此,本文从路径服务质量、端到端时延、探测开销以及请求成功率方面对服务路由机制进行性能测试与分析.

5.1 R-DSP 性能分析

为了更好地说明本文所提服务路由机制的优越性,文中对比分析四种服务策略(包含服务部署方式以及服务路径构建方法).

(1) Tradition-Along:采用传统的中间件服务部署方式,即在每个节点固定部署元服务.采用沿路搜索方式,即依据元服务种类逐节点按序搜索服务路径,不采用

分布式选择探测算法,即传统网络服务路径构建模式.

(2) Reconfigure-Along: 采用可重构服务模型,即每个节点可以部署任意元服务. 采用沿路搜索方式,即依据节点能力逐节点按序搜索服务路径,但不采用分布式选择探测算法.

(3) Tradition-DSP: 采用传统的中间件服务部署方式. 采用分布式选择探测算法构建服务路径.

(4) Reconfigure-DSP: 采用可重构服务模型,采用分布式选择探测算法构建服务路径,即本文所提服务路由机制.

图 5(a)(b) 表明可重构服务模型相比传统中间件服务部署方式能够将路径服务质量提升大约 11%, 这

是因为服务在可重构服务模型中的任意节点可按需部署,提高了网络资源利用率. 分布式选择探测算法在可重服务模型的基础上,能够将路径服务质量提高 15% 左右,这是因为 DSP 算法充分利用可重构模型中服务按需部署的特点,在最优服务路径构建的同时进行资源预留,始终按照最优服务路径进行服务部署. 图 5(c)(d) 表明分布式选择探测算法能够有效降低传统中间件服务部署模式中 93% 的探测开销,有效降低可重构服务模型中 97% 的探测开销. 由此表明,文中所提服务路由机制(R-DSP)采用可重构服务模型为基础,以分布式选择探测算法构建服务路径,能够在付出较小探测开销的情况下,将路径服务质量提升大约 24%.

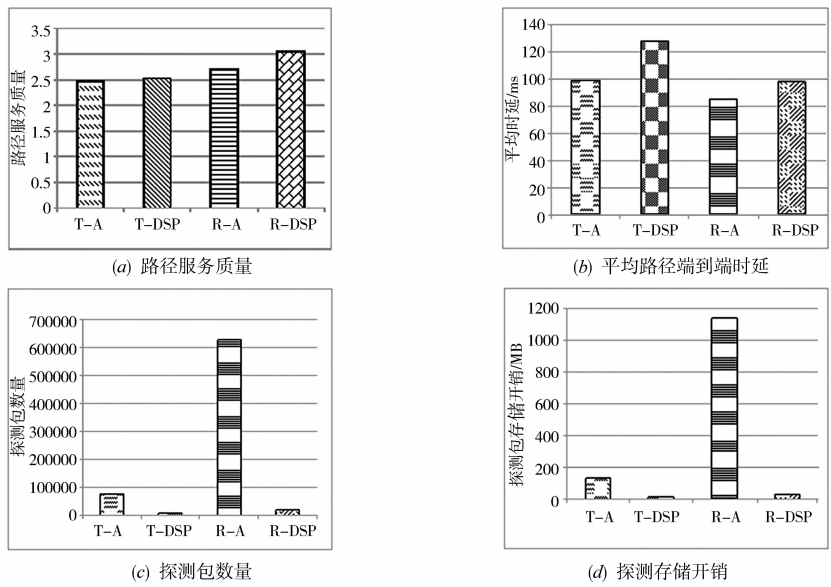


图5 四种服务策略下指标对比

5.2 服务请求成功率对比

本节中,本文选择引言中提到的分层服务图算法(Layered Graph)以及分布式服务矩阵路由算法(DSMR)进行性能对比,它们都是采用类似可重构的服务模型进行服务路径构建. 由于分层服务图算法以及DSMR算法都是以时延作为测量指标,因此本节主要分

析比较服务路径端到端时延来评估服务路由机制性能.

从图 6(a)可以看出,当曲线线性增长时,表示随着网络服务请求数量的增加,平均路径端到端时延逐渐增加. 当曲线趋于平稳时,表示网络达到最大服务请求承载量,无法再构建新的服务路径. 其中,分层服务图

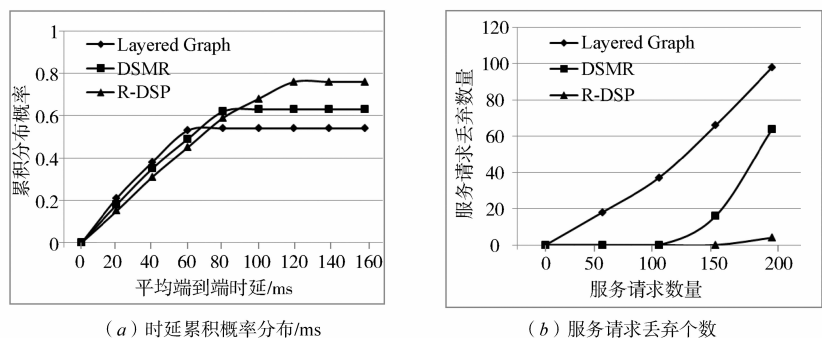


图6 各算法性能指标对比

算法最先达到网络最大服务承载能力,其次是 DSMR 算法,最后是本文提出的服务路由机制 (R-DSP). 这是由于服务路由机制采用可重构服务部署模式有效提高了网络资源利用率,同时采用分布式选择探测算法有效降低了探测开销,从而提高了最大服务请求承载量. 图 6(b) 表明服务请求成功率与最大服务承载量呈同步变化趋势. 因而,基于分层服务图算法的最大服务请求成功率约为 54%,DSMR 算法约为 63%,而 DSP 算法能够达到 76%. 由此表明,相比现有算法,服务路由机制将服务请求成功率提高了 20% 以上.

6 总结

针对现有网络中间件服务部署僵化单一,难以支持未来多样化网络服务需求的问题,本文基于可重构信息通信基础网络中的元服务思想,提出了支持元服务灵活编排、按需部署以及支持未来不断演进的服务路由机制. 针对服务路由机制运行中关键的服务路径构建问题,本文针对性地设计了分布式选择探测算法,通过在各个服务节点进行路径评估与筛选,极大降低了算法开销. 与现有算法相比,本文提出的算法在路径服务质量以及服务请求成功率上具有明显优越性. 本文提出的服务路由机制能够灵活有效满足未来多样化的网络服务需求,对于未来网络的不断演进具有重要意义.

参考文献

- [1] ANWER B, BENSON T, FEAMSTER N, et al. A slick control plane for network middleboxes [A]. Proceedings of the ACM SIGCOMM Workshop HotSDN [C]. USA: ACM Press, 2013. 147 – 148.
- [2] 张宏科, 罗洪斌. 智慧协同网络体系基础研究 [J]. 电子学报, 2013, 41(7): 1249 – 1254.
ZHANG Hong-ke, LUO Hong-bin. Fundamental research on theories of smart and cooperative networks [J]. Acta Electronica Sinica, 2013, 41(7): 1249 – 1254. (in Chinese)
- [3] 苏伟, 陈佳, 张宏科, 等. 智慧协同网络中的服务机理研究 [J]. 电子学报, 2013, 41(7): 1256 – 1260.
SU Wei, CHEN Jia, ZHANG Hong-ke, et al. Research on the service mechanisms in smart and cooperative networks [J]. Acta Electronica Sinica, 2013, 41(7): 1256 – 1260. (in Chinese)
- [4] ERL Thomas. Service-Oriented Architecture (SOA): Concepts, Technology, and Design [M]. England: Prentice Hall, 2005.
- [5] DUTTA R, ROUSKAS G N, BALDINE I, et al. The SILO architecture for services integration, control, and optimization for the future internet [A]. Proceedings of the IEEE International Conference on Communications Workshops (ICC) [C]. Glasgow, Scotland: IEEE Press, 2007. 1899 – 1904.
- [6] BRAUN T, HILT V, HOFMANN M, et al. Service-centric networking [A]. Proceedings of the IEEE International Conference on Communications Workshops (ICC) [C]. USA: IEEE Press, 2011. 1 – 6.
- [7] MCKEOWN N. Software-defined networking [J]. INFOCOM keynote talk, 2009, 17(2): 30 – 32.
- [8] Alcatel Lucent. Network Functions Virtualization: Introductory White Paper [OL]. http://www.tid.es/es/Documents/NFV_White_PaperV2.pdf, 2016 – 08 – 11.
- [9] QAZI Z A, TU C-C, CHIANG L, et al. SIMPLE-fying middlebox policy enforcement using SDN [J]. Proceedings of the SIGCOMM, 2013, 43(4): 27 – 38.
- [10] GEMBER A, PRABHU P, GHADIYALI Z, et al. Toward software-defined middlebox networking [A]. Proceedings of the ACM Hotnets '12 [C]. Seattle, WA, USA: ACM Press, 2012. 7 – 12.
- [11] GEMBER-JACOBSON A, VISWANATHAN R, PRAKASH C. OpenNF: enabling innovation in network function control [J]. Proceedings of the SIGCOMM, 2014, 44(4): 163 – 174.
- [12] MOGUL J C, AU YOUNG A, BANERJEE S, et al. Corybantic: towards the modular composition of SDN control programs [A]. Proceedings of the Hotnets '13 [C]. USA: ACM, 2013. 1 – 7.
- [13] QUINN P, BELIVEAU A. Service function chaining (SFC) architecture [J/OL]. Internet Engineering Task Force, 2015: <https://tools.ietf.org/html/rfc7665>.
- [14] CHOI S Y, TURNER J S, WOLF T. Configuring sessions in programmable networks [J]. Proceedings of the 20th IEEE INFOCOM, 2001, 41(2): 60 – 66.
- [15] HUANG X, GANAPATHY S, WOLF T. A scalable distributed routing protocol for networks with data-path services [A]. Proceedings of the 16th IEEE ICNP [C]. Orlando, FL: IEEE Press, 2008. 318 – 327.
- [16] 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): 187 – 198.
LAN Ju-long, CHENG Dong-nian, HU Yu-xiang. Research on reconfigurable information communication basal network architecture [J]. Journal of Communications, 2014, 35(1): 187 – 198. (in Chinese)
- [17] CHOI S Y, TURNER J S, WOLF T. Configuring sessions in programmable networks [J]. Computer Networks, 2003, 41(2): 269 – 284.
- [18] 苗春尧, 苏伟, 张宏科, 等. 一种智慧协同网络多参数的多路径路由算法 [J]. 电子学报, 2015, 43(10): 1882 – 1887.

- MIAO Chun-jiao, SU Wei, ZHANG Hong-ke, et al. A multi-parameter multipath routing algorithm of smart and cooperative networks [J]. Acta Electronica Sinica, 2015, 43(10): 1882 – 1887. (in Chinese)
- [19] DUAN Qiang, YAN Yu-hong, VASILAKOS Athanasios V. A survey on service-oriented network virtualization toward convergence of networking and cloud computing [J]. IEEE Transactions on Network and Service Management, 2012, 9(4): 373 – 392.
- [20] CLAYMAN S. User Space Routing, Open Source Software [OL]. <http://clayfour.ee.ucl.ac.uk/usr/>, 2016 – 08 – 11.
- [21] MARTINS J, AHMED M, RAICIU C, et al. ClickOS and the art of network function virtualization [A]. Proceedings of the 11th USENIX Conference on Networked Systems Design and Implementation [C]. USA: USENIX Press, 2014. 459 – 473.

作者简介



梁宁宁 女, 1982 年出生, 天津人. 博士, 解放军信息工程大学信息工程学院讲师, 主要研究方向为宽带信息网络和下一代网络.

E-mail: zy24987624@163.com



兰巨龙 男, 1962 年出生, 河北张北人. 国家数字交换系统工程技术研究中心总工程师、教授、博士生导师, 主要研究方向为新一代信息网络关键理论与技术, 目前作为首席科学家主持国家“973”项目“可重构信息通信基础网络体系研究”.