

放、动态特性产生了更多影响系统安全的不确定因素,无法使用绝对的、精确的标准来衡量系统的安全性;(4) 安全需求的个性化,不同的应用系统和服务对安全的需求不同,即使是组成同一应用系统的服务也具有不同的安全需求;(5) 安全措施实施的自适性,Web 环境是高度动态和多变的,安全措施的实施必须能够实时地适应这种变化。

信任管理提供了一个适宜解决新的 Web 应用形态下的安全问题的框架,但当前几个已实现的信任管理系统还存在着一些不足:(1) 安全分析主体的单一,仅考虑服务方的安全保护,没有考虑服务调用方的安全问题;(2) 安全度量的绝对化,采用策略一致性证明验证的方法进行安全度量和决策,该方法过于精确,不能很好地适应 Web 安全环境的多变性和不确定性;(3) 无法实时地满足动态的安全环境的变化,安全策略验证的能力和效率有限,并且大部分信任管理系统在策略一致性证明验证前必须收集足够的安全凭证,另外,安全策略的制定过程较为繁复,也阻碍了这些信任管理系统的应用。

一些主观信任模型提出的信任度量和评估,其实质是采用一种相对的方法对安全信息进行度量和评估,能够较好地反映出 Web 安全环境的多变性和不确定性,并且该方法较适合信任信息收集、评估的自动化实现。信任度量和评估与实际的安全策略的实施相结合将是信任管理系统实现的新途径,但必须看到,当前几个代表性的信任度评估模型还存在一些问题:(1) 信任的表述和度量的合理性有待于进一步解释,现有的模型倾向于采用事件概率的方式来表述和度量信任关系,都是基于一定的概率分布假设;(2) 不能很好地解决恶意推荐对信任度评估的影响,现有的模型大多采用简单算术平均的方法综合多个不同推荐路径的信任度;(3) 当前的信任度评估模型缺少灵活的机制,如参数设置,以反映不同实体进行信任评估时所具有的个性特点。

5 结束语

本文介绍了信任管理思想的出现,并就信任管理模型和几个典型的信任管理系统实现进行了详细的讨论和分析,另外还介绍和分析了与当前信任管理研究相关的几个具有代表性的主观信任度评估模型。随着 Web 应用的不断发展,Web 安全将具有更多的内涵。同时,Web 动态性、开放性的增强将给安全问题带来更多的不确定性和复杂性。基于策略的信任管理系统还存在着诸多问题,如一致性证明验证算法能力有限、策略制定过程每繁琐、难以处理不确定的安全信息等等。而主观信任度评估与安全策略实施相结合的信任管理思想刚刚被提出来,有待于进一步的研究和实现。

References:

- [1] Blaze, M., Feigenbaum, J., Ioannidis, J., *et al.* The role of trust management in distributed systems security. In: Secure Internet Programming: Issues for Mobile and Distributed Objects. Berlin: Springer-Verlag, 1999. 185~210.
- [2] Khare, R., Rifkin, A. Trust management on World Wide Web. World Wide Web Journal, 1997,2(3):77~112.
- [3] Blaze, M., Feigenbaum, J., Lacy, J. Decentralized trust management. In: Dale, J., Dinolt, G., eds. Proceedings of the 17th Symposium on Security and Privacy. Oakland, CA: IEEE Computer Society Press, 1996. 164~173.
- [4] Abdul-Rahman, A., Hailes, S. A distributed trust model. In: Proceedings of the 1997 New Security Paradigms Workshop. Cumbria, UK: ACM Press, 1998. 48~60. <http://www.ib.hu-berlin.de/~kuhlen/VERT01/abdul-rahman-trust-model1997.pdf>.
- [5] Abdul-Rahman, A., Hailes, S. Using recommendations for managing trust in distributed systems. In: Proceedings of the IEEE Malaysia International Conference on Communication' 97 (MICC' 97). Kuala Lumpur: IEEE Press, 1997. <http://citeseer.nj.nec.com/360414.html>.
- [6] Yahalom, R., Klein, B., Beth, T. Trust relationships in secure systems — a distributed authentication perspective. In: Proceedings of the 1993 IEEE Symposium on Research in Security and Privacy. IEEE Press, 1993. 50~164. <http://isbn.nu/0818633700>.
- [7] Beth, T., Borchering, M., Klein, B. Valuation of trust in open network. In: Gollmann, D., ed. Proceedings of the European Symposium on Research in Security (ESORICS). Brighton: Springer-Verlag, 1994. 3~18.
- [8] Blaze, M., Feigenbaum, J., Keromytis, A.D. Keynote: trust management for public-key infrastructures. In: Christianson, B., Crispo, B., William, S., *et al.*, eds. Cambridge 1998 Security Protocols International Workshop. Berlin: Springer-Verlag, 1999. 59~63.

- [9] Chu, Y.-H., Feigenbaum, J., LaMacchia, B., *et al.* REFEREE: trust management for Web applications. *World Wide Web Journal*, 1997,2(2):127~139.
- [10] Povey, D. Developing electronic trust policies using a risk management model. In: *Proceedings of the 1999 CQRE Congress*. 1999. 1~16. <http://security.dstc.edu.au/staff/povey/papers/CQRE/123.pdf>.
- [11] Gambetta, D. Can we trust trust? In: Gambetta, D., ed. *Trust: Making and Breaking Cooperative Relations*. Basil Blackwell: Oxford Press, 1990. 213~237.
- [12] Herrmann, P., Krumm, H. Trust-Adapted enforcement of security policies in distributed component-structured applications. In: *Proceedings of the 6th IEEE Symposium on Computers and Communications*. Hammamet: IEEE Computer Society Press, 2001. 2~8. <http://www.computer.org/proceedings/iscc/1177/11770002abs.htm>.
- [13] Blaze, M., Feigenbaum, J., Strauss, M. Compliance Checking in the PolicyMaker Trust Management System. In: *Proceedings of the Financial Cryptography' 98. Lecture Notes in Computer Science 1465*, Berlin: Springer-Verlag, 1998. 254~274.
- [14] Blaze, M., Ioannidis, J., Keromytis, A. Trust management for IPSec. In: *Proceedings of the Internet Society Symposium on Network and Distributed Systems Security (SNDSS 2001)*. 2001. 139~151. <http://www.cis.upenn.edu/~strongman/papers/tmipsec.pdf>.
- [15] Blaze, M., Ioannidis, J., Keromytis, A. Offline Micropayments without Trusted Hardware. In: Syverson, F. ed. *Financial Cryptography 2001. Lecture Notes in Computer Science 2339*, Berlin: Springer Verlag, 2002.
- [16] Blaze, M., Feigenbaum, J., Ioannidis, J., *et al.* The KeyNote trust management system version 2. Internet RFC 2704, 1999.
- [17] Jøsang, A. The right type of trust for distributed systems. In: Meadows, C., ed. *Proceedings of the 1996 New Security Paradigms Workshop*. Lake Arrowhead, CA: ACM Press, 1996.
- [18] Jøsang, A. A model for trust in security systems. In: *Proceedings of the 2nd Nordic Workshop on Secure Computer Systems*. 1997. <http://security.dstc.edu.au/staff/ajosang/papers.html>.
- [19] Jøsang, A., Knapskog, S.J. A metric for trusted systems. *Global IT Security*. Wien: Austrian Computer Society, 1998. 541~549.
- [20] Jøsang, A. A Subjective Metric of Authentication. In: Quisquater, J., ed. *Proceedings of the ESORICS' 98*. Louvain-la-Neuve.: Springer Verlag, 1998. 329~344.

Research and Development of Trust Management in Web Security*

XU Feng, LÜJian

(State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210093, China);

(Institute of Computer Software, Nanjing University, Nanjing 210093, China)

E-mail: xf@softlab.nju.edu.cn

<http://moon.nju.edu.cn>

Abstract: Trust management is a hot topic of Web security research in recent years. In this paper, the emerging of trust management is presented. Its concepts and models are described in detail, and several typical trust management systems and trust valuation models are introduced. The existing problems of current works and future research direction are discussed.

Key words: Web security; trust management

* Received February 25, 2002; accepted July 2, 2002

Supported by the National Natural Science Foundation of China under Grant No.60273034; the National High-Tech Research and Development Plan of China under Grant Nos.2001AA113110, 2002AA116010; the Foundation of Nature Science and Hi-Technology of Jiangsu Province of China under Grant Nos.BG2001012, BK2002203, BK2002409

互联网络服务质量路由算法研究综述*

崔 勇, 吴建平, 徐 恪, 徐明伟

(清华大学 计算机科学与技术系, 北京 100084)

E-mail: cy@csnet1.cs.tsinghua.edu.cn

http://netlab.cs.tsinghua.edu.cn/~cuiy

摘要: 如何提供不同的服务质量(quality of service,简称 QoS)是互联网络面临的一个重要问题,而服务质量路由(quality-of-service routing,简称 QoS SR)则是其中的核心技术和热点问题.QoS SR 的主要作用是为 QoS 业务请求寻找可行路径,这体现了 QoS SR 的两个目标:(1) 满足业务 QoS 需求;(2) 最大限度地提高网络利用率.由于 QoS SR 是 NP 完全问题,研究者们设计了很多启发式算法进行了广泛深入的研究.在有权图和 QoS 度量的基础上介绍了 QoS SR 的基本概念,详细分析了面向单播应用的 QoS SR 算法中的热点问题,并按照所求解的问题类型和求解方法,将这些算法分成以下几类:多项式非启发类、伪多项式非启发类、探测类、限定 QoS 度量类、路径子空间搜索类、QoS 度量相关类、花费函数类和概率求解类.在分析每类中典型算法的基础上,总结和对比了各类的特点,进而详细剖析了算法的有效性,并基于此总结了基于概率模型求解 QoS SR 问题的方法.最后指出了该领域中需要进一步研究的热点问题.

关键词: 服务质量路由;NP 完全问题;启发式算法;有效性

中图法分类号: TP393 **文献标识码:** A

当前 Internet 只提供尽力发送(best-effort)服务,网络层不区分用户业务的种类,而将网络资源公平地提供给各类业务,在分组丢失、延迟等方面公平地对待各类业务.这种尽力发送的机制使网络层无法保证传输的参数,然而丢失率、带宽、端到端延迟、延迟抖动等对于应用业务往往是至关重要的.例如,文件传输业务要求分组的丢失率尽可能低,而传输延迟不是关键因素;实时多媒体业务则更看重延迟和抖动.这就要求网络能够区别对待各种业务,并对它们提供不同的服务质量.由于 IP 是无连接的协议,不要求像 ATM 网络那样在数据传输前从源到目标节点建立连接.这种尽力发送的体系结构是无连接、与状态无关的,它既不能支持资源预留,也不能预测传输参数,甚至还可能产生多媒体实时业务不希望的乱序等.由此,面向服务质量的网络体系结构应运而生.

1 服务质量路由概述

定义 1(服务质量). 服务质量(quality of service,简称 QoS)是网络在传输业务流时,业务流对网络服务的需求的集合,其中业务流是指与特定 QoS 相关的从源到目的地的分组流^[1].

也就是说,QoS 是应用业务对网络传输服务提出的一组可度量的要求,主要包括带宽、端到端延迟、分组丢失率、抖动、花费等.网络在传输相应的数据业务时,必须满足这组要求.QoS 需求可以通过一个约束集来描述,包括链路约束、路径约束和树约束^[2].链路约束定义了从源到目的地的每一条链路的约束,如带宽约束;路径

* 收稿日期: 2001-12-31; 修改日期: 2002-04-10

基金项目: 国家自然科学基金资助项目(90104002;69725003);国家高技术研究发展计划资助项目(2002AA103067;2001AA121013)

作者简介: 崔勇(1976 -),男,新疆乌鲁木齐人,博士生,主要研究领域为计算机网络体系结构,协议的仿真和测试,多目标优化的路由算法及其评价;吴建平(1953 -),男,山西太原人,博士,教授,博士生导师,主要研究领域为计算机网络体系结构,计算机网络协议测试,形式化技术;徐恪(1974 -),男,江苏洪泽人,博士,讲师,主要研究领域为计算机网络体系结构,计算机系统性能评价;徐明伟(1971 -),男,辽宁朝阳人,博士,副教授,主要研究领域为计算机网络体系结构,计算机网络性能评价,协议测试.

约束定义了从源到目的地的一条路径上端到端 QoS 需求,如延迟;树约束定义了对组播中整个组播树的约束,例如对组播树延迟的约束是对树中从源到所有目的地的路径中最大延迟的约束。

定义 2(可行路径(可行树)). 可行路径(可行树)是网络中从源到(所有)目标节点的一条路径(组播树),并且该路径(树)具有足够的尚未分配的资源,能够满足特定的 QoS 需求。

定义 3(服务质量路由(QoS SR)). 服务质量路由(QoS routing)是一种基于数据流 QoS 请求和网络可用资源进行路由的机制^[1]。或者,QoS SR 是一种动态路由协议,并且在其路径选择标准里可能包含可用带宽、链路和端到端路径利用率、资源消费量、延迟、跳数以及抖动等 QoS 参数^[3]。

当前 Internet 的主要路由协议都是“尽力发送”的,选择路由时即便源到目的节点之间存在“更好的”路径,只要不是最短路径就不会投入使用,因此可能导致某些链路空闲时另外一些链路的拥塞。QoS SR 路由就是将传统的最短路径变为一条更好的路径,其主要目标包括以下两点^[1]。

- (1) 为每一个接纳的 QoS 业务连接请求,找到满足其 QoS 要求的可行路径(组播树);
- (2) 优化全局资源利用率,平衡网络负载,从而最大化网络接受其他 QoS 请求的能力。

为了提供 QoS 保证,数据传输前通常需要沿着计算好的路径,从源到目的地传播一个消息,用来通知路径上的所有节点为这个 QoS 业务保留相应的资源(如带宽、缓存等),而后续的数据传输则沿着这条已经预留了资源的路径。因此,QoS SR 具有面向连接的特性。

根据计算可行路径的时刻,QoS SR 可以分为预计算和在线计算两种。预计算(pre-computation)采用一个后台进程,根据网络状态信息来构造路由表,当 QoS 请求时,只需通过查找路由表就能确定可行路径^[4]。在典型网络设置中,QoS 连接请求的到达速率远大于网络的变化速度,因此可以使用预计算模式^[5]。由于 QoS 业务的多样性,路由表为了包含每个可能的 QoS 业务,其规模会相当庞大因而降低了可扩展性。在线计算则是当 QoS 请求到达时,再根据状态信息计算可行路径。这种方式虽然不需要事先构造路由表,但每次路由计算延迟较大,并且路由器负担很重。

QoS SR 问题很难完全解决的原因包括:(1) 寻找同时满足两个以上路径约束(优化)的可行路径,具有 NPC(NP-complete)的计算复杂度^[6];(2) 为了提高可扩展性所使用的层次化模型,产生了多种参数如何聚集的问题^[7];(3) 网络状态信息的陈旧性极大地影响了 QoS SR 算法的性能^[8];(4) 将 QoS SR 融入到当前这种尽力发送的路由体系结构中,原有的尽力发送业务将受到巨大的冲击。

2 网络模型和 QoS 度量

2.1 有权图模型

网络由交换节点、链路和主机组成,在研究当中往往将其抽象为有权图 $G(V,E)$ 。

定义 4(有向图). 有向图 G 是由一个非空有限集合 $V(G)$ 和其中某些元素的有序对集合 $E(G)$ 构成的二元组,记为 $G=(V(G),E(G))$ 。其中, $V(G)$ 称为图 G 的节点集,元素 $v \in V$ 称为图 G 的一个顶点或节点; $E(G)$ 称为图 G 的边集,元素 $e_{ij} \in E$ 记为 $e(v_i, v_j)$ 或 $e_{ij}=v_i v_j$,为 V 中元素的有序对,称为图 G 的一条从 v_i 到 v_j 的边(弧)。

定义 5(有权图). 在有向图 $G=(V,E)$ 中,令元素 $e \in E$ 具有一组有序数列 (w_1, w_2, \dots, w_k) 作为 e 的属性,则称 G 为有权图,其中 (w_1, w_2, \dots, w_k) 称为弧 e 的度量(权值)。

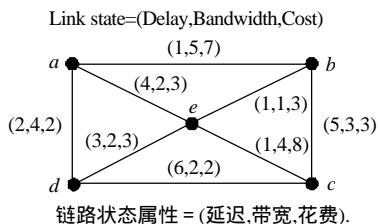


Fig.1 Weighted graph network model
图 1 有权图网络模型

节点集 V 可以表示路由器、交换机、集线器等网络节点设备,在研究路由问题时, V 中的元素为具有路由能力的交换节点; E 则为连接 V 中两个节点的链路及具有的 k 个属性 (w_1, w_2, \dots, w_k) ,这些属性可以是可用带宽、延迟、队列长度、花费等参数。对于对称网络有 $e_{ij}(w_1, w_2, \dots, w_k)=e_{ji}(w_1, w_2, \dots, w_k)$,而不对称网络中通常 $e_{ij} \neq e_{ji}$ 。现实中的网络往往是不对称的,但在研究中为了简便起见,常常使用对称网络模型以减少弧的数量^[9],本文亦如此。如图 1 所示为一个有权图模型。

定义 6(路径). 在图 G 中,如果对 $i=1,2, \dots, k-1$ 有 $v_i, v_{i+1} \in E$, 则 $P=(v_1, v_2, \dots, v_k)$ 为图 G 的一条从 v_1 到 v_k 的路径, 也记作 $P=\{e_{i,i+1}|i=1,2, \dots, k-1\}$.

QoS 的目标就是选择一条从源节点 v_1 到目标节点 v_k 的可行路径 P , 使其满足业务 QoS 要求. 例如, 图 1 中从节点 a 到节点 c 的路径有 $(a,b,e,c), (a,d,c), (a,e,c)$ 等.

2.2 QoS 度量

QoS 要求需要通过可测量的 QoS 度量来实现. 常用的几种 QoS 度量主要包括可用带宽、端到端延迟、分组丢失率、抖动和花费, 不同的度量具有不同的性质. 按照这些性质, QoS 度量可以分为可加性度量、可乘性度量和最小性度量 3 类. 对于图 G 中的路径 $P=(v_1, v_2, \dots, v_s)$, 若 $e_{i,i+1} \in P$ (其中 $i=1,2, \dots, s-1$), 将 $e_{i,i+1}$ 的第 j 个属性记为 $w_{i,i+1}^j$ 或 $w^j(e_{i,i+1})$, 整个路径 P 的第 j 个属性记为 w_P^j , 则以上 3 类度量的定义如下^[10]:

定义 7(可加性度量). 若 $w_P^j = \sum_{i=1}^{s-1} w_{i,i+1}^j$, 则称路径 P 的第 j 个属性为可加性度量.

可加性度量包括延迟、抖动、花费、转发跳数(hop count)等. 例如, 一条路径 P 的延迟为从源到目的地的所有链路延迟的总和. 如图 1 中路径 (a,b,e,c) 的延迟总和为 3, 花费为 18.

定义 8(可乘性度量). 若 $w_P^j = \prod_{i=1}^{s-1} w_{i,i+1}^j$, 则称路径 P 的第 j 个属性为可乘性度量.

例如, 分组丢失率为可加乘度量, 其中链路丢失率 $0 \leq w_{i,i+1}^j \leq 1$. 在求解有关可乘性度量的过程中, 可以类比如参照可加性度量的有关求解方法^[11].

定义 9(最小性度量). 若 $w_P^j = \min_{i=1,2, \dots, s-1} \{w_{i,i+1}^j\}$, 则称路径 P 的第 j 个属性为最小性度量.

例如, 带宽为最小性度量, 即路径带宽为路径上瓶颈链路的带宽. 如图 1 中路径 (a,b,e,c) 的带宽为 1 (其中链路 (b,e) 为带宽瓶颈). 对于求解 $w_P^j = \max_{i=1,2, \dots, s-1} \{w_{i,i+1}^j\}$ 的问题可转化为 $-w_P^j = \min_{i=1,2, \dots, s-1} \{-w_{i,i+1}^j\}$ 问题. 对于不同类型的 QoS 度量以及多个 QoS 度量的组合, 其计算的代价是不同的. 例如, 求解多重最小性度量的组合可以在多项式时间内完成, 而多重可加性度量的组合通常不能在多项式时间内完成.

3 单播 QoS 算法

按照所求解的问题类型和求解方法, QoS 算法可以分成以下几类: 多项式非启发类、伪多项式非启发类、探测类、限定 QoS 度量类、路径子空间搜索类、QoS 度量相关类、花费函数类和概率求解类. 其中概率求解类的一个主要方面是对信息陈旧性和算法有效性的考虑, 我们将在第 4 节加以分析.

3.1 多项式非启发类

多项式非启发类算法针对的并不是典型的 QoS 问题, 而是多项式可解的问题, 但是这类问题及其求解思路是 QoS 算法的基础. Wang 和 Crowcroft 使用 Dijkstra 最短路径树算法实现了带宽延迟受限的源路由求解^[12]. 首先在网络拓扑图中将带宽不足要求的链路剪除掉, 然后再以延迟为关键字使用最短路径树算法计算, 这样求得的路径满足带宽约束并具有最短延迟. 此外, 他们对最短最宽路径提出了分布式预计算方法. 两个节点间的最宽路径是指在这两个节点之间所有的路径中, 瓶颈带宽最大的路径^[12,13]. 最短最宽路径(shortest-widest path)是指在多条最宽路径中最短的路径. 算法中每个节点通过链路状态协议维护全局状态, 并使用扩展 Bellman-Ford 算法^[14](或 Dijkstra 算法^[15])为每个可能的连接目的地预先计算出下一跳, 这样, 在连接请求时只需简单地查找路由表. 这个算法在各个节点网络状态信息一致的情况下可以保证无回路.

3.2 QoS 度量相关

如果所有的 QoS 度量都与某一类度量相关, 那么 QoS 问题也可在多项式时间内求解. 基于这种思想, Ma-steenkiste 证明了当网络中使用一类加权公平队列(WFQ)(如调度算法^[16])时, 端到端延迟、抖动、队列长度等都是带宽的函数, 而不再彼此独立. 这样, 原来多约束的 NPC 问题就有可能简化^[17]. 考虑到这些参数的相关性以后, 通过扩展的 Bellman-Ford 算法在多项式时间内就可以通过源路由策略求解. Orda 对基于速率调度的

QoSR 算法做了较为全面的研究^[18].每个节点通过广播可用速率代替传统的可用带宽等状态,将延迟保证映射为速率保证.通过执行至多 M 次标准最短路径树算法,能够找到端到端延迟受限的路径,并能够满足抖动等其他方面的约束^[19].但是这些分析对传输延迟是无效的,因此往往限定在高速网络中^[20].

3.3 探测法

探测法的思想是从源节点开始,通过逐个询问其他节点,逐步逼近并到达目标节点.算法通常基于广播和多路路由的机制,在多条路径中寻找可行路径;在询问抵达目标节点之前不能断定可行路径.为了避免回路和扩散重复信息,节点需要记录大量探测数据.此外,探测法需要较长的路由时间,但它可以有效地与资源预留相结合.在 Cidon 等人提出的分布式多路路由算法^[21]中,每个节点维护全局状态,根据 QoS 请求,首先找到多条花费合理的路径,控制信息(资源预留)并行地通过这些路径从源发送向目的节点,直到抵达目的节点时完成连接的建立.在多条路径上同时预留资源可能会加快路由速度,但大大增加了网络的动态变化性.该算法的一些变体着重于研究如何折衷路由时间和路径优化.

Shin 和 Chou 设计了延迟受限的分布式算法^[22].每个节点不需要保存全局状态,而通过广播方式从源向目的节点传送累加了延迟的路由信息.中间节点在收到路由信息时,首先累加自己的延迟,如果仍然小于规定的延迟,则再转发给其他邻居.但是收到的路由信息必须满足以下两个条件之一:(1)它是第一次收到该信息;(2)信息中累计的延迟比该节点原来收到的要小.路由信息的扩散路径就是延迟受限的可行路径.若给路由信息设置适当的优先级,并使用特定的调度算法,则可以保证在每条链路上最多发送一次这种路由信息.

为了降低上述 Shin-Chou 算法的通信开销,Chen 和 Nahrstedt 提出了一种分布式路由的框架^[23]:(1)根据到目的节点的拓扑距离,探测只在一个有选择的子集上转发;(2)限制路径长度并在探测失败时动态地增加这个长度.这种算法实际上是在路由时间和通信开销之间选择的折衷,因为探测失败时,会增加路由时间和通信开销.如果在每个节点保存了全局信息(即便不准确),并为源节点赋予一定的票(ticket)数,每个探测至少需要具有一票,则上述算法可以获得更好的性能^[24].

3.4 扩展距离向量算法

扩展 Bellman-Ford 算法(EBFA)就是一种典型的用于求解多重受限 QoSR 问题的伪多项式非启发类算法^[25].由于很多求解该问题的启发式算法都是基于 EBFA 的^[26,27],因此这里首先介绍 EBFA.

定义 10(路径偏序关系“ $<$ ”).假设所有的约束都是路径约束,所有的链路度量都是可加性度量,即 $w_p^j = \sum_{i=1}^{k-1} w_{i,i+1}^j$,路径 P 的度量矢量为 $\vec{w}_p = (w_p^1, w_p^2, \dots, w_p^k)$,则 $\vec{w}_p < \vec{w}_q$,当且仅当对 $\forall j \in \{1, 2, \dots, k\}$ 有 $w_p^j < w_q^j$.

类似地,可以定义偏序关系“ $>$ ”和“ $=$ ”.偏序关系中并不是任意两个元素 \vec{w}_p 和 \vec{w}_q 都必然存在“ $<$ ”、“ $>$ ”和“ $=$ ”3种关系中的一种,例如, $\vec{w}_p = (1, 2, 3)$ 与 $\vec{w}_q = (3, 2, 1)$ 之间不存在其中的任何一种关系.

定义 11(最优路径).设 P 是从源节点到目标节点的一条最优路径,如果不存在从源节点到目标节点的路径 Q 使得 $\vec{w}_q < \vec{w}_p$,则称 P 为一条最优路径.

偏序关系造成两个节点之间的最优路径可能不惟一,对不同的 QoS 约束,可能具有不同的最优路径满足该约束;而对任意 QoS 约束,若存在一条可行路径,则必存在最优路径满足该 QoS 约束. EBFA 中每个节点 u 维护一个集合 $PATH(u)$,记录了从源节点 src 到 u 的所有最优路径.类似于距离向量的 RIP 协议,每个节点 u 将自己计算出的集合 $PATH(u)$ 发送给所有的邻居,而每个节点 u 则依据所收到的信息来计算 $PATH(u)$.由于 $|PATH(u)|$ 与 $|N|$ 和 $|E|$ 成指数关系,因此 EBFA 的时间和空间复杂度都是指数级的.

3.5 限定 QoS 度量

如果将除了一类之外的所有 QoS 度量都限定在一个有限集上,则 QoSR 问题是多项式可解的^[26,28].基于这种思想, Yuan 和 Liu 在 EBFA 的基础上设计了粒度受限的启发式算法^[27],通过限制 $PATH(u)$ 中最优路径的数目达到减小复杂度的目的.算法对于 k 约束问题,将 $(k-1)$ 种度量映射到 $(k-1)$ 个有限集中.设 X^i 表示 w^i 可能的取值

个数,则相应的算法复杂度为 $O\left(|N| \cdot |E| \prod_{i=2}^k X^i\right)$.

Chen 和 Nahrstedt 设计了多重路径受限的源路由算法^[28].算法的思想是将 QoS 度量的取值范围 R 或 Z 映射到一个有限集合 C ,从而保证多项式可解.以花费 Cost 为例,设 C 为最大花费约束, x 为一个小于 C 的正整数.算法将每个链路的花费映射到集合 $C' = \{1, 2, \dots, x+1\}$ 中,其中属于区间 $[0, C]$ 的花费映射到 $\{1, 2, \dots, x\}$ 中,属于区间 $(C, +\infty)$ 的花费映射为 $(x+1)$;花费 C 映射为 x .可以证明,满足新问题的可行路径在原来的问题中也有一条可行路径,而新问题可以通过扩展 Dijkstra 算法或者扩展 Bellman-Ford 算法,在多项式时间内求解.算法的性能与 x 密切相关:随着 x 的增大,找到可行路径的概率也增大,但同时引入了更大的开销.因此,可以通过选取 x 来调节该算法的性能.

对于延迟受限的最小花费问题(DCLC),存在一些伪多项式时间的近似算法^[29,30].对于任意 $\epsilon > 0$,存在一个多项式时间的算法能够找到一条路径,在满足延迟受限的同时,其花费不超过最小花费的 $(1+\epsilon)$ 倍.由于这些算法的计算复杂度与 ϵ 有关,因此被称为伪多项式算法.例如,Lorenz 算法的复杂度为 $O\left(nm \log n \log \log n + \frac{nm}{\epsilon}\right)$ ^[30],其中 m, n 分别为网络的节点数和边数.Orda 等人设计了通过量化花费函数的预计算算法^[5],该算法针对多重可加性约束的路径优化问题,基于通用 Bellman-Ford 算法,将计算复杂度减小到 $O\left(\frac{1}{\epsilon} H m \log C\right)$,其中 H 为最长路径的跳数, C 为路径花费的上限.

3.6 路径子空间搜索

路径子空间搜索是指通过启发式算法,直接限制路径数量,从而在路径集合的子空间中搜索可行路径的方法.这种方法在源节点或者中间节点就可以计算出可行路径,而不需要像探测法那样一直询问到目的节点,因而具有较高的启发性.Yuan 和 Liu 设计了路径受限的多重路径受限的分布式算法^[27].他们在 EBFA 中直接限制 $|PATH|$ 的大小从而减小算法复杂度.令 X 为 $|PATH|$ 最大数目,在原 EBFA 算法向 $PATH$ 加入一条最优路径之前,判断是否满足 $|PATH| < X$,若不满足,则不加入该最短路径.通过这样简单的限制,能够保证算法复杂度为 $O(X^2|N| \cdot |E|)$.他们通过试验数据说明了路径受限算法性能大大优于粒度受限的算法.

Salama 等人对复杂度为 NPC 的 DCLC 问题设计了分布式启发算法^[31].与 EBFA 类似,为每个可能的目的网络,每个节点保存一个花费向量和一个延迟向量.为建立 DCLC 可行路径,源节点向目的节点发送一个控制信息,已经部分建立的可行路径的末端节点 i 收到该信息后,将信息继续向目标节点传递.设 (i, j) 是从 i 到目标节点的最小花费路径中下一跳的链路, (i, k) 为最小延迟路径中下一跳的链路,则传递规则是:只要加上从节点 j 开始的最小延迟路径仍然满足 QoS 约束,节点 i 就选择 (i, j) ;否则选择 (i, k) .由于算法本质上可能产生回路,因此他们设计了检测回路以及遇到回路时回退的方法.

3.7 花费函数

为了降低 QoSR 算法的复杂度并提供更好的 QoS 支持,一种可能的方式是通过研究花费函数,从而保证所计算的最小花费路径能够满足 QoS 需求.例如,通过将链路花费与利用率结合,能够降低连接阻塞概率,抑制路由抖动^[8].

Korkmaz 等人设计了一种在多个可加性约束的条件下求解优化路径的通用算法^[11].算法首先构造路径花费函数 $g_I(P) = \left(\frac{w^1(P)}{c^1}\right)^I + \dots + \left(\frac{w^k(P)}{c^k}\right)^I$,其中 $w^i(P)$ 为路径 P 的第 i 维属性, c^i 为一个业务流所对应的 QoS 要求.最小化花费函数所找到的路径,为多重受限 QoSR 问题提供了一个连续的近似频谱:从 $I=1$ 的线性近似,到 $I=\infty$ 时的渐近近似.因此,原来的多约束 QoSR 问题转化为最小化 $\lim_{I \rightarrow \infty} g_I(P)$ 所对应的路径问题.

Ergun 等人设计了一种启发式算法,使得一条链路基于不同的花费能够提供不同的延迟^[32].该算法的思想是将传统的 DCLC 问题分解为两个问题:(1) 使用传统方法找到一条可行路径;(2) 将整个路径的延迟分割成

路径中每个链路延迟的约束,从而得到路径的最小花费.文中对这两个问题分别设计了 ϵ 优化的近似算法.对于分割求解的方式,Lorenz 等人研究了花费函数为连续凸函数时的分割方式^[33];Raz 等人研究了花费函数为离散函数时的分割方式^[29].Fortz 等人研究了如何根据测量结果设置路径权值,从而使 OSPF 协议所计算的最短路径能够平衡负载并满足 QoS 需求^[34].

Juttner 等人基于拉格朗日松弛法设计了 DCLC 的源路由求解算法^[35].算法首先构造路径花费函数 $g_I(P) = c(P) + I \cdot d(P)$,其中 c 为原来的花费, $d(P)$ 为路径 P 的延迟, $I \geq 0$.然后以 $g_I(P)$ 为关键字使用 Dijkstra 算法计算最短路径 P ,如果 $I = 0$ 且同时 P 满足延迟要求,则 P 为原 DCLC 问题的最优解;如果 P 不能满足要求,则可通过增大 I 而加大对 $g_I(P)$ 的惩罚力度,求得满足延迟要求的路径.令 $L(I) = \min\{g_I(P) \mid \forall P \in P(s, t)\} - ID$,其中 D 为延迟要求,则对 $\forall I \geq 0$ 有 $L(I)$ 为原 DCLC 问题的花费下界.算法通过计算特定的 I 从而最大化 $L(I)$,得到花费下界,并给出一条可行路径.

3.8 算法小结

为了降低复杂性和提高性能,有些 QoSR 算法将以上几种思想结合起来.例如,将花费函数与概率模型结合,以概率相同的方式分割路径花费^[29];将路径子空间搜索与探测法结合,在搜索失败的情况下回退探测^[31];QoS 度量相关与花费函数的 ϵ 优化结合^[5]等.此外,多类算法中都使用到 ϵ 优化的方法.表 1 给出了单播算法的比较,每个算法以第一作者命名,状态信息表示计算 QoSR 的节点需要收集和维持的网络状态信息的类别,其中部分算法将在第 4 节进行分析.

4 算法有效性分析

一个 QoSR 算法首先要在理论上保证避免回路(或者能够检测排除回路),然后还需要进一步考虑实际网络的情况,包括网络模型、路由信息是否陈旧等问题,才能够保证算法是实际可用的.

4.1 路由回路问题

在 Salama 等人所提出的算法^[31]中,由于控制信息可能在最小延迟路径和最小花费路径之间不断变化,因此可能造成回路,这就属于算法本身理论上的缺陷.在文献[31]中采用检测回退的方法来解决这个问题:当控制信息对同一节点访问两次时,说明算法产生了回路.这时回退路径选择过程,直到找到一个原来在其后使用最小花费路径的节点,将该节点原来沿最小花费路径传递改为沿最小延迟路径传递.可以证明,在所有节点网络信息一致的前提下,这种机制能够消除回路^[31].

Sobrinho 通过代数学方法详细研究了分布式 QoSR 中路由回路和最短路径问题^[36].文献[36]定义了链路度量的二元操作和一个抽象的全序关系以及在这个关系中路径度量的保序性.其中保序性是指,对任意两条路径 P, Q ,如果给这两条路径 P, Q 同时追加上路 L 时,在序关系中, P 和 Q 的关系与 $(P+L)$ 和 $(Q+L)$ 的序关系相同,也就是说,追加过程不改变原来路径度量的序关系.该文证明了路径中 QoS 约束的度量函数的保序性是使用 Dijkstra 算法计算最短路径的充要条件;否则,计算过程中有可能产生回路.

4.2 陈旧信息的影响

在研究 QoSR 问题时,通常假设每个节点已知本地状态信息,并通过协议交互获得全局状态.然而,这些状态信息是不精确的^[37],其原因包括以下 4 个方面:(1) 在传播本地状态的过程中,存在不可忽略的网络传输延迟和协议交互延迟;(2) 网络状态是不断更新的(如可用带宽、延迟),考虑到传输网络状态信息的开销,所以路由更新间隔不可能无限小(定期更新的时间间隔或者触发更新的触发门限);(3) 链路度量的粒度往往在路由计算开销和阻塞概率之间折衷;(4) 层次化的网络结构造成信息压缩.对于前两方面所造成的网络状态信息在时间上的不精确性,我们称为陈旧性.为了讨论陈旧信息对算法性能的影响,需要区分路由失败和建连失败的概念.路由失败是指根据所保存的网络状态信息和新的 QoS 连接请求,无法计算出可行路径.建连失败是指,虽然算法计算出了可行路径,但是实际上由于信息的陈旧性,该路径已经不能够支持这个 QoS 连接.建连失败通常是沿着这条不可行路径预留资源时发生的,因此导致了额外开销以及对其他 QoS 请求的阻碍.

Shaikh 等人发现若信息的陈旧性小,则计算 DCLC 路径时应该使用剪枝的机制;否则不应使用剪枝机制^[8]。此外,随着陈旧性的增加,路由失败的概率减小而建连失败的概率迅速增加。为了避免陈旧性,一种方式是基于本地状态路由^[23,38],如 Shin 等人设计的分布式探测算法^[22]。Nelakuditi 等人设计了网络边缘路由器根据业务流阻塞的统计数据计算可行路径的方法,能够将业务流均衡地分配到网络上^[38]。

Table 1 The comparison of Unicast QoSR algorithms

表 1 单播服务质量路由算法的比较

Algorithm	Problem to solve	Routing strategy	Computation complexity	State information	Heuristic class
EBFA ^[25]	Multiple path-constrained	Distributed routing	Pseudo-Polynomial	Global	Non-heuristic
Cidon ^[21]	General problem	Distributed routing	$O(e)$	Global	Detection
Shin ^[22]	Delay-Constrained	Distributed routing	$O(e)$	Local	Detection
Chen ^[28]	Bandwidth-Constrained	Source routing	$O(xne)$	Global	Detection
Yuan ^[27]	Multiple path-constrained	Distributed routing	$O\left(ne\prod_{i=2}^k X^i\right)$	Global	QoS metric-constrained
Chen ^[23]	General problem	Distributed routing	$O(e)$	Local	QoS metric-constrained
Orda ^[5]	Multiple path-constrained optimal	Distributed routing	$O(\text{Polynomial}/e)$	Global	QoS metric-constrained e -optimal ⁽²¹⁾
Yuan ^[27]	Multiple path-constrained	Distributed routing	$O(X^2ne)$	Global	Path subspace search ⁽²³⁾
Salama ^[31]	Delay-Constrained least-cost ⁽²³⁾	Distributed routing	$O(n^3)$	Global	Path subspace search
Ma ^[17]	Multiple link-constrained ⁽²⁰⁾	Source routing	$O(kne)$	Global	QoS metrics related ⁽²⁵⁾
Ma ^[17]	Bandwidth-Constrained	Source routing	$O(n\log n+e)$	Global	QoS metrics related
Orda ^[18]	Delay-Constrained least-cost	Distributed routing	Polynomial ⁽²⁶⁾	Global	QoS metrics related
Korkmaz ^[11]	Multiple path-constrained optimal path	Source routing	Polynomial	Global	Particular cost function ⁽²⁷⁾
Ergun ^[32]	Delay-Constrained optimal ⁽²⁸⁾	Source routing	$O(\text{Polynomial}/e)$	Global	Cost function partitioned e -optimal ⁽²⁹⁾
Raz ^[29]	Multiple path-constrained	Source routing	$O\left(\frac{1}{e}en^3\log e\right)$	Global	Cost function partitioned e -optimal ⁽³⁰⁾
Juttner ^[35]	Delay-Constrained least-cost	Source routing	$O(m^2\log^4 m)$	Global	Lagrange relaxation ⁽³¹⁾
Wang ^[12]	Bandwidth-Constrained	Source routing	$O(n\log n+e)$	Global	Non-heuristic
Wang ^[12]	Optimal bandwidth ⁽³²⁾	Distributed routing	$O(ne)$	Global	Non-heuristic
Guerin ^[39]	Bandwidth-Constrained	Source routing	$O(n\log n+e)$	Global (imprecise) ⁽³³⁾	Probability solution ⁽³⁴⁾
Guerin ^[39]	Delay-Constrained	Source routing	Polynomial	Global (imprecise)	Probability solution
Lorenz ^[40]	Delay-Constrained	Source routing	$O(De)$	Global (imprecise)	Probability solution e -optimal ⁽³⁵⁾

n is the node number, m is the edge number, X is the number of optimal paths, k is the constraint number of multiple path constraints, and D is the upper bound of delay (delay is a positive integer)⁽³⁶⁾。

算法名称, 所解决的路由问题, 路由策略, 时间复杂度, 状态信息, 启发式类别, 多重路径受限, 分布式路由, 伪多项式, 全局, 非启发式, 一般问题, 探测法, 延迟受限, 本地, 带宽受限, 源路由, 限定 QoS 度量, 多重路径受限优化, $O(\text{polynomial}/e)$,⁽²¹⁾限定 QoS 度量 e 优化,⁽²²⁾路径子空间搜索,⁽²³⁾延迟受限最小花费,⁽²⁴⁾多重路径链路受限,⁽²⁵⁾QoS 度量相关,⁽²⁶⁾多项式,⁽²⁷⁾特定花费函数,⁽²⁸⁾延迟受限优化,⁽²⁹⁾分割花费函数 e 优化,⁽³⁰⁾分割离散花费函数 e 优化,⁽³¹⁾拉格朗日松弛,⁽³²⁾带宽优化,⁽³³⁾不精确全局,⁽³⁴⁾概率求解,⁽³⁵⁾概率求解 e 优化,⁽³⁶⁾ n 为节点数, m 为边数, X 为最优路径的数目, k 为多重路径约束的约束个数, D 为延迟的上限(延迟为正整数)。

4.3 网络模型的影响

研究者们往往通过模拟试验对所设计的 QoSR 算法进行性能评价,然而不同的模型可能导致不同的评价结果,而模型对算法性能的影响还可能进一步反馈到算法的改进中。由于 Internet 缺少典型的拓扑结构和业务流,因此网络模型的选择非常困难^[41,42]。

常用的拓扑模型包括^[42]:(1) 众所周知(Well-Known)的拓扑结构,如 MCI 骨干网^[27]、ARPAnet 等;(2) 随机生成的拓扑结构,如平面随机图^[35]、Transit-Stub 模型^[7]、层次化模型等;(3) 特定规则的结构,如网格结构^[27]、完全图等。Shaikh 等人通过模拟试验^[8]说明,在链路密集的网络中,由于网络直径相对较小,以及对 QoS 请求的连接通常可以找到较短的可行路径,因此应该减少网络状态更新的次数(增大更新周期或触发间隔)。同时,由于链路密集造成的每次广播开销很大,应该考虑使用支撑树的方式转发状态信息。

业务流的模型应该包括 QoS 连接请求的到达方式、QoS 连接的持续时间、QoS 参数的分布等。Shaikh 等人的研究结果表明^[8],应该仅对持续时间较长的连接采用 QoSR 寻找可行路径,而对于短时间的业务应该基于事先提供静态路由进行转发。比较重尾(long-tailed)分布和指数分布的连接时间,当平均连接时间相同时,重尾分布具有更多的短期连接,因此更有必要在网络边缘检测短时间连接,以便仅对长时间连接提供 QoS 路由。此外,在带宽受限的算法中,如果考虑了信息的陈旧性,由于突发到达的(burst)连接可以认为是一个要求更大带宽的连接请求,这样会增加链路状态的波动以及对不合理资源分配的敏感性,因此应避免剪枝造成的非最短可行路径。

4.4 基于概率求解

由于信息的陈旧性可能导致建连失败,因此可以考虑求解以最大概率满足 QoS 约束的可行路径,这类问题往往基于概率求解。在源路由的链路状态模型下,Guerin 和 Orda 研究了不精确的状态信息下,带宽和延迟分别受限的 QoSR 问题^[39]。源节点对网络中的每条链路 l 具有概率分布函数 $p_l(w)$,表示链路 l 具有剩余 w 个单位带宽的概率。其中 $w \in [0, 1, \dots, c_l]$, c_l 为链路的容量。带宽受限路由的任务就是寻找一条路径,能够以最大概率满足新的 QoS 连接请求所要求的 x 个单位带宽。这样,以 $-\log p_l(x)$ 为链路 l 的度量,使用最短路径树算法即可求解。延迟受限情况与此类似,设节点对网络中的每条链路 l 具有概率分布函数 $p_l(d)$,表示链路 l 延迟为 d 个单位的概率。Lorenz 为此设计了将全局约束转化为局部约束的启发式算法:将端到端延迟沿着路径分割成链路延迟约束,并且保持所有的链路具有相同的概率,满足分割后的链路延迟约束^[40]。

Zhang 和 Liu 设计了基于蚂蚁算法的通用 QoSR 算法,可以包含对带宽、延迟、丢失率抖动等参数的考虑^[43]。蚂蚁根据不同路径上信息素的多少,以概率选择前进的路径,同时沿着路径行走时在路径上留下信息素。据此,该算法将 QoS 业务类型分成 M 类,每类分别应用蚂蚁寻路的方法。在建立 QoS 连接的过程中,业务类型 d 在 r 点以一定的概率 $r_d(r, s)$ 选择下一跳 s 点,同时更改概率值 $r_d(r, s)$,从而达到全局优化。

5 总结与展望

5.1 总结

服务质量路由(QoSR)是未来互联网络的一个核心功能,其主要目标包括两个:(1) 为每个接受的 QoS 业务流提供服务质量保证;(2) 达到网络全局资源的最佳利用。前者要求在多约束条件下计算可行路径;后者则要求有多条可行路径中进一步优化。优化的方式通常是首先设计花费(cost)函数,然后求解函数值最优的可行路径。然而,通常多约束下求解(优化的)可行路径属于 NP 完全问题,不能在多项式时间内精确求解,为此研究者们设计了很多启发式算法或近似算法。这些算法还具有以下 3 方面的不足:(1) 计算复杂度过高而不能在网络中实际应用;(2) 算法性能过低而找不到实际存在的可行路径;(3) 大部分算法只是针对 QoSR 问题中某些特殊情况的。

目前研究的几类热点算法仍然存在以下问题:多项式非启发类的算法只能解决单一可加性约束(优化)的问题;基于 QoS 度量相关的算法只能应用于基于速率调度的特定网络环境下,而且不支持对传输延迟的考虑;伪多项式的 EBFA 能够精确求解多重受限 QoSR 问题,但由于其指数级别的复杂度因而也是不可取的;基于探测的启发式算法路由时间长、通信开销大,还可能阻塞其他业务以及中间节点需要保存大量的状态;限定 QoS 度量的算法复杂度与 QoS 度量的粒度关系密切;路径子空间搜索的算法需要较好的启发函数,它以增大路由失败概率和降低路由性能为代价,减小了计算开销;基于特定花费函数的算法难以实现花费的可扩充性,同时,花费的计算代价较高;基于概率求解的算法不易确定和传播链路概率值,并且需要在花费和最大概率之间折衷。此外,很多算法使用 ϵ 优化的方式,无形中增加了业务的 QoS 要求。

5.2 展望

随着网络和应用业务的快速发展,QoSR 日益成为网络研究的核心问题之一。目前的研究成果表明,在一定区域内部通过合理地配置基于链路状态的 QoSR 协议,为支持 QoS 所引入的开销是完全可能接受的^[4],然而 QoSR 领域在以下几个方面还有待进一步深入研究。

5.2.1 可扩展性

通过网络状态聚集能够以对数缩减信息量,相应的分层路由可以通过限制每层内节点的数量使用源路由方式,从而很好地解决可扩展性问题.但这同时又引入了新的问题:如何聚集状态信息、如何基于聚集状态设计良好的启发式算法.目前所设计的状态信息聚集方法往往会丢失大量的可用信息,严重影响了 QoS 算法的性能;同时,所设计的启发式算法也往往是基于全局状态源路由策略的.此外,由于很多 QoS 问题的精确求解复杂度都是 NPC 的,因此需要设计快速、有效的启发式算法.

5.2.2 信息陈旧性

状态信息的陈旧性是实际网络中不可避免的,并且会对 QoS 算法的性能和有效性造成显著影响.然而对这种陈旧性的分析涉及到复杂的随机数学模型,相关的讨论往往基于模拟试验数据,而缺少定量的理论分析.此外,通常的算法只是停留在理论设计和分析上,缺乏对信息陈旧情况下算法实际性能的考虑.因此,可以考虑基于概率模型求解,寻找以最大概率满足 QoS 约束的可行路径,从而有效地减少建连失败对网络造成的额外负担.

5.2.3 多路路由与重路由

在多条可能的路径上同时探测可行路径的方法中,如何与资源预留相结合尚无定论^[21].此外,网络提供多条从源到目的的路径,并将这些路径并行复用起来(如增加带宽)而对用户业务透明,这是一种多路复用的路由方式.然而,其主要问题是多条路径之间如何同步,以及如何避免分组的延迟抖动、乱序等^[44].由于网络资源和可行路径分配的不合理,因而在某些情况下需要重新路由.重路由可在网络资源不足时进行,能够有效地重新分配网络资源,但由于存在状态保存、同步和开销等问题,使重路由变得很困难.

5.2.4 网络模型

不同的网络拓扑结构和 QoS 业务流往往对 QoS 算法的性能造成很大的影响,包括网络直径、节点互联粒度、网络资源分布、路径长度分布、连接持续时间分布、QoS 需求分布等参数.因此,QoS 需要在以下两个方面与网络模型结合起来:(1) 评价 QoS 算法和协议的性能、有效性;(2) 针对实际网络设计高效的 QoS 算法和协议,如对持续时间短的连接使用静态路由^[8].由于 Internet 缺乏“典型的”网络拓扑结构和业务流,因此,如何使用适当的网络模型来指导 QoS 的研究,将是进一步的研究方向.

5.2.5 与其他网络组件结合

今后的网络应该是 QoS 和尽力发送相结合的^[45].网络的路由目标是最大化资源效用,这包括尽可能地接纳更多的 QoS 连接请求,同时最大化尽力发送业务的吞吐量和相应速度.由于这两者是矛盾的,因此二者融入的过程会产生很多问题.例如,在有资源预留的链路空闲时,没有资源预留的链路却可能由于尽力发送业务而造成拥塞.这种拥塞的链路仍然有可能因为尚未预留资源而接受 QoS 请求.此外,QoS 算法必须与其他网络组件结合才能提供 QoS 保证,包括状态收集、资源预留、分组调度等,因此可以考虑这种结合过程对 QoS 算法和其他网络组件的简化.

References:

- [1] Crawley, E., Nair, R., Rajagopalan, B. *et al.* A framework for QoS-based routing in the Internet. RFC 2386, 1998.
- [2] Lee, W.C., Hluchiy, M.G., Humblet, P.A. Routing subject to quality of service constraints integrated communication networks. *IEEE Network*, 1995,9(4):46~55.
- [3] Quality of service—glossary of terms. QoS Forum. 1999. <http://www.qosforum.com>.
- [4] Apostolopoulos, G., Guerin, R., Kamat, S. Implementation and performance measurements of QoS routing extensions to OSPF. In: Doshi, B., ed. *Proceedings of the IEEE INFOCOM'99*. New York, NY: IEEE Communication Society, 1999. 680~688.
- [5] Orda, A., Sprintson, A. QoS routing: the precomputation perspective. In: Sidi, M., ed. *Proceedings of the IEEE INFOCOM 2000*. Israel: IEEE Communication Society, 2000. 128~136.
- [6] Garey, M.S., Johnson, D.S. *Computers and intractability: a guide to the theory of NP-completeness*. Oxford: Freeman, W.H., 1979.
- [7] Hao, F., Zegura, E.W. On scalable QoS routing: performance evaluation of topology aggregation. In: Sidi, M., ed. *Proceedings of the IEEE INFOCOM 2000*. IEEE Communication Society, 2000. 147~156.
- [8] Shaikh, A., Rexford, J., Shin, K.G. Evaluating the impact of stale link state on quality-of-service routing. *IEEE/ACM Transactions on Networking*, 2001,9(2):162~176.