

多 QoS 约束的层次多播路由算法框架^{*})

颜 昕 李腊元

(武汉理工大学计算机科学系 武汉 430063)

摘 要 为了解决网络路由的扩展性问题,大型网络通常被划分成若干个不同的域。拓扑聚集是对这些域的拓扑状态信息进行汇总的过程。在拓扑聚集的基础上, QoS 层次多播路由算法用来构造满足 QoS 要求的域间多播树。现有的 QoS 层次多播路由算法在其拓扑聚集和路径计算的过程中都只考虑了存在两个 QoS 特征值的情况。本文提出了一种具有多 QoS 约束的层次多播路由算法框架 HMRMQ (Hierarchical Multicast Routing with Multiple QoS constraints), 此算法框架不仅为基于多 QoS 特征值的拓扑状态聚集和状态信息表示提供了新的方法, 而且提出了一种适应于多 QoS 约束的层次多播路由新算法。我们提出的状态信息表示法和拓扑聚集算法都具有很好的扩展性, 分布式的路由算法也便于某些安全性策略的实施。理论分析和实验结果不仅证明了 HMRMQ 的正确性和有效性, 同时也表明了 HMRMQ 在网络路由的扩展性、路由成功率、网络代价以及报文负载等方面都具有良好的性能。

关键词 拓扑聚集, 层次路由, 多 QoS 约束, QoS 多播路由

The Hierarchical Multicast Routing with Multiple QoS Constraints

YAN Xin LI La-Yuan

(Department of Computer Science, Wuhan University of Technology, Wuhan 430063)

Abstract In order to deal with the scalability problem of network routing, large networks are often structured hierarchically by grouping nodes into different domains. Topology aggregation is the process of summarizing the topological state information of the domains. Based on topology aggregation, QoS-aware hierarchical multicast routing algorithms are used to construct a multicast tree satisfying QoS requirements among network domains. The existing QoS-aware hierarchical multicast routing algorithms don't take into account more than two QoS metrics during their topology aggregations and route computations. In this paper, we propose an algorithm framework of hierarchical multicast routing with multiple QoS constraints, called HMRMQ, which provides both new schemes for topological state aggregation and state information representation based on multiple QoS metrics and a novel hierarchical multicast routing algorithm with multiple QoS constraints. The schemes of representing and aggregating multiple QoS metrics are very scalable, and the distributed routing algorithm is facile to implement certain security policies. Our theoretical analysis and simulation results not only prove the correctness and availability of HMRMQ, but also show that HMRMQ achieves the routing scalability and the improved routing performance in terms of routing success ratio, average network cost and average message overhead.

Keywords Topology aggregation, Hierarchical routing, Multiple QoS constraints, QoS-aware multicast routing

1 引言

随着 Internet 规模的迅速扩大,网络路由面临着复杂度过高的问题。在大规模网络中实现路由计算的主要困难在于:一是因路由表空间的扩大导致了路由计算的复杂度增加;另一方面,由于路由计算都是以动态网络状态信息的频繁更新为基础的,而大量状态信息的产生、传输和处理都会消耗大量的网络带宽和路由器资源,也增加了路由计算的复杂度。因此,网络路由的扩展性问题已成为新的研究热点。

为了解决路由的扩展性问题,Internet 被划分成若干个不同的域或 AS (Autonomous System), 并且路由器和主机都采用层次结构的地址标识,即 IP 地址的第一部分标识 AS, 第二部分标识特定 AS 内的主机或路由器。我们可以采用 PN-NT^[1] 中一般性的定义来描述这种划分网络层次的思想:将网

络中的节点按其地理位置或管理的需要划分成若干个域或 PG (Peer Group), 在每个域中,至少有一个接口与其它域直接相连的节点称为边界节点,而只与本域内节点相连的节点称为内部节点;随后,这些域被抽象成逻辑节点,形成层次化网络的上层,这一处理过程称为拓扑聚集 (Topology Aggregation, TA), 如图 1 所示。网络经分层之后,同一域的状态信息只在本域内扩散,域间所广告的信息是经过聚集之后的状态信息,域的外部节点并不知道该域的内部结构。由此可见,网络层次划分和拓扑聚集(即层次路由)不仅减小了路由表的存储空间(复杂度由 $O(n^2)$ 降为 $O(\log n)$, 这里 n 为节点总数^[2]), 也显著减少了进行路由计算所需要的状态信息量,还可以用来实施某些安全性策略,但同时也引入了网络状态信息的不精确性^[3]。需要说明的是,网络上层的逻辑节点还可以被进一步地划分和聚集,此过程可以递归进行,从而形成

^{*}) 本文研究得到国家自然科学基金(No. 60172035, 90304018)资助。颜 昕 博士研究生, 讲师, 研究领域为高性能计算机网络、活动网络和网络仿真技术;李腊元 教授, 博士生导师, 主要研究领域为高性能计算机网络与协议工程。

多层结构的网络拓扑。为了使问题简化,本文只讨论两层结构。

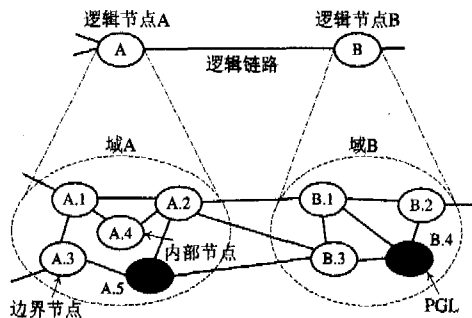


图1 网络层次划分与拓扑聚集

为满足日益增长的实时多媒体业务对网络 QoS 的需求,层次路由算法应提供对 QoS 的支持。QoS 层次路由的目的就是要在层次化的网络中找到满足 QoS 约束条件且代价最小(或最大)的可行性路径或多播树,为此必须解决以下问题:(1)如何实现带 QoS 状态参数(即特征值)的拓扑聚集,尤其是带多个 QoS 状态参数的拓扑聚集;(2)如何表示聚集的网络状态信息;(3)如何利用聚集的状态信息找到满足约束条件的路径或多播树。

目前,有关拓扑聚集和 QoS 多播路由的研究大多都是分开进行的。文[4,5]分别提出了能保证代价扭曲有界和代价无损压缩的拓扑聚集算法,但都只是针对一个可加性 QoS 特征值(代价或延迟)的拓扑聚集。已经证明,存在两个或两个以上不相关、可加性度量的 QoS 路由问题属于 NP-完全问题^[6],QoS 参数的数量和特征(可加性和凹性)不仅决定了多播路由算法,也直接影响到拓扑状态的聚集和状态信息的表示。文[7]提出了一种能实现带宽无损压缩、延迟扭曲有界的生成树聚集算法,但只适用于链路特征对称的网络,不具备通用性。在文[8]中,作者用带附加参数的延迟-带宽数值点来表示每对边界节点间最好路径的 QoS 特征值,以构成包含全部边界节点的完全图(Full-Mesh),但其中面向数据源的星型(Star)拓扑聚集算法对网络负载的类型十分敏感,并不适用于 IP 网络中的路由计算。文[9,10]用延迟-带宽平面内的两条线段来表示 Full-Mesh 图中逻辑链路的 QoS 特征,并在此基础上提出了相应的 Star 拓扑聚集算法和基于票据的单播路由算法,但这种线段表示法难以表示链路的多 QoS 特征,不具备扩展性。

有关 QoS 多播路由的研究目前主要集中在对平面路由的讨论^[11],不适用于大型网络。文[12]提出了一种基于 BGMP 的 QoS 扩展策略,文[13]则提出了 QoS MIC 协议的域间路由方案,文[14,15]分别提出了支持 QoS 的层次多播路由算法,但这些有关 QoS 层次多播路由的研究都只考虑了存在一个 QoS 约束(即带宽受限、延迟或代价最小)的情况。

本文提出了一种具有多 QoS 约束的层次多播路由算法框架 HMRMQ(Hierarchical Multicast Routing with Multiple QoS constraints),此算法框架包括:一种基于多 QoS 状态参数(包括带宽、延迟、代价等)并具有很好扩展性的拓扑聚集算法和一种适应于多 QoS 约束并易于实施某些策略的层次多播路由算法。

2 系统模型

2.1 层次网络模型

定义1 整个网络由一系列的域以及连接这些域的链路构成。设一加权有向连通图 $N=(G, L)$ 表示整个网络,其中 G 是网络中域的集合, L 是域间有向物理链路的集合, $|G|$ 和 $|L|$ 分别表示域的个数及域间的链路数。假设域 $G=(V, B, E)$,其中 V 是域中节点的集合, $B \subset V$ 是边界节点的集合, E 是域内有向物理链路的集合, $|V|$, $|B|$ 和 $|E|$ 分别表示域中的节点数、边界节点数及链路数。不失一般性,只考虑简单图。

定义2 设用 (w, d, c) 表示链路或路径的 QoS 特征值,其中 w, d 和 c 分别是带宽、延迟和代价的大小度量。

定义3 为了标识网络中的任意节点,可用 g_i, v_j (其中 i 和 j 是正整数,且 $0 < i \leq |G|, 0 < j \leq |V|$) 来表示域 g_i 中的节点 v_j 。

2.2 QoS 多播路由模型

定义4 设 R 为整个网络中所有节点的集合, $T(s, M)$ 表示跨域的多播树,其中 $s \in R$ 为一多播组的源节点, $M \subseteq \{R - \{s\}\}$ 为该多播组的目的节点集。

定义5 在多播树 $T(s, M)$ 中,存在下列关系:

$$bandwidth(p(s, t)) = \min\{bandwidth(e), e \in p(s, t)\} \quad (1)$$

$$delay(p(s, t)) = \sum_{e \in p(s, t)} delay(e) + \sum_{n \in p(s, t)} delay(n) \quad (2)$$

$$cost(T(s, M)) = \sum_{e \in T(s, M)} cost(e) + \sum_{n \in T(s, M)} cost(n) \quad (3)$$

其中 $p(s, t)$ 为树上的源节点 s 到目的节点 t 的路径, $n \in R$ 和 $e \in L \cup E$ 分别表示整个层次网络中的节点和链路。

定义6 QoS 层次多播路由问题就是要找到一棵跨域的多播树 $T(s, M)$,能满足下列 QoS 约束条件,且 $cost(T(s, M))$ 为最小。

带宽约束: $bandwidth(p(s, t)) \geq W$

延迟约束: $delay(p(s, t)) \leq D$

其中 W 是带宽约束, D 是延迟约束。

3 拓扑聚集

拓扑聚集是层次网络的下层汇总和压缩本层的拓扑状态信息并将压缩过的状态信息广告到上层网络的过程。TA 过程有效地减少了域间广告的状态信息量,但经过汇总和压缩的状态信息通常会产生一些扭曲和失真,从而影响了利用这些状态信息进行路由计算的精度,因此,要求 TA 算法既能实现有效的信息压缩又要保证较小的信息失真,同时还应具有较低的计算复杂度。现有的 TA 算法主要包括简单节点法、Full-Mesh 算法、Star 算法以及生成树(Spanning Tree)算法。在简单节点算法中,整个域被压缩成一个逻辑节点,节点的直径(通常是最坏的状态参数值)被广告到网络上层^[1];此算法能实现最大的信息压缩(压缩信息的复杂度为 $O(1)$),但信息失真十分严重。Full-Mesh 算法使用每对边界节点间的逻辑链路来构造完全图;此算法不会造成信息失真,具有最好的适应性,但经其压缩过的信息复杂度仍为 $O(|B|^2)$,这里 $|B|$ 为边界节点数。参照 PNNI 的定义,Star 算法(又称复合节点表示法)用边界节点和一个虚拟的核来表示整个域的节点,每个边界节点都与核相连,它们之间的逻辑链路称为 spoke,边界节点之间直接相连的逻辑链路称为 bypass;此算法可以看成是简单节点法与 Full-Mesh 表示法之间的折中(压缩信息的复杂度为 $O(|B|)$)。生成树算法是将 Full-Mesh 图编码成最大(或最小)生成树进行广告并能在接收端还原成 Full-Mesh 图的 TA 算法^[7];此算法能实现凹性状态参数的无损压缩以

及可加性状态参数的扭曲有界(压缩信息的复杂度为 $O(|B|)$),具有很好的整体性能^[16],但仅适用于具有对称特征的链路。

我们提出一种新的拓扑聚集算法-混合 TA 算法。其基本思想是:首先,在每对边界节点间选择最好的物理路径来构造逻辑链路,并把该路径的 QoS 特征值(带宽、延迟和代价)作为逻辑链路的特征参数,从而构造包含全部边界节点的 Full-Mesh 图;然后,对 Full-Mesh 图中具有不同特征的 QoS 状态参数使用不同的聚集方法进行表示和广告,即对带宽等具有凹性特征的状态参数采用最大生成树(Maximum Spanning Tree, MST)算法进行编码,对延迟、代价等具有可加性特征的状态参数则使用 Star 算法进行压缩。混合 TA 算法基于以下的原因和假设:(1)在动态网络中,带宽的变化相比跳数、代价及传输延迟的变化更为频繁;(2)在链路状态更新的策略中,带宽的变化常常作为触发更新的条件;(3)带宽的变化在很大程度上也反映了节点和链路中数据包队列长度的改变;(4)可以假设链路的带宽具有对称性(即无方向性)。参照 MST 和 Star 算法各自的特点,可以看出,对路由计算敏感的、具有对称特征的凹性状态参数使用 MST 算法进行压缩是合适的,同时 Star 算法也能够保证可加性状态参数的扭曲有界^[17]。

3.1 构造 Full-Mesh 图

Full-Mesh 是一个由边界节点构成的完全连通图。为了构造 Full-Mesh 图,我们要在每对边界节点间选择最好的物理路径来构成逻辑链路,但实际上最好路径的选择是十分困难的,尤其在有多个 QoS 参数的情况下,最好的物理路径可能并不存在。例如,图 2(a)表示一个域,域中每条物理链路都有 3 个 QoS 状态参数,分别是带宽(w)、延迟(d)和代价(c)。从节点 A 到节点 C 的物理路径共有 4 条:ABC、AC、

ADC 和 ADEC,根据定义 5,求出它们的特征值分别是:(8,7,2)、(6,3,1)、(3,2,2)和(5,6,3)。注意到 ABC 是带宽最好的路径,而 ADC 是延迟最好的路径,AC 则是代价最好的路径。我们希望用一条逻辑链路来表示这四条物理路径,如图 2(b)所示,这条逻辑链路的特征值用一个三维空间点(w, d, c)来表示,如何找到这个表示最好特征值的点是我们所关注的问题。

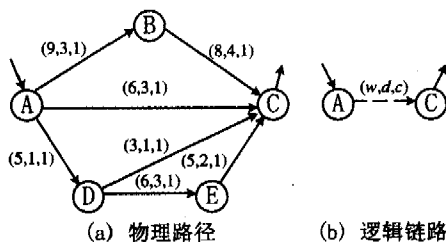


图 2 两边界节点间物理路径的逻辑表示

以带宽、延迟和代价为坐标轴的三维空间如图 3(a)所示,图中点 P_1 、 P_2 、 P_3 和 P_4 的坐标分别是图 2 中四条物理路径 ABC、AC、ADC 和 ADEC 的三个特征值,点 P_{best} 和 P_{worst} 分别是带宽最大、延迟和代价最小的最好点及带宽最小、延迟和代价最大的最坏点。图 3(b)和(c)是三维空间图 3(a)分别在延迟-带宽平面和代价-带宽平面内的投影,图中区域 $I^{(1)}$ 和区域 $I^{(2)}$ 所对应的空间称为实际可接纳空间(Actual Admissible Space, AAS),即一个连接请求对带宽、延迟和代价的要求如果全部落在这个空间之内,那么该连接请求会被接纳,否则将被拒绝。由于域的外部节点并不知道实际可接纳空间的边界,因此,Full-Mesh 图中逻辑链路的特征值应尽量准确地表示这部分空间。

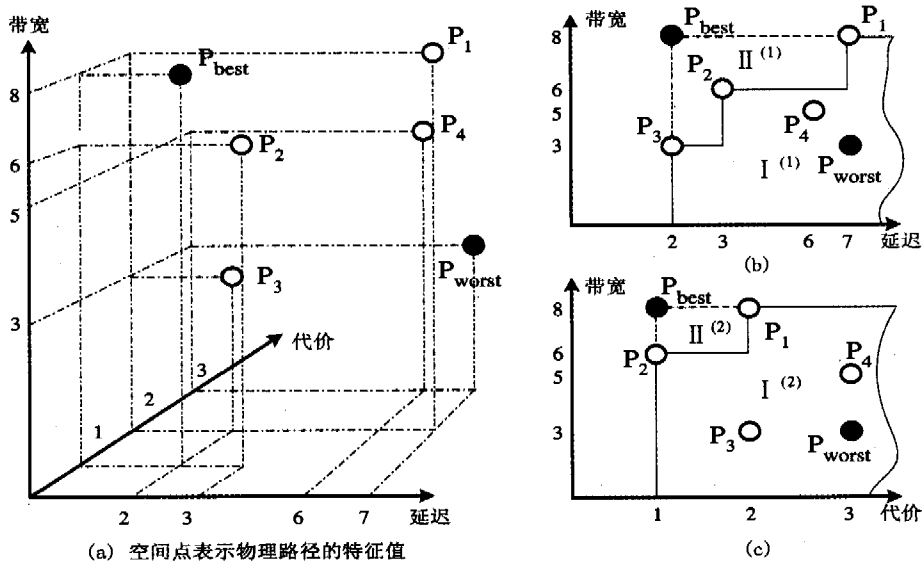


图 3 基于物理路径 QoS 特征值的空间划分

传统的方法是把 P_{best} 看作是最好点,用 P_{best} 的坐标作为逻辑链路的特征值。但从图 3(b)和(c)中可以看出,这种表示法扩大了实际可接纳空间,从而导致了重路由(Crankback)概率的增加,因为区域 $II^{(1)}$ 和区域 $II^{(2)}$ 实际上并不属于 AAS 的范围。为了减小重路由发生的概率,我们在 P_{best} 的坐标中引入一个附加参数-拉伸系数 α ,以求尽量减小 $II^{(1)}$ 和 $II^{(2)}$ 的区域范围。

假设 $P = \{p_1(Q_1^1, \dots, Q_n^1), \dots, p_n(Q_n^1, \dots, Q_n^k)\}$ 是两边界节点间 n 条物理路径的集合,每条路径 p_i 有 K 个 QoS 状态参数 $(Q_i^1, Q_i^2, \dots, Q_i^k)$,其中有 M 个凹性参数和 $K-M$ 个可加性参数。不失一般性,每个 QoS 参数(包括 M 个凹性参数和 $K-M$ 个可加性参数)最好值的表达式为:

$$Q_{best}^k = \max\{Q_i^k \mid 1 \leq i \leq n\}, k=1, \dots, M \quad (4)$$

$$Q_{best}^k = \min\{Q_i^k \mid 1 \leq i \leq n\}, k=M+1, \dots, K \quad (5)$$

定义 7 任意物理路径 $p_i(Q_i^w, Q_i^d, \dots, Q_i^k)$ 的拉伸系数

$$\alpha_i = \sum_{k=1}^M \frac{Q_{best}^k}{Q_i^k} + \sum_{k=M+1}^{K-M} \frac{Q_i^k}{Q_{best}^k}$$

其中 $\alpha_i \geq K$ 。

以图 2 为例,物理路径的集合 $P = \{p_1(8,7,2), p_2(6,3,1), p_3(3,2,2), p_4(5,6,3)\}$,根据定义 7,每条路径 $p_i(w_i, d_i, c_i) \in P, i=1,2,3,4$ 的拉伸系数为

$$\alpha_i = \frac{Q_{best}^w}{w_i} + \frac{d_i}{Q_{best}^d} + \frac{c_i}{Q_{best}^c} \quad (6)$$

其中 Q_{best}^w, Q_{best}^d 和 Q_{best}^c 分别是带宽、延迟和代价的最好值。从而可以求出具有最小拉伸系数的物理路径是 $p_2(6,3,1)$ 。

定义 8 逻辑链路的特征值应为 $(Q_{best}^w, Q_{best}^d, \dots, Q_{best}^k, \alpha)$,其中 $\alpha = \min\{\alpha_i | 1 \leq i \leq n\}$ 。

如果某一连接请求对 K 个 QoS 特征值的要求分别是 (R_1, R_2, \dots, R_K) ,那么判断某条逻辑链路是否满足此 QoS 约束的依据是:

$$R_i \leq Q_{best}^i, i=1, \dots, M$$

$$R_i \geq Q_{best}^i, i=M+1, \dots, K$$

$$\sum_{i=1}^M \frac{Q_{best}^i}{R_i} + \sum_{i=M+1}^{K-M} \frac{R_i}{Q_{best}^i} \geq \alpha \quad (7)$$

其中 α 是最小拉伸系数, M 是凹性状态参数的个数。只有当 (7) 式中的三个约束条件同时满足时,此连接请求才会被接纳,否则将被拒绝。当只存在三个 QoS 参数(带宽、延迟和代价)时,(7)式简化为

$$W \leq Q_{best}^w, D \geq Q_{best}^d, C \geq Q_{best}^c$$

$$\frac{Q_{best}^w}{W} + \frac{D}{Q_{best}^d} + \frac{C}{Q_{best}^c} \geq \alpha \quad (8)$$

其中 W, D 和 C 分别是带宽、延迟和代价的约束。

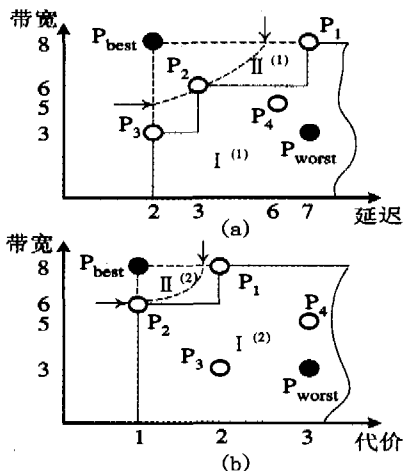


图 4 可接纳空间的调整

由于在最好点 P_{best} 的坐标中引入了拉伸系数 α (参照定义 8),使得区域 II⁽¹⁾ 和区域 II⁽²⁾ 的面积明显减少(箭头包围的部分被去掉),如图 4 所示,从而显著地减小了 Crankback 发生的概率。

3.2 构造生成树

根据混合 TA 算法的基本思想,对于 Full-Mesh 图中的带宽状态参数,我们拟采用 MST 算法进行表示和广告。此算法过程分为四步:(1)将网络域表示成关于带宽最大的 Full-Mesh 图(如图 5(a)和(b)所示);(2)在最大带宽 Full-Mesh 图中寻找最大带宽生成树(又称编码过程,如图 5(c)所示);(3)将最大带宽生成树向域外广告;(4)在状态信息的接

收端,最大带宽生成树被还原成最大带宽 Full-Mesh 图(又称解码过程)。

根据链路带宽无方向性的假设,构造最大带宽生成树的过程可以用贪心算法^[18]实现。此算法按带宽由大到小的顺序(降序)依次扫描 Full-Mesh 图中的链路,如果某一链路已与选择的链路构成了环,那么丢弃此链路,否则选择此链路,扫描过程重复进行,直到所有的链路被扫描。该算法的计算复杂度为 $O(|B|^2 \log |B|)$,这里 $|B|$ 为 Full-Mesh 图中的节点数。Full-Mesh 图经 MST 算法编码后,域间广告的状态信息量得以显著减少(压缩信息的复杂度由 $O(|B|^2)$ 降为 $O(|B|)$)。

最大带宽生成树的解码过程采用深度优先搜索算法。以图 5(c)为例,假设先从节点 A 开始,按 A-C-B-D 的顺序周游生成树。在周游过程中,我们始终保留已访问过链路的带宽最小值,从而可以计算出逻辑链路:A-C、A-B 和 A-D 的带宽值分别是:5、3 和 6。按照同样的方式可以分别计算出逻辑链路:B-C、B-D 和 C-D 的带宽值,最终还原出与图 5(b)完全一致的 Full-Mesh 图。该算法的计算复杂度为 $O(|B|^2)$ 。

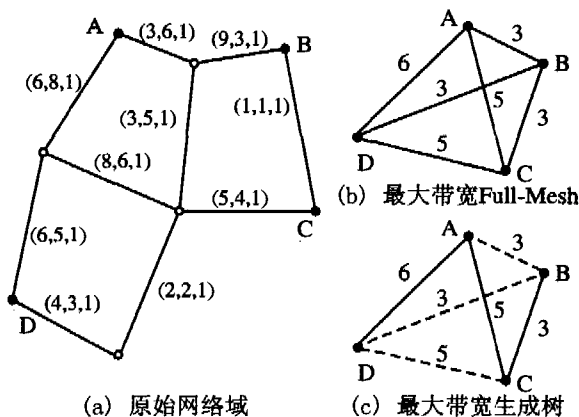


图 5 最大带宽生成树的编码和解码

定理 混合 TA 算法能够实现带宽状态参数的无损压缩。

证明:根据定义 8,由于边界节点间逻辑链路的特征值采用了最好点(即带宽最大、延迟和代价最小的空间点)的坐标,所以边界节点间的 Full-Mesh 图是关于带宽最大的 Full-Mesh 图。以图 5(b)和(c)为例,基于链路带宽的对称性,假设逻辑链路 AB、AC 和 BC 的带宽值分别为 L, L_1 和 L_2 ,那么 $L \geq \min(L_1, L_2)$,因为 AB 的带宽值是节点 A 和节点 B 之间所有物理路径的最大带宽值。另一方面,在最大带宽生成树 D-A-C-B 中, $L \leq \min(L_1, L_2)$,否则最大带宽生成树就应该是 D-A-B-C。所以 $L = \min(L_1, L_2)$,由此可见,最大带宽 Full-Mesh 图可以用 MST 算法进行正确地编码和解码。从而我们可以得出结论:混合 TA 算法能够实现带宽状态参数的无损压缩。

3.3 构造 Star 模型

非对称复合节点表示法(即 spokes 有着不同的权值或存在 bypass 的 Star 算法^[1])是通用的 TA 算法。此算法通常对 Full-Mesh 图进行压缩,是将 Full-Mesh 图表示成 Star 模型的过程。在还原 Full-Mesh 图时,任意两边界节点间逻辑链路的 QoS 特征值可以由这两个节点到虚拟核(nucleus)的两条逻辑链路的特征值经过简单的算术运算而得到。构造 Star 模型的关键是如何分配 spoke 或 bypass 的 QoS 特征值,使得

状态信息的失真尽可能小。

我们采用文[17]提出的求最小值的方法来分配 spoke 的 QoS 特征值。如图 6 所示,假设 Full-Mesh 图有 N 个顶点(即 Star 的边界节点), d_{ij} 为顶点 i 到顶点 j 的逻辑链路的 QoS 特征值,顶点 i 和 j 到 nucleus 的两条 spokes 的特征值分别是 X_i 和 X_j 。根据 Star 算法的聚集要求,上述参数应满足以下关系:

$$d_{ij} = X_i + X_j \quad (9)$$

考虑逻辑链路的有向性,在相反的方向上,(9)式中的关系仍然成立。现在的问题转化为如何求解 X_i 的值,使得函数 $f(X_1, \dots, X_N) = \sum_{\substack{j=1 \\ j \neq i}}^N (X_i + X_j - d_{ij})^2$ (其中 i, j 为 1 到 N 正整数)的值为最小^[17],用最小二乘法求解 X_i ,得

$$X_i = \frac{(2N-3) \sum_{\substack{j=1 \\ j \neq i}}^N d_{ij} - \sum_{\substack{j=1 \\ j \neq i}}^N \sum_{\substack{k=1 \\ k \neq j}}^N d_{jk}}{2(N-1)(N-2)} \quad (10)$$

式中的 i, j, k 都为 1 到 N 正整数。该算法无须用到 bypass,但能保证可加性状态参数的最小扭曲,其计算复杂度为 $O(|B|^2)$,压缩信息的复杂度为 $O(|B|)$,其中 $|B|$ 为边界节点数。

我们把 Full-Mesh 图中的拉伸系数 α 作为可加性状态参数看待(参照定义 8),因此当存在 3 个 QoS 状态参数(带宽、延迟和代价)时,Star 算法需要对延迟、代价和拉伸系数 α 这 3 个参数分别运用(10)式进行计算、表示和广告。

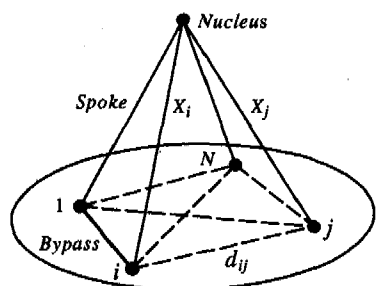


图 6 复合节点表示法(Star 模型)

4 路由算法

在两层结构的网络中,路由计算也分为两个层次:域间路由和域内路由。如图 7 所示,层次化的网络拓扑由 Waxman 算法^[19]生成,参照定义 4 和定义 6,图中 A 域内的节点 s 为一多播组 G 的源节点, E 域内的节点 t 为 G 的一目的节点,目的节点以渐进加入的方式形成域间多播树。在新成员加入多播树的过程中,需要计算从内部节点到外部域边界节点的路径(例如 $E.t \rightarrow A.1$)以及从边界节点到本域内部节点的路径(例如 $A.1 \rightarrow A.s$),这两种不同路径的计算过程分别称为域间路由和域内路由,另外,在同一域中边界节点之间的路径计算(例如 $B.1 \rightarrow B.2$)也称为域内路由。需要说明的是:多播组的源节点与目的节点处在同一域中的路由问题不属于本文所讨论的范畴。

4.1 概述

我们这里提出一种适应于多 QoS 约束的层次多播路由算法 HMRAMQ (Hierarchical Multicast Routing Algorithm with Multiple QoS constraints)。此算法的基本思想是:当某个接收方欲加入一多播会话时,它将发送一加入请求报文给该多播会话的源节点,并利用单播路由模式初始化此路由过程。当某个中间节点接收到加入请求报文后,就对该报文所经过的路径进行合法性测试(即检测路径的带宽和延迟是否

满足多播会话的 QoS 要求,并考虑策略的限制),如果此路径通过了合法性测试,那么加入请求报文将被继续前递至源节点,否则中间节点将进入分叉路由模式。在分叉路由模式下, HMRAMQ 可能搜索多条路径,并对这些路径进行合法性测试,在那些通过了合法性测试的路径(可行性路径)中,代价最优(或近优)的路径将被选取。在 HMRAMQ 的工作过程中,域间多播树以渐进的方式生成,多播组的成员可以动态地加入/退出一个多播会话,并不会对现存的多播树带来干扰,即采用多播树渐进生成的方式来实现多播树状态的无缝迁移。HMRAMQ 是一种分布式的层次多播路由算法,但在单播路由模式下,需要用到域外聚集的拓扑状态信息和域内详细的拓扑状态信息,进行集中式的路由计算。

在进行路由计算时,各域的边界节点需要把接收到的聚集状态信息广播到本域的内部节点,在内部节点的拓扑状态数据库中插入有关外部域边界节点的状态信息条目,让内部节点也能看到外部域的边界节点,便于域间路由的计算。拓扑状态数据库所维护的状态信息条目为(域内完整拓扑,域外聚集拓扑, QoS 特征值,拉伸系数),单播路由计算模块根据这些状态信息来填写单播路由表。路由器中多播路由模块所维护的多播路由条目为(组地址,源地址,上游节点地址,下游节点地址集合,上游接口,下游接口集合,预留资源),这些路由信息由多播互操作模块映射到多播路由表中,指导多播分组的转发。

HMRAMQ 所用到的控制报文主要有 7 类:(1)加入请求报文 Join-Req,它是希望加入多播组的新成员发向该多播组源节点的探测报文,用于探测到源节点的可行性路径;(2)加入确认报文 Join-Ack,此报文由接受加入请求的节点回送给请求加入的节点,用于对加入请求的响应;(3)加入撤消报文 Join-Und,此报文由请求加入的节点发向接受加入请求的节点,用于对加入请求的撤消;(4)资源预留报文 Resv,它是源节点发送给新成员的响应报文,此报文可沿路进行资源预留;(5)预留确认报文 Resv-Ack,此报文用于资源预留成功的通知;(6)预留撤消报文 Resv-Und,它是当中间节点没有可供预留的资源时向源节点和新成员发送的消息,此报文用于对已预留资源的撤消及预留失败的通知。(7)剪枝报文 Prune,它是多播树的叶子节点发向其上游节点的退出报文。

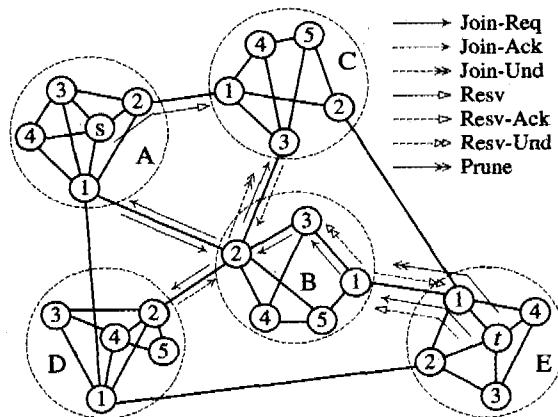


图 7 HMRAMQ 的工作过程

4.2 状态信息

根据 PNNI 的建议,我们可以在每个域中指定一个节点专门负责聚集本域的拓扑状态信息和还原外部域的拓扑状态信息,此节点称为 PGL(Peer Group Leader),如图 1 所示。一

方面,PGL先将本域的完整拓扑压缩成仅包含边界节点 Full-Mesh图,再把 Full-Mesh图转换为最大带宽生成树以及延迟、代价和拉伸系数的 Star表示,并把这些聚集的状态信息发送到本域的所有边界节点;另一方面,PGL把从边界节点那里接收到的、外部域的聚集状态信息(包括最大带宽生成树及延迟、代价和拉伸系数的 Star表示)还原成 Full-Mesh图,再把 Full-Mesh图广播到本域中所有的节点。

HMRAMQ在进行域间路由计算时需要用到外部域的聚集状态信息,这些状态信息包括外部域边界节点间物理路径的最大带宽、最小延迟、最小代价和拉伸系数。为了使域间路由的计算更加精确,以减小 Crankback 发生的概率,我们需要对这些状态信息作进一步调整。(6)式所表示的是拉伸系数在带宽-延迟-代价空间内的曲线,将此空间曲线分别在带宽-延迟平面和带宽-代价平面内作投影变换,得

$$\alpha_i^{(1)} = \frac{Q_{best}^w}{w_i} + \frac{d_i}{Q_{best}^d}, \alpha_i^{(2)} = \frac{Q_{best}^w}{w_i} + \frac{c_i}{Q_{best}^c} \quad (11)$$

因此(8)式转化为:

$$W \leq Q_{best}^w, D \geq Q_{best}^d, C \geq Q_{best}^c$$

$$\frac{Q_{best}^w}{W} + \frac{D}{Q_{best}^d} \geq \alpha^{(1)}, \frac{Q_{best}^w}{W} + \frac{C}{Q_{best}^c} \geq \alpha^{(2)} \quad (12)$$

式中的 $\alpha^{(1)}$ 和 $\alpha^{(2)}$ 分别为拉伸系数 $\alpha_i^{(1)}$ 和 $\alpha_i^{(2)}$ 的最小值,它们随其它的特征值一起被广告。

假设多播会话对带宽的要求为 W ,根据带宽状态参数的凹性特征和(12)式可以计算出外部域边界节点间逻辑链路的延迟值和代价值,此结果比用 Q_{best}^d 和 Q_{best}^c 表示逻辑链路的延迟和代价更加精确。

4.3 算法描述

假设多播组 G 的新成员节点 t 在请求加入之前已经通过 SAP(Session Announcement Protocol)^[20] 和 SDP(Session Description Protocol)^[21] 查询到源节点 s 对带宽和延迟的要求分别是 W 和 D 。HMRAMQ的工作过程主要分为加入计算、资源预留和退出计算 3 个阶段。

4.3.1 加入计算

如图 7 所示,当节点 t 欲加入多播组 G 时,它将发送一加入请求报文 Join-Req 给 G 的源节点 s ,HMRAMQ 进入单播路由模式,对 Join-Req 的路径进行初始化计算。 t 到 s 的单播路由计算分为两个层次:

1)域间路由计算。HMRAMQ 首先根据节点 E, t 所获取的全局拓扑状态信息剪掉从 E, t 到 A 域的所有路径中那些不满足带宽约束的物理链路和逻辑链路,然后应用 Dijkstra 算法计算从 E, t 到 A 域的最小代价路径。需要说明的是,在 A 域的所有边界节点中,与 E, t 最近(即代价最小)的边界节点作为 Join-Req 的目的节点。

2)域内路由计算。 E, t 的拓扑状态数据库中并没有其它域的详细信息,因此 E, t 不能计算从 $B, 1$ 到 $B, 2$ 以及从 $A, 1$ (或 $A, 2$) 到 A, s 的最小代价路径。由于 $B, 1$ 了解本域的详细拓扑状态,因此当 Join-Req 到达 $B, 1$ 时, $B, 1$ 可以应用 Dijkstra 算法计算出到 $B, 2$ 的最小代价路径,进而指导 Join-Req 在域内的转发。同理, $A, 1$ (或 $A, 2$) 也能计算出到 A, s 的最小代价路径。

Join-Req 在域间和域内转发的过程中,不仅记录下自己所经过的路径,而且对这条路径的 QoS 特征值(包括最小带宽、累计延迟和累计代价)进行跟踪。

假设 $w(E, t, B, 2)$ 和 $d(E, t, B, 2)$ 分别表示路径 $p(E, t,$

$B, 2)$ 的最小带宽和累计延迟, $w(B, 2, A, 1)$ 和 $d(B, 2, A, 1)$ 分别表示 $B, 2$ 到 $A, 1$ 的链路带宽和延迟。当中间节点 $A, 1$ 收到 Join-Req 后,将进行测试:

$$\min(w(B, 2, A, 1), w(E, t, B, 2)) \geq W$$

$$\wedge d(B, 2, A, 1) + d(E, t, B, 2) \leq D \quad (13)$$

如果上式成立, $A, 1$ 就把 Join-Req 前递到它的直接上游节点,同时向其直接下游节点 $B, 2$ 发送一加入确认报文 Join-Ack。此过程重复进行,直到源节点 s 最终收到 Join-Req,并进行类似的合法性测试,如果测试成功, s 将沿着 Join-Req 所经过的路径向新成员节点 t 反向发送一资源预留报文 Resv, HMRAMQ 随即进入资源预留阶段。

如果(13)式不成立, $A, 1$ 就将 Join-Req 回送给 $B, 2$, $B, 2$ 进入了分叉路由模式。这时 HMRAMQ 可能搜索多条路径(例如 $p(E, t, C, 3)$ 和 $p(E, t, D, 2)$),并对这些路径进行类似于(13)式的合法性测试。如果 $p(E, t, C, 3)$ 或 $p(E, t, D, 2)$ 通过了测试,节点 $C, 3$ 或 $D, 2$ 就向 $B, 2$ 发送 Join-Ack; $B, 2$ 如果收到了多个 Join-Ack,就通过向 $C, 3$ 或 $D, 2$ 发送加入撤消报文 Join-Und 来选择最小代价的路径;如果 $B, 2$ 没有收到 Join-Ack,则路由失败。需要说明的是:加入请求报文 Join-Req 在域间转发时,边界节点还可以根据某些策略来控制 Join-Req 的发送方向。

4.3.2 资源预留

如上所述,资源预留报文 Resv 由多播组 G 的源节点 s 加上时间戳 T_1 后发向新成员节点 t ,并沿着加入请求报文 Join-Req 所经过的反向路径进行资源预留。如图 7 所示,当 Resv 到达某个中间节点(例如 $B, 1$,且 $B, 1 \notin G$)时,如果 $B, 1$ 有可供预留的资源,就为该多播组预留资源并继续前递 Resv,否则它将沿着 Resv 和 Join-Req 所经过的反向路径向 s 和 t 分别发送预留撤消报文 Resv-Und,通知它们资源预留过程失败,并沿路撤消已经预留的资源。如果 t 最终收到了 Resv,即对 Resv 加上另一个时间戳 T_2 并检测:

$$T_2 - T_1 \leq D \quad (14)$$

如果上式成立,就向 s 或某个就近的树上节点发送预留确认报文 Resv-Ack,到此路由连接成功建立,多播应用开始工作。否则,将沿 Resv 经过的反向路径向 s 发送预留撤消报文 Resv-Und。

4.3.3 退出计算

当成员节点 t 要求退出多播会话时,如果 t 是多播树的叶子节点,则向其上游节点发送一剪枝报文 Prune, Prune 被传送至树叉节点后,树叉节点将删除关于 t 的路由信息并释放相关的网络资源,多播树其余的部分保持不变。否则,不需要做处理。

5 复杂性分析

计算的时间和所需的空间是衡量一个算法性能的两个重要依据。在本节中,我们将分析和讨论我们所提出的算法框架 HMRMQ 在所需的空间和计算时间方面的复杂性。

5.1 空间复杂度

HMRMQ 所需的空间主要包括单播路由表的存储空间及控制报文的开销。一个单播路由表中包括两类路由条目:域间路由条目和域内路由条目,其中域间路由的条目数为 $|G| - 1$,域内路由的条目数为 $|V_i| - 1$,参照定义 1,这里的 $|G|$ 表示域的个数, $|V_i|$ 表示域 g_i 的节点数;域 g_i 中单播路由表的存储空间为 $O(|V_i|(|V_i| + |G|))$ 。整个网络中所有单播路

由表的存储空间为 $O(\sum_{i \in G} |V_i|(|V_i| + |G|))$ 。如果式中的 $|G| \approx |V_i|$, 那么此式简化为 $O(\sum_{i \in G} |V_i|^2)$, 其复杂度远低于平面路由的空间复杂度 $O(\sum_{i \in G} |V_i|^2)$ 。

对于控制报文的开销, 在路由算法 HMRAMQ 工作的每个阶段最多同时使用两类报文: Join-Req 和 Join-Ack 或 Join-Nack, 并且在 Join-Req 所经过的路径上最多有 $O(\sum_{i \in G} |E_i| + |L|)$ 条链路, 其中 $|L|$ 表示域间的链路数, $|E_i|$ 表示域 g_i 的链路数。因此, 对于有 $|M|$ 个成员的多播组, 其报文复杂度最多为 $O(2|M|(\sum_{i \in G} |E_i| + |L|))$ 。

5.2 计算复杂度

HMRMQ 的计算时间主要包括路由算法的计算时间和 TA 算法的计算时间。在 HMRAMQ 的加入计算阶段, 在 Join-Req 报文探索的每节点上都可能使用 Dijkstra 算法来计算到源节点的最小代价路径。因此, 域间路由的计算复杂度应为 $O(|V'|(|E'| + |V'| \log |V'|))$, 这里 $|V'| = |V_n| + \sum_{i \in G} |B_i|$, $|E'| = |L| + |E_n| + \sum_{i \in G} |B_i|$, 其中 $|G|$ 表示域的个数, $|L|$ 表示域间的链路数, $|V_n|$ 和 $|E_n|$ 分别表示新成员节点所在域的节点数和链路数, $|B_i|$ 表示域 g_i 的边界节点数; 域内路由的计算复杂度为 $O(\sum_{i \in G} |V_i|(|E_i| + |V_i| \log |V_i|))$, 其中 $|V_i|$ 和 $|E_i|$ 分别表示域 g_i 的节点数和链路数。由于 HMRAMQ 其它工作阶段的计算时间小于加入计算阶段的计算时间, 因此, 对于有 $|M|$ 个成员节点的多播组, 其计算复杂度应是 $O(|M|(|V'| |E'| + |V'|^2 \log |V'|))$ 与 $O(|M| \sum_{i \in G} (|V_i| |E_i| + |V_i|^2 \log |V_i|))$ 之和。

表 1 混合 TA 算法的计算复杂度

计算两边界节点间所有物理路径的特征	$O(V E ^2)$
求最好点和拉伸系数	$O(2 E)$
计算一条逻辑链路的特征值	$O(V E ^2) + O(2 E)$ $= O(V E ^2)$
构造 Full-Mesh 图	$O(B ^2 V E ^2)$
计算最大带宽生成树	$O(B ^2 \log B)$
构造 Star 模型	$O(B ^2)$
还原 Full-Mesh 图	$O(B ^2)$
总计计算时间	$O(B ^2 V E ^2) + O(2 B ^2) = O(B ^2 V E ^2)$

混合 TA 算法的计算复杂度如表 1 所示, 表中的 $|V|$ 、 $|B|$ 和 $|E|$ 分别表示一个域的节点数、边界节点数及链路数。

6 仿真实验

为了验证 HMRMQ 算法框架的性能特征, 我们对仿真软件 NS2 的功能作了扩展, 增加了拓扑生成器、QoS 路由、链路状态更新及按需路由计算等功能模块, 并在此平台之上实现了我们所提出的混合 TA 算法和 HMRAMQ 路由算法。为了测评 HMRMQ 的拓扑聚集和路由算法的性能并与其它的多播路由算法进行比较, 我们定义以下 5 个性能参数:

交换的信息量 = 拓扑、带宽、延迟和代价分组数之和
 维护的信息量 = 单播和多播路由表中路由条目数之和

路由成功率 = $\frac{\text{成功加入的成员节点总数}}{\text{加入请求报文的总数}}$

平均网络代价 = $\frac{\text{全部多播树的总体代价}}{\text{成员节点的总数}}$

平均报文负载 = $\frac{\text{控制报文的总数}}{\text{加入请求报文的总数}}$

这里所比较的路由算法是我们在文[22]中所提出的多 QoS 约束的平面多播路由协议 MRPMQ 及文[14]所提出的 QoS 层次多播路由协议 QHMRP。实验重点比较了层次路由与平面路由(HMRMQ 与 MRPMQ)之间以及不同的层次路由算法(HMRMQ 与 QHMRP)之间的性能差异。

6.1 仿真环境

仿真实验所采用的网络拓扑由我们开发的拓扑生成器 Top-Builder 产生, Top-Builder 是一种基于 Waxman 模型^[19]、独立于 NS2 并提供与 NS2 接口的拓扑仿真软件。整个仿真拓扑由 10 个网络域组成, 每个域有 18 个节点, 共 180 个节点, 各域的边界节点数由节点的度决定(这里域内和域间节点的度分别取 3 和 2)。域内和域间的网络拓扑都用 Waxman 模型生成, 并选用相同的模型参数 ($\alpha=0.15, \beta=0.2$)。域内有向链路的带宽值在 [10Mbps, 50Mbps] 间均匀分布, 延迟在 [0.5ms, 3ms] 间均匀分布; 域间有向链路的带宽值在 [50Mbps, 100Mbps] 间均匀分布, 延迟在 [1ms, 5ms] 间均匀分布; 所有链路的代价都为 1 个单位。需要说明的是: MRPMQ 的仿真拓扑是只包含一个网络域(该域有 180 个节点)的平面结构。

实验中我们采用 NS2 的 CBR(Constant Bit Rate)流量模型, 以产生恒定的流量。多播组的数目固定为 5, 每个组的规模都相同且不超过全部节点的 20%。多播组的源节点和成员节点在网络节点中随机抽取, 在 QHMRP 和 HMRMQ 的仿真过程中, 我们忽略源节点和成员节点处在同一个域中的情况。每个实验值都是经过多次仿真得到的平均值。

6.2 实验结果

表 2 比较了 MRPMQ、QHMRP 和 HMRMQ 工作时全部路由器交换和维护的状态信息总量。从表中可以看出, 网络层次划分和拓扑聚集不仅有效地降低了路由器之间交换的信息量, 也显著减小了路由表的存储空间。由于 QHMRP 只考虑了带宽约束, 因而路由器维护的信息总量略少于 HMRMQ, 但它采用了 Full-Mesh 拓扑聚集算法, 使得路由器间交换的信息总量要比 HMRMQ 大得多。

表 2 交换和维护的信息量

路由算法	交换的信息总量	维护的信息总量
MRPMQ	91576	32592
QHMRP	21049	3645
HMRMQ	10683	3817

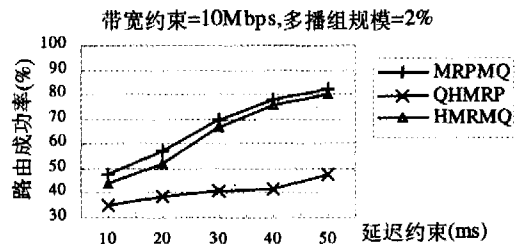


图 8 路由成功率与延迟约束

MRPMQ、QHMRP 和 HMRMQ 的路由成功率随延迟约束和多播组规模变化的曲线分别如图 8 和 9 所示。可以看出, 对延迟的约束越严格、可用的网络资源越少(由于多播组规模增加), 各路由算法的路由成功率就越低。相较而言, 平面路由的成功率要比层次路由的成功率稍高一些, 但不显著,

其主要原因是聚集后的拓扑状态信息基本上能够反映下层网络的特征。由于QHMRP并没有考虑延迟约束,因而相对于延迟约束它的路由成功率很低(如图8所示)。但在图9中QHMRP的路由成功率最高,这是因为在可行性路径的探索过程中QHMRP采用了泛洪(Flooding)法,此算法提高了路由计算的成功率,但增加了控制报文的负载。

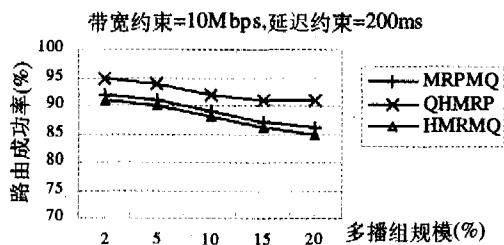


图9 路由成功率与多播组规模

图10和图11分别表示MRPMQ、QHMRP、HMRMQ的平均网络代价与延迟约束和多播组规模之间的变化关系。总体来讲,对延迟的约束越宽松、多播组的规模越大,各路由算法的平均网络代价就越低。其中QHMRP的平均网络代价最高,且基本上不随延迟约束的增加而变化,这是因为其多播树的生成不取决于延迟约束;HMRMQ则具有最佳的网络代价性能。从图中还可以看出,层次路由的平均网络代价比平面路由低,仅在多播组规模较小时有些例外(如图11所示),其主要原因是当网络资源丰富时,采用逻辑拓扑(即层次化的拓扑)计算出来的路由路径可能要比采用平面拓扑计算出来的路径长一些。

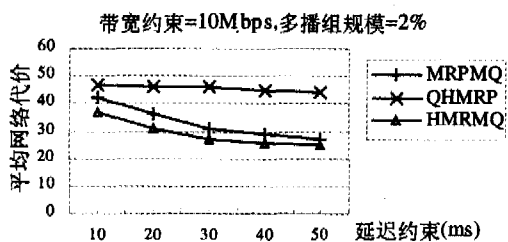


图10 平均网络代价与延迟约束

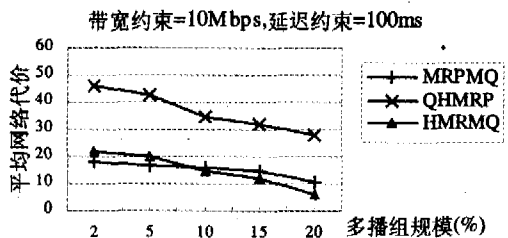


图11 平均网络代价与多播组规模

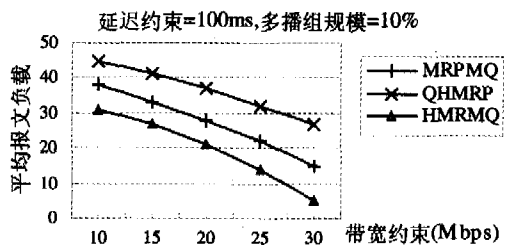


图12 平均报文负载与带宽约束

约束和延迟约束之间的变化关系分别如图12和13所示。从图中可以看出,对带宽和延迟的限制越宽松,各路由算法的平均报文负载就越大,因为有更多的加入请求被接受,从而会产生更多的控制报文。如前所述,QHMRP采用了Flooding算法来寻找可行性路径,因此它的平均报文负载最大。HMRMQ的平均报文负载比MRPMQ小,这是因为网络层次划分和拓扑聚集限制了控制报文的扩散。

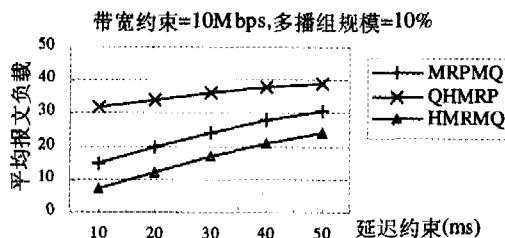


图13 平均报文负载与延迟约束

结束语 本文针对大规模网络中QoS多播路由的扩展性问题,提出了一种具有多QoS约束的层次多播路由算法框架HMRMQ。本文的主要贡献在于:(1)提出了一种能表征多个QoS链路状态参数(包括带宽、延迟、代价等)并具有很好扩展性的聚集特征表示法-带拉伸系数的最好点法;(2)提出了一种基于多QoS状态参数的拓扑聚集算法-混合TA算法;(3)在聚集算法的基础上,提出了一种适应于多QoS约束并易于实施某些策略的层次多播路由算法-HMRMQ。本文从理论上证明了HMRMQ的正确性和有效性,分析了其时空计算复杂度。仿真实验比较了HMRMQ与MRPMQ和QHMRP的性能差异,实验结果表明我们的算法在交换和维护的信息量、路由成功率、网络代价以及报文负载等方面都具有良好的性能。总之,HMRMQ算法框架为QoS层次多播路由的研究提供了一种新的方法和有效途径。

后续的研究工作包括:(1)进一步分析拓扑结构、更新策略、网络负载等因素对拓扑聚集和路由算法的影响;(2)由于动态网络中的状态信息存在固有的非精确性,因此,我们需要对HMRMQ算法框架作进一步改进,提高混合TA算法的聚集精度,并使HMRMQ能够适应状态信息的不精确性。

参考文献

- 1 Private network-network interface specification version 1.0 (PN-NI 1.0). The ATM Forum Technical Committee, af-pnni-0055.000, Mar. 1996
- 2 Guerin R A, Orda A. QoS routing in networks with inaccurate information: Theory and algorithms. IEEE/ACM Transactions on Networking, 1999, 7(3): 350~364
- 3 Fang Hao, Zegura E W. On scalable QoS routing: Performance evaluation of topology aggregation. In: Proc. of the Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2000), 2000, 1, 147~156
- 4 Awerbuch B, Shavitt Y. Topology aggregation for directed graphs. In: Proc. of the Third IEEE Symposium on Computers and Communications (ISCC'98), 1998. 47~52
- 5 Lee Whay Chiou. Minimum equivalent subspanner algorithms for topology aggregation in ATM networks. In: Proc. of the 2nd International Conference on ATM (ICATM'99), Jun. 1999. 351~359
- 6 Wang Z, Crowcroft J. Quality-of-service routing for supporting multimedia applications. IEEE Journal on Selected Areas in Communications, 1996, 14(7): 1228~1234
- 7 Lee Whay Chiou. Spanning tree method for link state aggregation in large communication networks. In: Proc of IEEE INFOCOM'95, Boston, USA, 1995. 297~302
- 8 Korkmaz T, Krunz M. Source-oriented topology aggregation with multiple QoS parameters in hierarchical ATM networks. In: Proc of the Seventh International Workshop on Quality of Service (IWQoS'99), 1999. 137~146

(下转第110页)

为以后合成的原子服务。

如果需要一种服务,它可以取得某一固定电话号码所在区域的地图,并且没有任何已存的服务可以满足这个要求,则可以通过服务合成的方法来动态创建。

第一步选择目标服务, $OS = \{Map\}$, $T = MapService$, 对于服务进行功能的语义匹配,结果如表 1。

表 1 选择目标服务的语义匹配结果

服务	输出参数	服务类型	响应时间(ms)	语义距离
S11	Map	MapService	2526	0
S12	TrafficMap	MapService	2380	1
S13	FacilityMap	FacilityMapService	2465	1
S14	TrafficMap	TrafficMapService	5051	1
S15	Map	MapService	4610	0
S16	FacilityMap	MapService	2205	1

对于功能匹配, S11 和 S15 的语义距离都为 0, 都是精确匹配。在性能方面主要考虑响应时间这个指标。因为 S11 的响应时间较短, 所以选择 S11 作为目标服务。S11 有三个参数, Diameter 参数由用户提供, 另外两个输入参数为 Latitude 和 Longitude, 需要进行处理。

第二步选择下一个服务来提供 S11 的 Latitude 参数, $OS = \{Latitude\}$, $T = LocationService$, 功能的语义匹配结果如表 2。

表 2 第二步语义匹配的结果

服务	输出参数	服务类型	响应时间(ms)	语义距离
S21	Latitude, Longitude	LocationService	2018	0
S22	Latitude, Longitude	LocationService	2550	0
S23	Latitude, Longitude	LocationService	1800	0

选择服务 S23, 要处理其输入参数 GprsLocation。

依此处理, 最后得到的合成服务为:

为 S11 提供 Latitude 参数和 Longitude 参数的 S23 的两个实例为相同实例, 应消除一个。

合成服务的输入参数为: Diameter 和 FixedPhoneNumber; 输出参数为 Map。用户只要输入一个固定电话号码和一个所需要地图的直径值, 就可以取得其所在区域的地图。

服务合成是否可以成功, 取决于服务库中是否有足够多

的合适的服务。在服务合成过程中, 如果需要某些特殊功能的服务不存在, 那么服务合成就会失败。

总结 电子商务经历了发布式和动态交互式两个阶段。当前动态交互式电子商务已经暴露出了许多问题, 不能适应电子商务进一步发展的需要。电子商务的进一步发展必然要引入新的技术, 即 Web 服务技术。将 Web 服务技术应用于电子商务, 服务合成是其中一个关键问题。本文提出了一个半自动化服务合成的原型系统和基于目标服务的合成算法, 并实现了一个合成系统, 对所提出的算法进行测试。

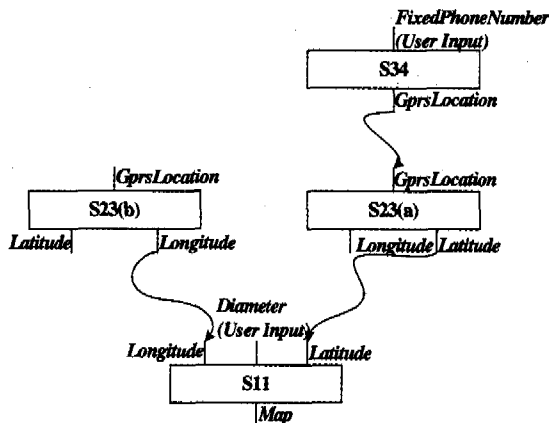


图 2 一个简单的合成实例的过程模型

参考文献

- Casati F, Ilnicki S, Jin L. Adaptive and dynamic service composition in EFlow. In: Proceedings of 12th International Conference on Advanced Information Systems Engineering (CAiSE), Stockholm, Sweden, June 2000
- McDermott D. Estimated-regression planning for interactions with Web services. In: Proceedings of the 6th International Conference on AI Planning and Scheduling, Toulouse, France, 2002
- Medjahed B, Bouguettaya A, Elmagarmid A K. Composing Web services on the Semantic Web. The VLDB Journal, November 2003, 12(4)
- Wu D, Sirin E, Hendler J, et al. Automatic Web services composition using SHOP2. In: Workshop on Planning for Web Services, Trento, Italy, June 2003
- Ankolenkar A, Burstein M, Hobbs J R, et al. DAML-S: Web Service Description for the Semantic Web. In: Proceedings of The First International Semantic Web Conference (ISWC), Sardinia (Italy), June 2002
- Sivashanmugam K, Verma K, Sheth A, et al. Adding Semantics to Web Services Standards. In: Intl Conf. on Web Services, Las Vegas NV, June 2003
- Paolucci M, Kawamura T, Payne T R, et al. Semantic matching of Web service capabilities. In: Proceedings of the 1st International Semantic Web Conference, Sardinia, Italy, June 2002. 318~332

(上接第 34 页)

- Lui King-Shan, Nahrstedt K, Chen Shigang. Hierarchical QoS routing in delay-bandwidth sensitive networks. In: Proc of the 25th Annual IEEE Conference on Local Computer Networks (LCN 2000), Nov. 2000. 579~588
- Lui King-Shan, Nahrstedt K, Chen Shigang. Routing with topology aggregation in delay-bandwidth sensitive networks. IEEE/ACM Transactions on Networking, 2004, 12(1): 17~29
- 闵应骅. 计算机网络路由研究综述. 计算机学报, 2003, 26(6): 641~649
- Moh W M, Xiang L, Zhao X. Extending BGMP for QoS-based inter-domain multicasting over the Internet. In: Proc of IEEE International Conference on Communications (ICC 2000), Jun 2000, 2: 728~734
- Yan Shuqian, Faloutsos M, Banerjee A. QoS-aware multicast routing for the Internet: The design and evaluation of QoS-MIC. IEEE/ACM Transactions on Networking, 2002, 10(1): 54~66
- Pradhan S, Yi Li, Maheswaran M. QoS-aware hierarchical multicast routing on next generation internetworks. In: Proc of the

- 20th IEEE International Conference on Performance, Computing and Communications, Arizona, USA, Apr 2001. 9~16
- 陆慧梅, 向勇, 史美林. 支持 QoS 的层次组播路由算法框架 QH-MR. 计算机学报, 2004, 27(6): 772~781
- Awerbuch B, Du Y, Khan B, et al. Routing through networks with hierarchical topology aggregation. In: Proc of the Third IEEE Symposium on Computers and Communications (ISCC '98), 1998. 406~412
- 刘进. 大规模层次化网络中保证服务质量的路由算法. [学位论文]. 北京: 清华大学, 2000
- Papadimitriou C H, Steiglitz K. Combinatorial optimization; Algorithms and complexity. New York: Dover Publications, Inc, 1998
- Waxman B M. Routing of multiple connections. IEEE Journal on Selected Areas in Communications, 1998, 6(9): 1617~1622
- Handley M, Perkins C, Whelan E. Session announcement protocol. RFC 2974, Oct 2000
- Handley M, Jacobson V. SDP: Session description protocol. RFC 2327, Apr 1998
- 李贻元, 李春林. 多 QoS 约束的多播路由协议. 软件学报, 2004, 5(2): 286~291