

结合分布式与集中式特点的动态多播路由算法

Dynamic Multicast Routing Algorithm Based on Distributed and Centralized Algorithm

李昌兵 杜茂康 (重庆邮电大学 重庆 400065)

胡 华 (重庆大学 重庆 400044)

摘要: 针对在多播成员动态变化的环境中的多播路由树的建立和调整等问题,本文提出结合分布式和集中式特点的动态多播路由算法,在新节点加入多播会话时,采用分布式的方法搜索新节点到当前多播树的满足 QoS 约束的最优路径。而在成员节点离开多播会话时,根据其多播会话质量的影响程度,决定是否执行多播树的重构操作。与静态多播路由算法比较,该算法具有更好的适应性和灵活性。仿真结果验证了算法的有效性。

关键词: QoS 动态多播 路由算法 集中式算法 分布式算法

1 引言

随着网络电视会议、远程教育等业务相继出现,对网络资源的优化也提出了新的要求,多播技术就是为了节约网络资源而提出的。作为多播技术研究重点的多播路由问题分为两种类型:一种是静态多播路由,即在整个多播会话期间不允许组成员动态变化;另一种是动态多播路由,允许组成员随时加入或离开。在网络电视会议、远程教育等多媒体业务中常采用动态多播路由。研究人员对静态多播 QoS 路由优化算法进行了广泛的研究。而对动态多播路由问题,已被证明是 NP 完全问题。目前研究较多的仍旧只是动态 Steiner 问题,对于考虑时延约束和费用优化的动态多播路由问题,目前研究得较少^[1-3]。

本文针对动态多播路由问题,提出了结合分布式和集中式特点的部分重构动态多播 QoS 路由算法,算法仿真表明,它比采用启发性算法和贪婪算法具有更低的代价。

2 问题描述

网络用加权有向图 $G(V, E)$ 表示。其中 V 是节点集 E 是节点间双向链路集。 $(\forall l = (u, v) \in E$, 链路 $l(u, v)$ 的状态可由其路由跳数 $h(u, v) = h(l)$, 开销 $c(u, v) = c(l)$, 延迟 $d(u, v) = d(l)$, 延迟抖动 $j(u, v)$, 可用带宽 $b(u, v)$ 和包丢失率 $l\alpha(u, v) = l\alpha(l)$ 联合表示。令 $M \subset V$ 为多播组, $v \in M$ 为组成员, 则多播树 T

是 G 的子图 $PT(v_s, v_d)$ 表示树 T 中一条从源节点 v_s 到目标节点 $v_d \in M - \{v_s\}$ 的路径。因此,基于路由跳数、开销、延迟、抖动、可用带宽和包无损传输率的多 QoS 约束动态多播路由问题可描述如下:

给定图 $G(V, E)$ 、多播信源 v_s 、动态多播组 $M \subset V$, 确定多播最短路径(SPT)树 $T \subseteq G$, 使得在满足下列 QoS 约束条件

$$\forall v_d \in M, D(P_T(v_s, v_d)) = \sum_{(u,v) \in P_T(v_s, v_d)} d(u,v) \leq D_{\max} \quad (1)$$

$$J(P_T(v_s, v_d)) = \sum_{(u,v) \in P_T(v_s, v_d)} j(u,v) \leq J_{\max} \quad (2)$$

$$B(P_T(v_s, v_d)) = \min_{\sum_{(u,v) \in P_T(v_s, v_d)} d(u,v) \leq D_{\max}} b(u,v) \geq B_{\min} \quad (3)$$

$$Z_{NL}(P_T(v_s, v_d)) = \prod_{(u,v) \in P_T(v_s, v_d)} (1 - l\alpha(u,v)) \geq Z_{\min} \quad (4)$$

的前提下实现如下优化目标:

$$H(P_T(v_s, v_d)) = \sum_{(u,v) \in P_T(v_s, v_d)} h(u,v) \rightarrow \min \quad (5)$$

$$C(P_T(v_s, v_d)) = \sum_{(u,v) \in P_T(v_s, v_d)} c(u,v) \rightarrow \min \quad (6)$$

在 QoS 约束条件中, D_{\max} 表示延迟上限, J_{\max} 表示延迟抖动上限, B_{\min} 表示带宽下限, Z_{\min} 表示包无损传输概率下限。若优化目标 5 和目标 6 发生冲突, 优先考虑目标 6, 即新节点的路径应尽量利用已有的多播树, 在相同开销的情况下, 选择跳数较小的路径。

3 算法原理

本文的目的是为动态环境设计一个满足 QoS 约束

的动态多播路由算法,使得每次多播组成员变化后,新构造的多播树满足 QoS 约束并且费用尽可能小。然而,实际的多播应用中,并不允许在多播会话期间其会话过程频繁被中断,因此本文所设计的算法并不是在每次组成员变化后重构树,而是寻找一个最佳时刻(阈值),当多播树的质量下降到指定程度时,便触发重构算法为新的成员组重新构造多播路由树,以此减少多播中断次数。文中同时考虑了基于分枝的损坏度和质量因子设置重构阈值的两种方法的优点,给出了一个更合理的质量因子 - 平衡因子(Balancing Factor BF)结合文献[3]中的思想,将质量因子定义为:

$$BF(R) = \frac{Ufn(R)}{Dmg(R)} \quad (7)$$

在执行树重构操作时,本文采用自适应遗传算法重构受损区域的多播树。这里采用自适应遗传算法的原因在于,遗传算法在进化中容易丧失多样性,从而产生早熟现象。一般来说,种群数的大小对算法的收敛性能影响很大。本文采用的种群控制策略是先将种群数分为三个等级进行宏观调控,同时对每个个体赋予适应度表示的寿命,用个体寿命再对种群的大小进行微观调控,从而形成一种种群数目随进化过程解的搜索情况自适应调整的遗传算法。采用这种方法可以避免单纯采用遗传算法所存在的早熟及不成熟收敛问题。本文采用的种群数自适应控制策略与交叉和变异概率的自适应控制策略,作者在文献[4]已有详细阐述,本文不再重复。

4 算法设计

4.1 节点加入处理设计

节点加入处理过程的设计集成了路由、接纳控制(CAC)和资源预留三者的功能。节点控制多播路由时以完全分布式的方式构建多播树。每个节点执行相同的算法。从源节点开始执行该协议。当有新节点加入时,多播树进行扩展。在多播会话期间,经常并发运行多个路由计算进程,各进程以其 ID 号相区别。节点加入算法的具体描述如下:

不同的节点(源节点,树分枝节点,中间节点和目的节点)对所收到的信息做出不同的反应。节点的处理过程由所收到的控制信息触发,这些控制信息包括加入请求信息、fork 请求信息、建立请求信息、确认/否

认信息和链路释放信息。应用程序发出一个带 QoS 参数(即要求的带宽及时延约束)的多播会话请求之后,多播成员组就将进行初始化。运行该应用程序的节点即成为多播成员组的源节点。当一个新的目的节点决定通过源节点加入多播树时,它首先查询源节点的地址,然后以单播方式向源节点发出加入请求信息。该请求信息将携带该调用过程能够接受的最长建立时间。在源节点收到该请求信息后就启动路由搜索过程,即源节点通过多播通道发送触发请求(fork - req)报文,以便触发多播树中其它节点的路由搜索进程。触发请求报文中包括连接的 QoS 要求,比如要求的带宽、端到端时延约束、连接的建立时间限制以及累积的时延等。同时源节点也在本地路由表中查找到新的目的节点的最短路径以及相应的 QoS 参数值。如果该路径满足连接的 QoS 约束条件,则源节点通过这条最短路径向加入节点发送带有路径信息的连接建立请求报文。

每个多播树的分枝节点收到触发请求信息之后,可以从该报文中获知其到源节点的累积时延值。然后就开始与源节点相同的操作,同时要检查到加入节点的最短路径与其相关的 QoS 信息以及连接的建立时间限制。如果满足约束条件,就沿该最短路径向加入节点发出连接建立请求报文。该报文中包括了请求的 QoS 参数以及该最短路径的 QoS 参数值,比如路径的最小可用带宽,源节点到该多播树节点的累积时延。连接请求报文每经过一个中间节点,该节点就检查建立时间限制和可用的资源,更新遍历路径的 QoS 参数值。在不超过连接建立时间的限制和节点具备可用资源的前提下,连接建立请求报文处理过程将持续下去,直到不满足上述条件时,该报文将被丢弃。

在这期间,目的节点将一直等待连接建立报文的到达。当每个连接建立报文到达加入节点,其携带的路径信息将被存入节点的本地数据库。当加入节点收到第一个连接建立报文之后,就启动连接配置过程,从本地数据库的候选路径中选择一条可行路径,这条路径一般是满足时延约束的具有最大可用带宽的可行路径。然后沿着选定的路径发送确认信息,启动反向资源预留过程。

资源预留过程一直持续进行,直到到达任意多播树节点为止,至此整条加入路径上的资源预留完毕,连

接建立过程结束。在此过程中,在某个中间节点(非原有多播树的节点),由于不满足接纳控制条件,比如不满足带宽或者时延要求,而致使资源预留过程失败,则该节点会沿原路返回向加入节点发送否认信息。收到否认信息的节点会释放先前预留的资源。当加入节点收到该否认信息后,会触发加入节点再从本地数据库中选择下一条可行路径。否认报文中将携带先前链路建立过程中所遇到的阻塞链路的列表信息。为了避免再次选择这样的链路,在选择下一条备选链路时,将利用这些阻塞链路的信息。连接配置过程将持续进行直到遇到下述三种情况:加入节点成功加入多播树;没有其它的可选路径;连接建立计时时间耗尽,连接超时。确认报文一旦到达接入多播树的节点,资源预留过程即告结束。反过来说,加入节点在收到第一个多播数据报文之后,即可确信连接建立过程已经顺利完成。

该算法的路由决策建立在最新的可行路径状态信息的基础上。这使得该算法可以避免产生拥塞的路径,而且所选的路径较易被接受。此外该协议的连接建立时间较短,使其适合在高速通信网中使用。

4.2 节点离开处理设计

在动态多播组成员中,节点允许随时加入或者离开多播会话。当某个节点 x 要离开多播会话时,即向多播源节点发送释放报文。此时,先将该节点标记为删除节点,如果请求离开的节点是多播树中的叶子节点,则将多播树中只为该节点进行服务的部分路径删除。部分路径定义如下:在路径中,设成员节点 u 最靠近节点 x ,并且 $u \in M$ 或者 $\deg_T(u) > 2$ 。这样在节点 x 和节点 u 之间的路径 $PT(x, u)$ 被删除。在这一步,节点 x 被删除,并且包含该节点的部分路径被裁剪。否则,不能从多播树中移出该节点,这是因为该节点还要给其它多播成员节点转发数据报。在这种情况下,将该节点保留到多播树中,此时它就转变成一个纯粹的转发节点,仅具备转发功能。至于该节点离开多播组后是否重构树,需要对 BF 值进行判断后决定。

4.3 多播树重构算法

4.3.1 重构算法步骤

当一个成员节点 x 离开,该节点成为标记节点,相应地,产生一个包含该节点的标记区域 R 。一个成员离开时,是否重构,需要对区域 R 的 BF 值进行判断:

- ① 如果 $BF > \rho$,不触发重构算法

- ② 如果 $BF > \rho$,触发重构算法

这里是 ρ 一个指定的参数。当重构一个区域的时候,在节点下游的成员节点缓冲中保存的路径被用来将该成员加入到多播树。本文采用种群自适应控制的遗传算法重构区域的多播树。遗传交叉与变异算子的设计方式与作者在文献[5]中所设计的策略及算子相同,本文不再重复阐述。基于种群数自适应遗传算法的重构过程大致如下:

- ① 根据在重构区域内确定的源结点和一组目的节点产生区域多播树的初始集合 A ;

- ② 将多播树进行剪枝操作以满足给定的条件;

- ③ 将②的结果进行编码(采用树形编码方式),产生初始进化群体;

- ④ 个体适应度评价,主要包括:

- 1) 分别计算个体的子目标函数值;
- 2) 将个体排序等级作为原始适应度;
- 3) 计算个体的浓度;
- 4) 计算个体的适应度;

- ⑤ A 中的最优个体保存到基因池,其余个体执行二进制锦标赛选择操作后保存到基因池。整个基因池中染色体数目等于种群规模;

- ⑥ 对种群数进行宏观调控,确定进化种群的最大数量;

- ⑦ 个体年龄增加 1,并对种群数进行微观调控,去除种群中年龄超过寿命的个体;

- ⑧ 交叉操作:根据作者在文献[4]中提出的自适应策略确定交叉概率,从基因池中选择两个个体进行交叉操作后生成种群 B ;

- ⑨ 变异操作:根据本文提出的自适应策略确定的变异概率对交叉后的个体进行变异操作后生成种群 C ;

- ⑩ 遗传操作终止条件判断,如果遗传代数 $<$ 设定代数(转步骤 4),否则输出结果(即重构后的多播树),算法终止。

4.3.2 个体初始化及编码设计

首先确定重构区域的源节点和一组目的节点,然后采用随机深度优先搜索算法产生从源节点 s 到这组目的节点的路径集。它可以保证形成到各目的节点的初始多播树解决方案。对初始多播树进行剪枝操作形成到给定目的节点的多播树。遗传操作染色体的个体

编码由该多播树上各边的序号序列组成。这种编码方式简单,也有利于降低遗传操作的复杂性,提高算法的执行效率。

4.3.3 个体评价及适应度函数设计

分别对个体计算预先定义的目标函数值,计算个体原始适应度(Fitinit)是对当代群体内个体相互之间的优劣关系进行排序,给所有个体分配一个等级值。以表示其在群体中的优劣顺序,个体等级值与其原始适应度值相一致,且值越小则个体性能越优良。其次,需要使用个体抗体浓度来保持多目标进化群体的多样性,它是实施有效多目标搜索的关键环节。本文使用基于 Shannon 的信息熵理论来保持群体多样性^[7,8],它不存在参数估算与敏感问题。

设多播目的结点数为 m , 计算个体 i 的抗体浓度。首先,个体的路径字符串信息熵为:

$$H_k(2) = -P \times \ln P \quad (8)$$

其中, P 为与某个体 j 相比较,个体 i 中包含至第 k 个目的结点路径相同的概率。

其次,计算个体 i 和 j 的平均熵:

$$H_{i,j}(2) = \frac{1}{m} \sum_{k=1}^m H_k(2) \quad (9)$$

再次,计算个体 i 与 j 的相似程度 $S_{i,j}$ ($S_{i,j} \in [0, 1]$)

$$S_{i,j} = \frac{1}{1 + H_{i,j}(2)} \quad (10)$$

计算个体 i 在群体中的抗体浓度 C_i :

$C_i = (\text{群体中与 } i \text{ 的相似程度大于 } \lambda \text{ 的个体数目}) / (\text{群体规模})$ ($\lambda \in [0.9, 1]$)

最后根据个体初始适应度 Fitinit 计算个体 i 的聚合适应度 Fit_i :

$$Fit_i = Fit_{init} \times C_i \quad (11)$$

5 算法仿真结果及性能分析

本文算法的网络拓扑采用 Waxman 模型生成,节点分布在(1000,1000)的网格中,节点度控制在 4 左右,这与实际情况相似。节点矩形区域中,每对节点 (u, v) 之间的距离采用欧拉距离 $d(u, v)$ 描述,两个节点 u, v 之间存在边的概率由下面公式决定:

$$p(u, v) = \beta \exp(-d(u, v) / 2\alpha a) \quad (12)$$

其中 $d(u, v)$ 为节点 u 到节点 v 之间的欧拉距离, α 控制短边与长边的比值, β 控制边的密度, α 越大,则

长边与短边的比值也越大, β 越大,边越密集, n 为网络图中节点数目。新节点加入请求概率由下式决定:

$$Prob(add) = \frac{\gamma(n_1 - m)}{\gamma(n_1 - m) + (1 - \gamma)m} \quad (13)$$

$\lambda \in (0, 1]$ 内的常数

其中, m 为多播树成员数,删除请求概率

$$Prob(remove) = 1 - Prob(add) \quad (14)$$

仿真实验中,在节点数从 10 递增到 100 的情况下,对本文算法、Greedy 和 LeastHop 算法的路由请求成功率、平均延迟和平均跳数进行了 100 次测试比较,实验结果如图 1~图 3 所示。由图 1 知,在各节点数目下,本文算法的路由请求成功率比 Greedy 和 LeastHop 算法的高,主要是因为本文算法在路由选择上满足了延迟等 QoS 的要求,而另两种算法没有考虑 QoS 因素,因此路由不满足 QoS 约束的可能性降低了路由请求的成功率。由图 2 知,在各节点数目下,本文算法的路由延迟比 Greedy 和 LeastHop 算法的平均低 20,主要是因为 DMRMQ 满足延迟约束。由图 3 知,LeastHop 算法的路由平均跳数最小,本文算法的路由平均

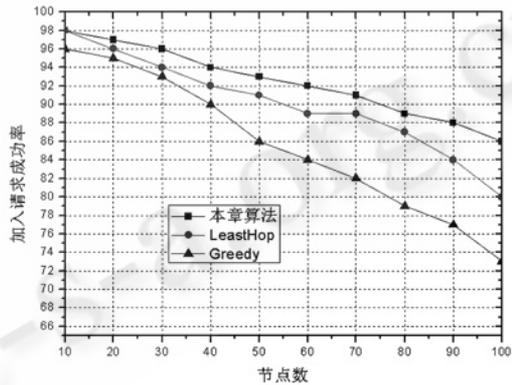


图 1 加入请求成功率与节点数关系

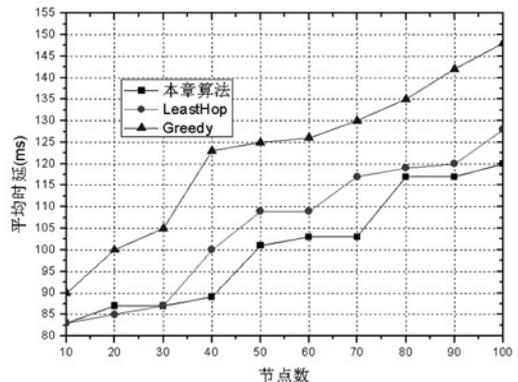


图 2 平均时延与节点数的关系

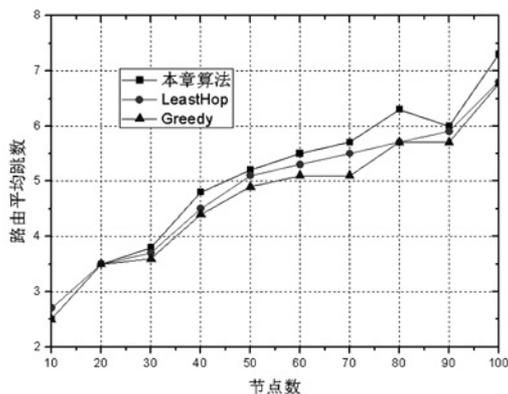


图 3 路由平均跳数与节点数的关系

跳数稍大,这主要是因为本文算法在某些情况下可能需要牺牲一定的跳数,以满足多 QoS 约束。总之,在假定网络节点输出链路容量相等的情况下,本文算法与没有考虑 QoS 约束的 Greedy、LeastHop 两种动态多播路由算法相比,路由平均跳数相差甚微,但却表现出更好的路由搜索和延迟性能。

6 结语

本文在分析动态多播路由特点的基础上,提出并设计了一种具有 QoS 约束的动态多播路由算法,该算法在新节点提出加入请求时,可通过分布式方式搜索到多条可行的路径,且能选择较优的路径将新节点连接到多播树,而且该算法能有效地减少生成多播树的开销。在节点离开时,采用平衡因子 BF 评价多播树一个区域的质量。然后根据 BF 的值可以确定是否重构

多播树或者多播树的一个区域。在重构过程中,采用种群数自适应的遗传算法搜索满足 QoS 约束要求且费用最小的多播树。仿真实验结果表明,这种结合分布式和集中式特点的动态多播路由算法为解决 QoS 约束的动态多播路由的研究提供了一种新的有效途径,算法具有较好的可扩展性和综合性能。具有较好的应用前景。

参考文献

- 1 Kwon G, Byers J W. Smooth multirate multicast congestion control. IEEE INFOCOM. San Francisco: IEEE Communications Society, 2003: 1022 - 1032.
- 2 Wen WS, Mukherjee B, Chan SH, Ghosal D. LVMSR: An efficient algorithm to multicast layered video. Elsevier Computer Networks, 2003, 41(4): 363 - 383.
- 3 陈琳. 基于服务质量的多播路由算法研究. 武汉大学博士学位论文, 2005.
- 4 李昌兵, 曹长修, 李银国, 余义斌. 一种基于 QoS 约束的组播树构建算法. 信息与控制, 2008, 1(37): 81 - 86.
- 5 李昌兵, 曹长修, 余义斌. 基于遗传算法与模糊选择的多播路由优化. 系统仿真学报, 2007, 19(21): 4940 - 4944.
- 6 崔逊学, 林闯. 基于多目标遗传算法的多播服务质量路由优化. 计算机研究与发展, 2004, 41(7): 1144 - 1150.
- 7 J Kapur, H Kesavan. Entropy Optimization Principles with Applications. San Diego: Academic Press, 1992.