

Ad Hoc 网络中基于多条件的最优路径选择

张书奎 崔志明

(苏州大学计算机科学与技术学院 苏州 215006)

摘要 在无线移动 Ad Hoc 网络(mobile Ad Hoc network, 简称 MANET)中,提供端到端 QoS 保证的关键问题之一是确定一条可行的、满足一组 QoS(如带宽、延迟抖动和丢失率)约束的路径。一般说来,寻找一条满足多个附加约束条件路径的问题,是 NP 完全问题,这导致了許多启发式算法的提出。本文提出了一种在 Ad hoc 网络中基于启发式多条件约束的最优路径选择算法 MCOPS(Multi-Constrained Optimal Path Selection),此算法首先裁减掉那些从源节点到目的节点不可行的路径,然后,在每条可行路径上使用 $K+1$ 次 Reverse-Dijkstra 算法,随后,在图上随机地移动,发现那些到达最终目的节点的机会较大的节点。MCOPS 算法由于是利用局部状态信息分布地计算,使其可方便地进行扩展到其他 QoS 约束的路由问题。仿真实验结果表明,该算法能够以较小的代价开销获得较高的路由成功率。

关键词 多约束,启发式,自适应,移动 Ad Hoc 网络,QoS 路由

Multi-Constrained Optimal Path Selection in Ad Hoc Network

ZHANG Shu-Kui CUI Zhi-Ming

(Computer Science & Technology College, Soochow University, Suzhou 215006)

Abstract One of the key issues in providing end-to-end Quality-of-Service(QoS) guarantees in mobile Ad Hoc network is how to determine a feasible route that satisfies a set of QoS requirements(such as bandwidth, delay, delay-jitter, and loss) of a connection request. In general, finding a path subject to multiple constraints such as delay and delay-jitter is a NP-complete problem. Multiple constrained QoS routing optimization is an important problem in the current Ad Hoc communication network research. In this paper, we propose heuristic MCOPS (Multi-Constrained Optimal Path Selection) for this problem, the algorithm first prunes all the links that cannot be on any feasible path from resource to destination, it then uses a randomized search to find a feasible path, in order to achieve efficient utilization of network resources, the algorithm tries to select a path with minimum-delay among all feasible paths. Simulation results show that MCOPS algorithm performs much better than the Jaffe routing algorithm under the circumstances of mobile Ad Hoc networks.

Keywords Multiple constraints, Heuristic, Adaptive, Mobile Ad Hoc networks, QoS routing

在 Ad Hoc 网络中,节点的移动性和网络拓扑频繁变化的特性,使得彼此间的连接易于断裂,从而导致一些节点处于非连接状态。当两个要通信的节点不在它们的信息传输范围内时,恰好有信息需要传递给这些节点,则会导致信息传送失败,另外,网络的无中心设施和自组织特性又加大了网络安全保护的难度,多跳路由、动态拓扑、带宽和能量限制等特性又很难保证网络的 QoS^[4~7]。Ad Hoc 网络由于其独有的特点,若要找到一条可行路径,使其满足一系列端到端的 QoS 约束,同时有效的利用网络资源,就是一个很困难的问题^[1~3]。一般说来,寻找一条满足多个附加约束(例如,延迟、延迟抖动)路径的问题,是 NP 完全问题^[1,2],这导致了許多启发式算法的提出。本文在对已有随机化算法的分析基础上,提出了一种基于多条件约束的最优路径选择算法 MCOPS(Multi-Constrained Optimal Path Selection):初始计算从每个结点到给定的目的节点的代价,随后,在网络拓扑图上随机地移动,发现那些到达最终目的节点的机会较大的节点,由于 MCOPS 具有利用局部状态信息分布地进行计算的特征,使其具有可扩展到其他 QoS 约束的路由问题。

1 基本术语和算法

Ad Hoc 网络表示为带权图 $G(V, E)$, 这里 V 是结点集

合, E 是边集合。源结点 s , 目的结点 t 。网络中的每条边都与多个 QoS 参数相关, 每条边上定义一个 K 维权向量, 包括 K 个非负 QoS 权值 $(W_k(i, j), i=1, \dots, K, (i, j) \in E)$ 。这些参数可以严格分为附加的(additive)与非附加的(non-additive)。对于附加的权值(如分组丢失率、延迟、延迟抖动、连接建立时间等), 路径的 QoS 值等于路径上各条边的相应权值之和。对于非附加的参数(如带宽), 路径的 QoS 值是路径中最小(或最大)边的权值, 约束表示为 $C_k, k=1, \dots, K$ 。一般说来, 通过从图中除去所有不满足 QoS 约束的边, 以及可能的不连通结点, 能够较容易地处理与非附加的参数相关的约束问题。处理与附加参数相关问题的方法则要困难得多。因此, 不失一般性, 这里考虑附加的方法。

定义 1 多约束路径(Multi-constrained path, 简称 MCP)问题 考虑一个网络 $G(V, E)$ 。每条边 $(i, j) \in E$ 与 K 个附加权 $W_k(i, j) \geq 0, i=1, \dots, K$ 相关联。给定 K 个约束 $C_k, k=1, \dots, K$, 问题就是: 找到一条从 s 到 t 的路径 p , 使得: $w_k(p) \stackrel{\text{def}}{=} w_k(i, j) \leq C_k, k=1, 2, \dots, K$ 。

定义 2 多约束最优路径(Multi-constrained optimization path, 简称 MCOP)问题 在定义 1 的路径 P 上增加一个条件: 对于 s 与 t 之间的任何可行的路径 $Q, L(P) \leq L(Q)$, 这里 $L(\cdot)$ 是一个路径长度(或代价)函数。

* 基金项目:江苏省高校自然科学基金研究计划资助项目(编号:03KJD520191)。张书奎 副教授,主要研究方向网络智能信息处理、数据通信等。
崔志明 教授,博士生导师,主要研究方向为智能化信息处理、数据挖掘及计算机网络应用。

可以看出,符合定义 1 条件的一条路径即为可行路径。满足定义 2 就是从可行路径集合中找出“长度”最短的路径, MCOP 问题的解也是 MCP 问题的一个解,但反之未必。当 K 超过 2 时, MCP 问题是 NP 完全问题,此问题在大型网络中被认为是难于处理的^[1,2]。

初始时,算法根据权值(如分组丢失率、延迟等) W_k 以及这些权的线性拟合 $W_1 + W_2 + \dots + W_k$, 计算每个结点 u 到目的结点 t 的最优路径的代价。这个初始化过程可以通过执行一次 Reverse-Dijkstra 算法来实现,其计算复杂度是 $O(n \log(n+m))$ ^[6]。接着,算法从源结点 s 开始,搜索落在评估的可行区域内的每个相邻结点。每个结点所在的区域,是根据一条端到端路径的已搜索部分的代价以及剩余最优部分的代价来计算的。随后随机选择一个已搜索的结点,通过它来到达其他的结点,由于它的随机化本质,可以进行多次搜索,以提高找到可行路径的可能性, γ 是搜索的最大次数,其伪码如下所示。

算法 1.1 MCOPS

- (1) 初始时,利用 Reverse-Dijkstra 算法对所有节点计算 $B_k[u, t], k=1, 2, \dots, K$ and $L[u, t]$ 。
- (2) 对 $(G=(V, E), s, t, C_k, k=1, 2, \dots, K)$, 如果 $B_k[s, t] > C_k, k=1, 2, \dots, K$, 则无满足条件的路径。
- (3) 如果 $L[s, t] > \sum_{k=1}^K C_k$ 则无满足条件的路径。
- (4) 尝试 γ 次,通过递归调用算法 1.2 MCOPS-BFS 发现一个可能的路径。

算法 1.2 MCOPS-BFS

- (1) 采用广度优先,对除初始节点 s 以外的每一个节点 u , 计算 $D_k[u] = \infty, \pi[u] = \text{NIL}$ (为空), $D_k[s] = 0, \pi[s] = -1, Q = \{s\}$ 。
- (2) 在 Q 中随机选择一个节点 u , 对每一个 $v \in \text{Adj}[u]$ (邻接点),
 如果 $\pi[v] = \text{NIL}; D_k[u] + W_k[u, v] + B_k[v, t] \leq C_k \forall k$;
 并且 $(\sum_{k=1}^K D_k[u] + W_k[u, v]) + L[v, t] \leq \sum_{k=1}^K C_k$, 则
 $D_k[v] = D_k[u] + W_k[u, v] \forall k; \pi[v] = u; Q = Q \cup \{v\}$ 。
 其中,对结点 u 设置标记: $B_k[u, v]$ 表示根据边权 W_k , 从 u 到每个可能的目标结点 v 的最短路径的代价。 $L[u, v]$ 与 $B_k[u, v]$ 相似,但它是根据所有权值的线性拟合 $W_1 + W_2 + \dots + W_k$ 所得出的最优值。 $D_k[u]$ 表示根据边权值 W_k , 从 s 到 u 的路径的代价, $k=1, 2, \dots, K$ 。算法将 u 的前继结点存放在数组 $\pi[\]$ 中,若 u 没有前继结点,也就是它还没有被发现,则 $\pi[u] = \text{NIL}$ 。

为说明算法如何判断是否存在一条可行路径,考虑图 1(a) 中的网络。图中每条边附有两个权 (w_1, w_2)。从 s 到 t 有五条连接 a, b, c, d, e, 约束值如图 1(b) 中圆圈所示, 根据各个权值所得的最优代价,在图中用黑色正方形表示。沿实线所示的路径,是根据边权值的线性拟合所得的最优路径。阴影部分表示绝对拒绝区域,不是阴影的部分表示可能性评价区域。MCOPS 对条件进行检查,算法就可以确定 a, b, c 不可行, d, e 可能存在可行路径(即便没有路径满足 d 的要求)。

若请求没有落在绝对拒绝区域,算法调用 MCOPS-BFS 广度优先搜索(Breadth-First Search)来随机查找一条可行的路径。由于它的随机化的本质,可以进行多次搜索,以提高找到可行路径的可能性。设 γ 是搜索的最大次数。通过模拟可以知道, $\gamma=1$ 时,算法找到可行路径的成功率很高。实际上,进行多次尝试所带来的性能上的提高可以忽略不计,例如, γ 由 1 增加到 5,性能仅仅提高 0.5%^[3]。

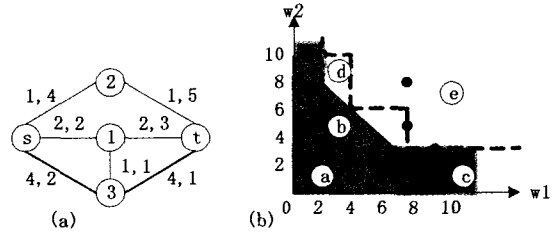


图 1 MCOPS 算法确定是否存在可行的路径

一般广度优先搜索中使用先进先出队列来管理已经发现的结点,在 MCOPS-BFS 中使用一个随机队列 Q 来达到同样的目的。MCOPS-BFS 从 Q 中随机地选择一个结点 u , 并尝试搜索 u 的邻接表中的每一个结点 v 。BFS 遍历从 s 可达的每一个结点,而 MCOPS-BFS 则遍历那些有可能到达 t 的结点。算法还记录了从 s 到 u 这一段已搜索部分边 (u, v) 的代价,根据每条边的权值和权值的线性拟合所得出的、未搜索部分中从 v 到 t 的最优代价。如此 MCOPS-BFS 就可以确定一个绝对拒绝区域,如同图 1(b) 中所示。如果一个连接请求落在这个区域之外,那么经由 v 就有可能到达 t , 于是 MCOPS-BFS 从 u 搜索 v , 更新 $D_k[v], k=1, 2, \dots, K$, 以及 $\pi[v]$, 并将 v 放入 Q 中。当 t 被找到时,或者 Q 为空时,搜索结束。

2 例子

考虑图 2 中的网络, 每条边都是双向的, 由两个附加权 (W_1, W_2)。这个例子中边是对称的, 但算法也可以在带有实数权值的不对称的边上运行。假设我们要寻找从 $s=0$ 到 $t=4$ 的一条路径, $C_1=13, C_2=12$ 。设每个结点 u 的标记 $B_k[u, 4], k=1, 2$, 及 $L[u, 4]$ 已经算出来, 如图 2 所示。由于 $B_1[0, 4]=10 \leq C_1, B_2[0, 4]=6 \leq C_2, L[0, 4]=2 \leq C_1 + C_2=25$ 算法继续进行, 以寻找可行路径。MCOPS 调用 MCOPS-BFS, 它的执行用虚线表示。MCOPS-BFS 随机地从 Q 中选择 u , 搜索 u 的邻接表中的结点。默认情况下, $s=0$ 首先被搜索, 并被放在 Q 中。由于 $Q=\{0\}$, MCOPS-BFS 选择结点 0, 搜索它的邻接结点 1 和 7, 并把它们放在 Q 中。假设 Q 从 Q 中随机选择结点 1。随后发现结点 8, 从而 $Q=\{7, 8\}$ 。由于 $D_2[1] + W_2(1, 2) + B_2[2, 4] = 1 + 8 + 5 = 14 > C_2$, 不能发现结点 2。接下来, 随机选取结点 8, 从 8 发现 9, 导致 $Q=\{7, 9\}$ 。随后, 假设随机选择 9。在此结点, MCOPS-BFS 预测到, 若从结点 9 发现结点 5, 由于 $D_1[9] + W_1(9, 5) + B_1[5, 4] = 9 + 3 + 2 = 14 > C_1$, 则没有机会到达目标结点。然而, 算法发现结点 3, 从而 $Q=\{7, 3\}$ 。设 MCOPS-BFS 随机选取结点 3, 由 3 发现目的结点 4。在这个结点上, 算法停止, 返回可行路径 $p=(0, 1, 8, 9, 3, 4)$ 。

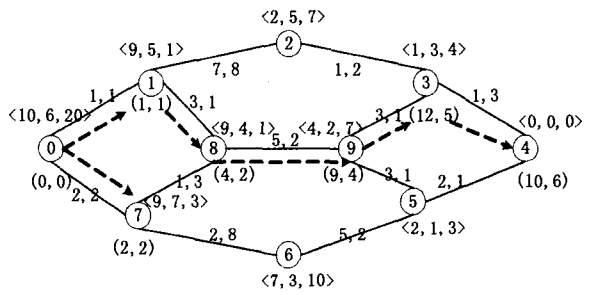


图 2 MCOPS 算法发现一条可行路径

3 性能分析

若对 Q 中的结点排序,并按照一种对 Q 中排在前面的结点有利的方式进行选择,可以提高随机搜索的性能。设定启发式选择函数为计算给定的约束与每个结点处的可行性区域之间的面积,面积越大,找到一条可行路径的可能性就越大。考虑图 2 中的例子,设要寻找从 $s=0$ 到 $t=4$ 的一条路径,算法从 s 发现结点 1 和 7,不同于前面的从 Q 中随机选择一个结点,算法首先根据上述的启发式函数,对结点进行排序。结点 1 和 7 的可行性区域如图 2(a)和(b)中所示。算法确定这些区域的过程如下:在结点 1 处,由于 $W_1(s,1)=1, B_1[1,t]=9$,从 s 经由结点 1 到达 t 的最优的 W_1 是 $1+9=10$;由于 $W_2(s,1)=1, B_2[1,t]=9$,从 s 经由结点 1 到达 t 的最优的 W_2 是 $1+5=6$;由于 $L[1,t]=18$,从 s 经由结点 1 到达 t 的最优的 W_1+W_2 是 $1+1+18=20$ 。在结点 7 处,由于 $W_1(s,7)=2, B_1[7,t]=9$,从 s 经由结点 7 到达 t 的最优的 W_1 是 $2+9=11$;由于 $W_2(s,7)=2, B_2[7,t]=7$,从 s 经由结点 7 到达 t 的最优的 W_2 是 $2+7=9$;由于 $L[7,t]=18$,从 s 经由结点 7 到达 t 的最优的 W_1+W_2 是 $2+2+18=22$ 。图 2(a)和(b)中分别表示了结点 1 和 7 处,给定的约束与可行性区域之间的面积。这些区域分别是 10 和 4,如图 2 粗实线所围区域。因此,结点 1 比结点 7 有更大机会被选择。

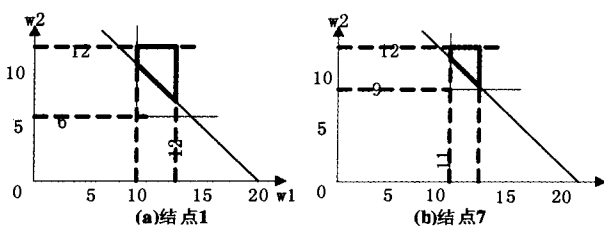


图 3 计算约束与试验可行区域之间的面积

MCOPS 算法可以用分布式的方式来实现。对于每个目的结点 v ,每个结点 u 保留标签 $B_k[u,v], k=1,2,\dots,K, L[u,v]$ 。这些标签类似于现有的分布式算法中的距离向量,可以根据每个边权 $W_k, k=1,2,\dots,K$,以及这些权值的线性拟合,用同样的程序来计算。在源结点处,算法首先建立一个路径请求消息 $path_request$,这个消息包括 t (目的结点)、 π (路径中已经搜索过的部分)、 D_k (路径中已经搜索过的部分的代价),以及给定的约束 $C_k, k=1,2,\dots,K$ 。随后,算法与相邻结点联系,以确定哪些结点满足试验可行区域。一旦确定了这些结点,算法就从它们当中随机选择一个结点,更新 $path_request$ 消息,并将这个消息发送给被选定的结点。当一个结点收到 $path_request$ 消息后,重复以上步骤,直到到达目的结点 t 。由于 $path_request$ 消息中保存了已经搜索部分的信息(π),只要不访问 π 中的结点,构造出来的路径就可以很容易的避免产生回路。

另外,如果同时把多个 $path_request$ 消息发送给多个邻接结点,这样就用并行的方式分别沿不同的路径进行搜索,以增大源结点找到可行路径的可能性,并能确定多条路径。

在算法的初始化步骤中,根据权 W_k 以及这些权的线性拟合 $W_1+W_2+\dots+W_k$,计算每个结点 u 到目的结点 t 的最优路径的代价的工作,也可以并行的加以解决。对每一条路径,执行一次 Reverse-Dijkstra 算法,是单源最短路径问题,可以采用 Moore 算法的并行化方式来代替^[1,2]。

4 仿真实验

为了评价 MCOPS 算法,本文使用 NS-2 进行了仿真实验,对 2 个主要的性能指标:路由成功率、平均端到端延迟进行检验,仿真实验环境的网络拓扑如图 4,也是 Jaffe 算法^[3]中使用的网络拓扑,32 个移动节点被随机地放在 $15 \times 15m^2$ 的区域内;节点的发射范围被限制在以 r 为半径的圆内, r 均匀分布于 $[0.5,6]m$,节点的移动速度均匀分布于 $[0.5,3]m/s$ 。当两个节点处在彼此的发射范围内时,相互之间就存在一条链路。

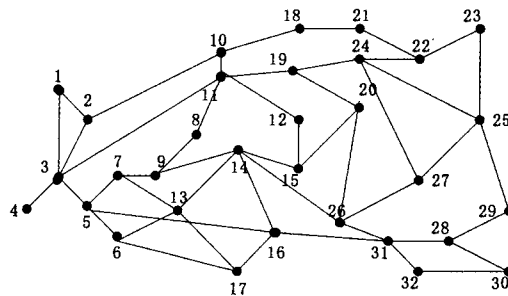


图 4 一个不规则的网络拓扑图

MAC 层选用 IEEE 802.11 协议,选择链路带宽为 2M bit/s,发包速率为 400 packet/s,仿真通信采用大小为 512 B 的定长数据包,每个连接请求的源节点、目的节点随机产生,通信持续时间为 300s。

每个报文对象的参数主要包括产生时间、死亡时间、抛弃标志、源地址、目的地址和优先级。所有的报文参数均初始化为 -1,在仿真过程中被动态设置和修改。这里优先级表明报文要求的类别,本文的仿真中优先级只有“高”、“低”两种,分别代表了高服务质量要求或低服务质量要求。所有仿真中,每个新生成的报文以率 0.25 被赋予高优先级,以概率 0.75 被赋予低优先级。

设置两个链接权重: $w1(I,j)=range[0,50], W2(I,j)=range[0,200]$ 。对于 C1 和 C2 的不同范围值,采用 MCOPS 算法和 Jaffe 算法^[3]分别计算 20 次,每次运行随机产生 100 次连接请求,获得路由成功率与平均端到端延迟如表 1 所示。实验结果显示,MCOPS 算法的平均端到端延迟比 Jaffe 算法平均减少了 7.94%,而且可以发现,当网络中的“繁忙”节点逐渐增多时,MCOPS 算法和 Jaffe 算法相比,平均端到端延迟的减少量也不断增大。另外对路由成功率的变化,MCOPS 算法在动态环境中同样比较好的性能,与 Jaffe 算法相比平均提高了 10.5%。

表 1 两个约束条件下的仿真结果

C1 和 C2 的范围	路由成功率 MCR/Jaffe	平均端到端延迟 MCR/Jaffe
C1=Range[100,115] C2=Range[400,470]	0.5834/0.5226	2.4106/2.7615
C1=Range[50,65] C2=Range[200,260]	0.2546/0.2289	1.5412/1.7865
C1=Range[75,90] C2=Range[300,360]	0.7812/0.7103	2.0023/2.1721
C1=Range[125,140] C2=Range[450,510]	0.7684/0.7012	3.0812/3.2529
C1=Range[160,180] C2=Range[610,640]	0.6825/0.6198	3.0952/3.2869

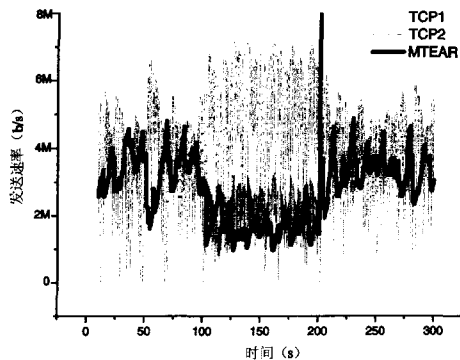


图 10 TEARM 和 TCP 的吞吐量

2.3.2 对低速带宽的响应性

采用哑铃型拓扑。TEARM 有两个接收端,接收端 R0 与一条 TCP 流竞争 8M 带宽。在 100s 时,接收端 R1 加入 TEARM 组播组,并且与一条 TCP 流竞争 4M 带宽,而该接收端在 200s 退出组播组。

由图 10 可以看出,低速的 R1 加入组播组后,TEARM 能够将其速率调整到合适的带宽,并且能够选出合适的代表作为当前受限接收端。TEARM 对速率的响应时间基本在 3s 以内,在 1s 内可以迅速选出 CLR。TEARM 可以对低速的接收端加入作出迅速的响应。但是当低速接收端在组播组内时,由于接收端 R0 长时间没有经历丢包,其 *cwnd* 值大于接收端 R0 实际的 *cwnd* 值。当接收端 R1 退出组播组时,因为接收端 R0 的估计速率高于其实际的带宽,导致发送端以过高的速率发送数据(图 10 中的尖峰)。但是根据接收端 R0 的反馈信息,发送速率很快就恢复到了公平带宽 4M。可以说,TEARM 对带宽具有良好的响应性。

总结和展望 TEARM 在接收端模拟了 TCP 的拥塞控制机制,而且对多个 epoch 的速率进行加权平均,估计吞吐

量,并采用了代表机制来抑制反馈。通过仿真可看出,TEARM 与 TFMCC 相比,具有相同的 TCP 公平性和速率平滑性,并且在可扩展性方面优于 TFMCC,以及能够很好地响应不同的链路丢失率和带宽,使得 TEARM 适合于流媒体业务的传输。但是,由于 TEARM 是一个单速率的组播拥塞控制协议,在某些情况下,低速的 CLR 无疑会限制其他高速接收端的吞吐量,使得 TEARM 缺乏协议内的公平性;此外,对于低速接收端迟加入的情况,高速的接收端估计速率很可能要远大于实际的链路带宽,会对协议的性能造成潜在的威胁。因此,需要将 TEARM 扩展到多速率组播拥塞控制协议中,通过引入分层机制来进一步改善协议的性能。

参考文献

- 1 石锋,吴建平. 组播拥塞控制综述[J]. 软件学报, 2002, 13(4): 1~9
- 2 Padhye V, Towsley Firoiu D, Kurose J. Modeling TCP Throughput: A Simple Model and its Empirical Validation. In: Proceedings of ACM Sigcomm, Vancouver, Canada, 1998
- 3 Widmer J, Handley M. TCP-Friendly Multicast Congestion Control (TFMCC): Protocol Specification [EB/OL]. draft-ietf-rmt-bb-tfmcc-01. txt. 2002
- 4 Widmer J, Denda R, Mauve M. A Survey on TCP-Friendly Congestion Control [J]. IEEE Network, 2001, 15(3): 28~37
- 5 Nonnenmacher J, Biersack E W. Scalable Feedback for Large Groups [J]. IEEE/ACM Trans on Networking, 1999, 7(3): 375~386
- 6 Bhattacharyya S, Towsley D, Kurose J. The Loss Path Multiplicity Problem in Multicast Congestion Control [A]. In: Proc. of IEEE INFOCOM [C]. New York: IEEE Communications Society, 1999. 856~863
- 7 Rhee I, Ozdemir V, Yi Y. TEAR: TCP emulation at receivers-flow control for multimedia streaming. [NCSU Technical Report]. Apr. 2000
- 8 ns-2 Network Simulator. <http://www.isi.edu/nsnam/ns>, 2002
- 9 Floyd S, Handley M, Padhye J, et al. TCP Friendly Rate Control (TFRC): Protocol Specification. Request for Comments (RFC) 3448, The Internet Society, January 2003

(上接第 30 页)

可以看出,由于 MCOPS 算法是一种分布式路由算法,它不需要维持全局状态,仅仅依靠局部信息选择路由,因此在 Ad Hoc 网络高度动态的环境下,能够取得更好的路由成功率。

结束语 本文提出了一种基于多条件约束的最优路径选择算法 MCOPS (Multi-Constrained Optimal Path Selection),其基本思想是:初始计算从每个结点到给定的目的结点的最优路径代价,这条最优路径与各条边的权值及它们的线性拟合相关。然后,每条路径需要使用 $K+1$ 次 Reverse-Dijkstra 算法。随后,算法在图上随机地移动,发现那些到达最终目的结点的机会较大的结点。MCOPS 算法由于其利用局部状态信息分布式地进行路由选择的特征,使其具有可扩展性,同时,也适合于 Ad Hoc 网的高度动态的环境,并且容易扩展到其他 QoS 约束的路由问题。

由于 Ad Hoc 网络的动态性、状态信息的聚集性等因素,在同一时刻,对于每一个结点而言,网络的真实状态信息并非总是可用的。对于同一对源结点和目的结点,每次执行算法时,可能会产生不同的路径。因此,需要进一步研究路径选择策略对负载均衡的影响以及如何对网络状态信息的不精确性作出补偿,实现网络中的负载分担,减少了繁忙节点的负荷,

从而延长整个网络的寿命,使得 MCOPS 在移动 Ad Hoc 网络环境下有更好的稳定性。

参考文献

- 1 Kulpes R, Van Mlegheem P, Korkmaz T, Krunic M. An Overview of Constraint-Based Path Selection Algorithms for QoS Routing. IEEE Communications Magazine, December 2002
- 2 Orda A. Routing with end-to-end QoS guarantees in broadband networks. IEEE/ACM Transmission Networking, 2002, 7(3): 365~374
- 3 Jaffe J M. Algorithms for Finding Paths with Multiple Constraints. Networks, 1999, 14: 95~116
- 4 Iwata A, Chiang C C, Pei G, Gerla M, Chen T W. Scalable routing strategies for Ad Hoc wireless networks. IEEE Journal on Selected Areas in Communications, 1999, 17(8): 1369~1379
- 5 Sivakumar R, Sinha P, Bharghavan V. CEDAR: A core-extraction distributed Ad Hoc routing algorithm. IEEE Journal on Selected Areas in Communications, 1999, 17(8): 1454~1465
- 6 Jain R, Puri A, Sengupta R. Geographical routing using partial information for wireless Ad Hoc networks. IEEE Personal Communications, 2001, 8(1): 48~57
- 7 Paul K, Bandyopadhyay S, Mukherjee A, Saha D. A stability-based distributed routing mechanism to support unicast and multicast routing in Ad Hoc wireless network. Computer Communications, 2003, 24(18): 1828~1845