

AMR: 一个基于网络最大流的 Ad-Hoc 多路径路由算法

陈跃泉, 郭晓峰, 曾庆凯, 陈贵海

(南京大学计算机软件新技术国家重点实验室, 计算机科学与技术系, 江苏南京 210093)

摘要: 移动 Ad-Hoc 网络研究中, 路由是一个关键问题. 现有的 Ad-Hoc 路由算法大多为单路径算法. 但是多路径方法可以更好地支持 QoS, 最近也受到较大关注. 在没有精确的网络拓扑结构情况下, 找出多条不相交路径是比较困难的. 本文提出了一个基于网络最大流的 Ad-Hoc 多路径路由算法 AMR (Aggregated multipath routing). 该算法可以有效地找出多条节点不相交的路径, 较大幅度地提高网络传输性能、减少网络拥塞. 经过性能测试, 表明 AMR 算法比 DSR 算法在数据传输率方面提高 20%—60%, 端对端平均延迟降低 40%—60%.

关键词: Ad-Hoc 路由; 多路径路由; AMR; 网络最大流

中图分类号: TP393.15 **文献标识码:** A **文章编号:** 0372-2112 (2004) 08-1297-05

AMR: A Multipath Routing Algorithm Based on Maximum Flow in Ad-Hoc Networks

CHEN Yue-quan, GUO Xiao-feng, ZENG Qing-kai, CHEN Gui-hai

(State Key Laboratory for Novel Software Technology, Department of Computer Science and Technology, Nanjing University, Nanjing, Jiangsu 210093, China)

Abstract: Routing is a challenge task in the mobile Ad-Hoc network because its topology changes dynamically for its nodes' movement. At present, most Ad-Hoc routing algorithms are of single path. However, more attention has been paid to multipath recently for it can support QoS better. The problem in multipath routing is that it is much more difficult to find multiple node-disjoint paths because of the lack of an accurate network topology. This paper proposes an aggregated multipath routing algorithm (AMR) for mobile Ad-Hoc networks based on the maximum-flow method, as an extension to the single path routing algorithm (DSR). This algorithm can find the node-disjoint paths effectively, improve the packet delivery performance greatly and decrease the network congestion. The results show that compared with DSR, AMR is able to improve packet delivery ratio by about 20% - 60% and reduce average delay by about 40% - 60%.

Key words: Ad-Hoc routing; multipath routing; AMR; maximum-flow

1 引言

移动 Ad-Hoc 网络是一种无需固定设备支撑的移动网络, 它在军事、紧急救援和移动办公等领域具有广泛的应用前景. 由于网络节点的移动性, 导致整个网络拓扑结构随之动态变化, 因而 Ad-Hoc 网络路由是一个关键问题. 围绕 Ad-Hoc 网络路由问题, 已开展了许多研究. 现行的路由算法大多是按需单路径路由算法, 如 DSR^[1], AODV^[2], TORA^[3]等. 按需路由算法与传统网络那种通过周期性交换路由表来更新路由表 (如 DSDV^[4]) 的方法不同, 在需要发送数据包时, 才通过泛洪策略获取从源节点到目的节点的路由. 这些算法基本上都采用单路径方式传送数据, 控制开销和网络延迟较大. 当负载较大时, 将面临网络拥塞的问题. 为此, 近年来多路径路由算法的研究引起了重视. SMR^[5] 算法着重于建立和维护最大数目的

多条不相交路径, 文献[6]算法着重于建立和维护最大数目的多条不相交路径, 文献[6]中提出了通过路径重定向来获取多条路径的方法, 文献[7]通过获取多条链路不相交来实现多路径.

在多路径路由算法中, 需要研究: 路由发现机制、数据的多路径发送方法以及从复杂网络信息中提取高效的路径集等问题. 其它研究主要侧重于前两个问题, 较少涉及有效选择多条路径的问题. 文献[8]从传统的可靠性角度出发, 寻找可靠性最高的路由路径集. 本文则从另一种角度, 利用最大流的原理, 在多项式复杂度的情况下, 寻找出指定数目的不相交路径. 我们在 DSR 基础上提出了累积多路径 Ad-Hoc 网络路由方法 AMR. 该方法可以比较有效地找出多条节点不相交的路径, 然后把数据包通过不同的路径进行传送, 以减少网络拥塞. 而且, 只有当所有的路径都失效时, 才需要重新选择路

收稿日期: 2003-06-16; 修回日期: 2004-03-05

基金项目: 国家高技术研究发展计划 863 课题 (No. 2002AA141090); 国家自然科学基金 (No. 60073029); 国家重点基础研究发展规划 973 项目 (No. 2002CB312002); 教育部高校优秀青年教师奖

由,有效地降低了控制包开销.

2 DSR 算法介绍及其问题分析

DSR(Dynamic Source Routing)是经典的基于请求/响应按需单路由算法.并且根据文献[9]表明其有较佳的性能,所以我们将 DSR 作为参考对象.

DSR 在需要路由路径时,通过泛洪发起路由请求.节点收到路由请求时,将把最先收到的路由请求重复广播出去,而抛弃此后到达的重复请求.目的节点收到路由请求时,按照寻路请求到达路径的反方法,发送回源节点.源节点收到响应后,采用该响应所使用的路径发送数据.为了节省路由请求的耗费,DSR 引入缓冲机制.在路由维护阶段,DSR 可以通过 MAC 层协议来获取网络出错.假设出错的链接为 $x \rightarrow y$,那么 x 将发送该链接出错信息到上游节点和源节点.此时, x 节点和源节点都将参与路由路径的修复工作.如果 x 节点中还包含到达目的节点的替换路径,那么 x 节点将把该替换路径告诉源节点,同时将该数据包通过该替换路径发送;否则,如果源节点也没有替换路径就需要重启泛洪路由请求,寻找新的路由路径.

但是,由于 DSR 在路由请求过程中,每个节点只保留最先到达的请求,而把其它重复请求全部丢弃掉,这将使得 DSR 中的节点只能获取一条到达目标节点的路径,而无法获取整个网络拓扑结构,无法有效地利用整个网络信息.虽然 DSR 的中间节点也可以采用替换路径的方式发送数据,初具多路径方式的雏形.但是,它毕竟未能使用多条路径同时传输数据,不能有效利用网络资源,在传输效率、端对端延迟等性能方面有待提高.AMR 就是针对这些方面的不足,提出的多路径路由算法,可以有效地提高这方面的性能.

3 AMR(Aggregated Multipath Routing) 算法

AMR 是一种基于源节点的多路径路由算法.与 DSR 不同,它在路由请求时不是随意丢弃重复的路由请求包,而是有选择地将一些有用的路由请求包的路径信息储存并响应给源节点,从而在源节点获得比较完整的网络状态信息.根据此信息,采用最大流原理找出多条节点不相交的路径.数据发送时,将数据包同时通过多条路径发送出去,以此提高网络利用率.

3.1 路由发现

当源节点需要选择到达目的节点的路由时,由源节点发起路由泛洪请求,此过程类似于 DSR.中间节点收到路由请求包,首先,判断该包的发送路径是否成环.若是,丢弃该路由请求包并结束路由包处理,否则,做以下处理以保证路由路径中不包含环路.

如果是首次收到该路由请求包,将该包重复广播出去,并记录下该包经历的跳数 $Hops_{old}$.否则,如果收到路由请求包跳数(设为 $Hops_{new}$),把满足 $Hops_{new} < Hops_{old} + hops$ 的路径记录到本节点的路由缓冲中,其中 $hops$ 是可以容忍的额外跳数.

判断该节点是否是目的节点.若是,延迟一小段时间,以获取其它的路由路径信息.然后将这些缓冲的路由路径打包,

并通过最短路径响应回源节点.同时,通过信息给非最短路径上的上一跳地址,以期响应.

当中间节点或者源节点收到响应路由包,把获取的整条路径(图 1(a))拆分成单跳链接(图 1(b))存放本节点的网络拓扑图中,并对每段链接设置一个超时失效计时器和对每段链接数进行有效计数,以使这些链接保持有效性.由此相关节点上都将保留有相关的网络状态信息.

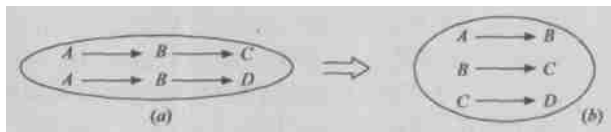


图 1 (a) 链路拆分前的情况; (b) 链路拆分后的情况

当源节点收到第一个响应路由包后,延迟一小段时间,以获得大量的网络状态信息,然后使用多路径寻找算法获得多条路径,并通过这些路径并发地发送数据.寻找多条路径的算法将在下一节中详细讨论.

3.2 路由维护

当中间节点检测到某一链接断开时,如果有替换路径,通过该路径转发数据.然后通知源节点和上游的中间节点.但是,此后路径选择权仍然在源节点上,由源节点决定以后数据发送的路径.这与 DSR 替换路径的方式不同,因为由源节点选择路径有利于建立多条节点不相交路径.

3.3 DSR 和 AMR 算法粗析

定理 1 AMR 的空间复杂度为 $O(K * M)$,其中 K 为路径条数, M 为最大路径长度.

证明 由于一条路径所占用的空间为 $O(M)$,而所寻的总共路数有 K 条路径,因而我们可求得占用空间为 $O(K * M)$;证毕.

通过模拟结果可知 K 一般为 3 到 4 条.因此,在空间开销上,AMR 开销比 DSR 稍有增加,但其复杂度是可扩展的,在最坏情况下为 $O(|V|^2)$,所以额外的开销是可容忍的.

在网络传输开销方面,主要是控制包开销,虽然 AMR 每次路由请求的开销比 DSR 有所增加,但是由于多路径的路径总的存活时间较长^[7],所需路由发现的次数减少,模拟结果也表明 AMR 在网络传输开销有较大的优势(详细讨论见 5.2 节).

4 AMR 中的路由路径选择算法

4.1 节点不相交性

路径之间的不相交可以分为链路不相交和节点不相交两种:链路不相交指任意两条路径间没有共同的链路,而节点不相交指任意两条路径间,除了源节点和目的节点之外,没有其它共同的节点.

4.2 路径选择算法

我们采用基于最大流原理^[10]的路径选择算法.例如在图 2 中,当前路径集为 $\{s \rightarrow B \rightarrow C \rightarrow t\}$.通过 Edmonds-Kap 算法^[10]找到增广路径 $s \rightarrow A \rightarrow C \rightarrow B \rightarrow D$

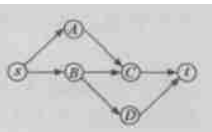


图 2 增广路径示例

其中 $C \rightarrow B$ 是反向链路.从而可以

扩展当前路径集为 $\{s A C t, s B D t\}$ 。我们的算法中也将依据这样的原理构造 k 条节点不相交路径。

下面我们分别介绍网络拓扑图拆分算法和路由选择算法,由此二算法可以构造 k 条节点不相交路径。

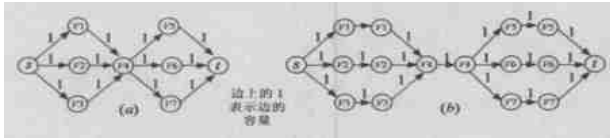


图3 (a) 拆分前的网络容量图; (b) 拆分后的网络容量图

算法1 网络拓扑图拆分算法

输入:网络拓扑图 G (图3(a))

输出:拆分后的图 G_1 (图3(b))

根据图 G 的节点数目 n , 构造图 G_1 的节点数目 $n_1 = n * 2 - 2$;

除了源节点和目的节点, 其它节点均一分为二, 设节点 x 拆分为节点 x 和 x' ;

任意节点, 如果被拆分, 其拆分后节点间构建有向边 $x \rightarrow x'$ (在 G_1 中);

对于图 G 中的边 $x \rightarrow y$, 在图 G_1 中构造边 $x' \rightarrow y$;

算法2 基于最大流原理的路由选择算法

输入:网络状态拓扑图 G

输出:多条路径集合 paths

根据算法1, 拆分源节点的网络拓扑图 G 为 G_1 ;

while (true) do begin

使用 Dijkstra 算法找到一条从源节点到目的节点的最短增广路径;

如果未能找到这样的路径, 跳出循环、寻路结束;

添加增广路径到当前的流量图 G_2 中;

残余网络处理;

end

根据获得的流量图 G_2 构造源节点到目的节点的路径集 paths.

While (true) do begin

(1) 设置源节点为当前节点 x , 设置当前路径 path 为空;

(2) 如果 x 为目的节点, 将 path 添加到路径集 paths 中. 继续下一次循环;

(3) 寻找 G_2 中形如 $x \rightarrow y$ 的边. 若不存在, 寻找路径集已经完成, 结束循环;

(4) 添加 $x \rightarrow y$ 到 path 中, 设置 y 节点为当前节点, 返回(2);

end

算法2中对增广路径的处理方法, 可参见文献[10]. 在算法2中, 我们可以选择最大数目的节点不相交路径, 也可以确定一个常量 k , 选择 k 条节点不相交路径. 只需在算法2第步中对 path 路径集的数目作限制即可. 为此我们根据算法1和2可以获取最大数目的节点不相交路径.

4.3 算法正确性证明

定理2 根据算法2第步中的算法可以将流图转换成路径集.

证明 假设从源节点每次流出来的流为1, 经过 x (除了源节点) 流过的流只有1, 从而使 x 流向相邻节点只有一个 (设节点为 y). 把 $x \rightarrow y$ 添加到相应的路径中, 因而这条路径

每次至少增加一个节点, 因此使得从源节点流出来的流能够到达目的节点, 能够找出一条从源节点到目的节点的路径. 而且, 每进行一次循环, 将减少一个与源节点相邻的节点, 所以算法最终能够结束. 因而能够把流图转化成路径集. 证毕.

4.4 算法复杂度分析

定理3 算法1中拆分算法, 其时间复杂度 $O(|V| + |E|) = O(|V|^2)$. 其中 $|V|$ 为 G 节点个数, $|E|$ 为 G 边个数.

证明 在算法1的第步中, 拆分一个节点的复杂度为 $O(1)$, 总共 $|V| - 2$ 节点, 所以复杂度为 $O(|V|)$. 第步中, 拆分后的节点构造边, 时间复杂度为 $O(|V|)$. 第步中, 把原来 G 中的边集转化成 G_1 中的边集, 总共复杂度为 $O(|E|)$.

综上所述, 算法1的时间复杂度为 $O(|V|) + O(|V|) + O(|E|) = O(|V| + |E|) = O(|V|^2)$. 证毕.

定理4 基于最大流原理的 K 条路径选择算法 (算法2) 时间复杂度为 $O(|V|^2)$, $|V|$ 为 G_1 节点集的顶点个数.

证明 在算法2的第步中, 根据定理3, 该步时间复杂度为 $O(|V| + |E|)$. 第步中, 一次进行最短路径寻找所使用 Dijkstra 算法复杂度为 $O(|V|^2)$, 增广路径处理时间复杂度为 $O(|E|)$, 总共进行 K 次寻路, 所以总共的时间复杂度为 $O(K * (O(|V|^2) + |E|)) = O(|V|^2)$. 第步中, (3) (4) 进行边构造的时间复杂度为 $O(|V|)$, 总共进行 K 条路径构造, 时间复杂度为 $O(K * |V|) = O(|V|)$.

综上所述, 该算法的时间复杂度为 $O(|V| + |E|) + O(|V|^2) + O(|V|) = O(|V|^2)$. 证毕.

从定理3,4可以看出, 基于最大原理的多路径选择算法可以在多项式时间复杂度里面完成.

5 模拟结果和性能分析

5.1 模拟环境和性能评估标准

我们采用了 GoMoSim^[11] 模拟器来测试 AMR 的性能. 在模拟测试中, 采用的信道带宽为 2Mbps, 节点传输距离约 250m, MAC 层协议为 IEEE 802.11.50 个节点在 800m × 700m 的矩形区域里随机移动, 移动符合 WayPoint 随机模型. 该模型有两个重要参数: 移动速度和停留时间. 模拟测试中, 我们设置的移动速度为 5 m/s - 10 m/s 之间, 停留时间为 30 S - 300 S 之间, 每隔 30 S 模拟一次. 我们以停留时间作为节点移动特性的强弱标准, 并以此评价 AMR 和 DSR 的性能. 每次模拟的时间均为 600 秒. 模拟包含 40 条 CBR 流的传输, 每条流每隔 2 S 发送一个大小为 1K 字节的数据包.

参照文献^[12]的要求, 我们将通过数据传输率、端到端延迟、控制包开销以及丢包数等参数来评价 DSR 和 AMR 的性能.

5.2 模拟结果及性能分析

图4反映出 AMR 算法所发的控制包比 DSR 少 50%, 其中控制包包括请求包、响应包和错误包. 这是由于 DSR 在每次链路断开时, 都要重新请求获取链路, 因此控制包也就相应的增加. AMR 算法由于使用替换路径, 可以减少请求包和错误包, 而且由于我们限制了响应的次数, 也减少了响应包数目. 在停

留时间为 120 秒情况下,控制包降到最低点.随着停留时间的增加,控制包又逐渐上升.这是由于网络拥塞的原因而导致控制包的增加.

从图 5 可以看出,AMR 算法数据包的传输率保持在 90%左右,比 DSR 好 20% - 60%左右.当移动速度比较高时,DSR 传输率只有 53%左右,而 AMR 算法能够达到 90%左右.这主要是由于移动速度较大时,DSR 链路经常断,造成经常重新寻路,又由于缓冲的存在导致路由信息无法获得最新的信息,从而使得泛洪请求时的路由信息不佳.而 AMR 算法在断路时,可以找到一条替换路径,无需重新寻路,只有在所有路径都断掉情况下才再次做泛洪路由请求.而且,由于没有使用缓冲,所以找到的路由信息比较有效.当移动速度较低(停留时间较大)时,DSR 算法和 AMR 算法传输率都不断地上升,原因是在网络趋于稳定时,链路断开的概率比较小.

图 6 显示,在同样的环境中,AMR 算法的丢包数比 DSR 算法少 1000 - 3000 个包,而且一直保持在 600 - 1400 个包,这样一个比较低的数值.在移动速度较高时,DSR 因链路断开而重新泛洪请求时,获取的信息大部分来自邻接点的缓冲,无法获得最新的网络信息,导致丢包数目增加.而 AMR 算法在链路断掉时,可找到替换路径,重新把包发送出去.而且泛洪请求时可获取最新的网络信息,所以丢包数目也就相应较低.在移动速度较低时,两者的丢包率都是下降.这是由于网络拓扑变化较小,趋于稳定.

由图 7 我们可以看出,在端到端平均延迟方面 AMR 算法比 DSR 算法降低 40% - 60%,而且延迟一般保持在 0.35 - 0.45 s 比较低的水平.端到端平均延迟主要原因是排队缓冲

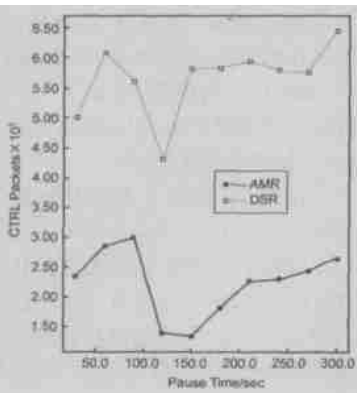


图 4 DSR 和 AMR 控制包的比较

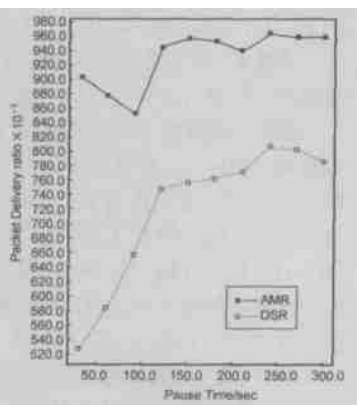


图 5 DSR 和 AMR 数据传输率比较

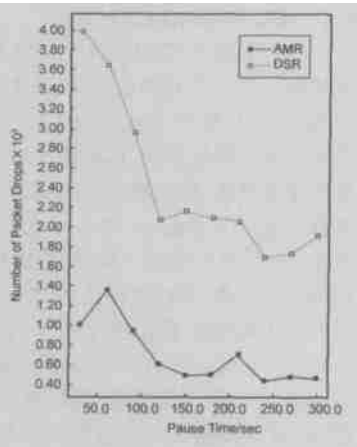


图 6 DSR 与 AMR 丢包数比较

延迟和传播延迟等. AMR 算法把数据包分布在不同路径上运输,减少了排队缓冲延迟.有趣的是随着停留时间增加(移动速度下降),端到端的平均延迟也随之增加.这是由于网络拥塞导致缓冲延迟增加,进而导致端到端延迟的增加.

AMR 的网络传输开销主要包括控制包开销和数据包传输,从以上分析我们可以看出,虽然由于每次的路由发现的所带来的开销较 DSR 有所增加,但路由发现次数减少,进而从控制包开销的比较看,AMR 比 DSR 控制包开销减少 50%左右,而且在数据传输率方面,AMR 比 DSR 提高 20 - 60%左右,所以 AMR 比 DSR 在网络传输开销方面有较大的优势.

在与其他的多路径协议比较方面,由于大部分多路径算法未形成规范,我们做了间接的比较:与 DSR 相比,SMR^[5]在控制包方面降低约 25%,数据传输率提高 5% - 25%,端到端延迟降低 5% - 50%,空间存储开销增加 2 倍(或更多)左右;Wu 的算法^[6]与 DSR 相比,控制包降低 5% - 30%,端到端延迟降低 10% - 40%,空间存储开销增加 2 - 3 倍,在路由响应阶段路径重定向时间复杂度 $O(|V|^2)$. [7] 相对于 DSR 在控制包降低 10% - 30%,端到端延迟增加 10%左右,空间存储开销增加 3 - 7 倍. AMR 在控制包比 DSR 减少 50%左右,数据传输率提高 20% - 60%,端到端延迟降低 40% - 60%,较低的丢包率,空间存储开销增加 3 - 4 倍,路径选择算法时间复杂度为 $O(|V|^2)$. 从以上比较,我们可以看出 AMR 在算法时间复杂度,网络传输开销(主要指控制包开销和数据传输率),存储开销,端到端延迟和可靠性(主要指丢包率)等方面都有较大的提高.

6 结论和未来工作

由于无法精确获得整个网络的拓扑结构,获取多条节点不相交路径仍然是 Ad Hoc 移动网络多路径路由中亟待解决的问题. 本文提出了基于网络最大流的多路径算法 AMR,能够比较有效地找到多条节点不相交路径. 通过模拟测试显示,该算法在控制包、发送率、丢包率和端到端平均延迟等方面具有较好的性能. 进一步的研究中,我们将建立一个基于概率的缓冲模型,更有效地利用缓冲机制,提高网络的可靠性. 通过控制方法预测路径的存在以减少请求次数. 基于多路径协议的网络拥塞控制模型、基于多路径分发密钥以提高网络安全和基于多路径的组播通信也是值得研究的问题.

参考文献:

- [1] D Johnson, D A Maltz. The dynamic source routing protocol for mobile

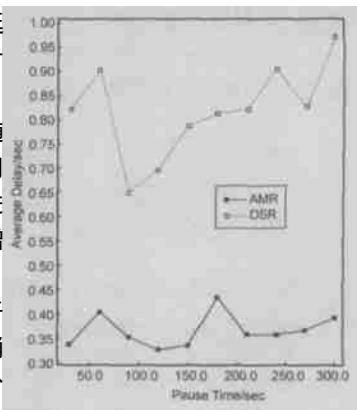


图 7 DSR 与 AMR 端到端平均延迟比较

- ad hoc wireless networks [EB/OL]. <http://www.ietf.org/internet-drafts/draft-ietf-manet-dsr-09.txt>, IETF mobile Ad Hoc Networks Working Group, Internet Draft, work in progress, 2003, 04 - 15.
- [2] C E Perkins, E M Royer. Ad-hoc on demand distance vector routing [A]. IEEE Workshop on Mobile Computing Systems and Applications [C]. Washington : IEEE Computer Society, Feb 1999.
- [3] Vincent D Park, M Scott Corson. A highly adaptive distributed routing algorithm for mobile wireless networks [A]. Proceedings of INFOCOM '97 [C]. New York : ACM Press, 1997. 1405 - 1413.
- [4] Charles E Perkins, Pravin Bhagwat. Highly dynamic Destination - Sequenced Distance - Vector routing (DSDV) for mobile computers [A]. Proceedings of the SIGCOMM '94 Conference on communications Architectures, Protocols and Applications [C]. New York : ACM Press, 1994. 234 - 244.
- [5] S J Lee, M Gerla. Split multipath routing with maximally disjoint paths in Ad hoc networks [A]. Proceed of IEEE ICC2001 [C]. Washington : IEEE Compute Society, 2001. 3201 - 3205.
- [6] Kai Wu, Janelle Harms. Performance study of a multipath routing method for wireless mobile Ad hoc Networks [A]. Cincinnati, Ohio Aug : Proceedings of IEEE/ ACM 9th International Symposium on Modeling, Analysis and simulation (MASCOTS 01) [C]. New York : ACM Press, 2001. 99 - 107.
- [7] A Nasipuri, R Castaneda, S DAS. Performance of multipath routing for on- demand protocols in mobile Ad Hoc networks [J]. ACM/ Kluwer Mobile Networks and Applications (MONET) Journal, 2001, 6(4) : 339 - 349.
- [8] P Papadimitratos, ZJ Hass, E G Sirer. Path set selection in mobile Ad hoc networks [A]. Lausanne, Switzerland : ACM Mobic 2002 [C]. New York : ACM Press, 2002.
- [9] J Broch, D A Maltz, D B Johnson, Y Hu, J Jetcheva. A performance comparison of multi - hop wireless Ad Hoc network routing protocols [A]. ACM/ IEEE Int. Conf. On Mobile computing and network [C]. New York : ACM Press, 1998. 85 - 97.
- [10] Thomas H Cormen, Charles E Leiserson Ronald L Rivest, Clifford Stein. Introduction To Algorithms (second edition) [M]. Massachusetts : MIT Press, 2001.
- [11] UCLA Parallel Computing Laboratory and Wireless Adaptive Mobility Laboratory. GoMoSim: A Scalable Simulation Environment for Wireless and Wired Network Systems [EB/OL]. <http://pcl.cs.ucla.edu/projects/glomsim.htm>.
- [12] S Corson, J Macker. Mobile Ad Hoc networking (MANET) : Routing Protocol performance issues and evaluation considerations [EB/OL]. <http://www.ietf.org/rfc/rfc2501.txt>, RFC2501, 1999 - 01.

作者简介:



陈跃泉 男, 1979 年 10 月出生于福建省龙海市, 硕士研究生, 主要研究领域为 Ad Hoc 移动网络路由, 网络安全.

郭晓峰 男, 1977 年 12 月出生于福建省龙海市, 硕士生, 主要研究领域为 Ad Hoc 移动网络路由, 对等网络, 拥塞控制.

曾庆凯 男, 1963 年 1 月出生于安徽省来安县, 教授, 主要研究领域为分布计算, 信息安全等.

陈贵海 男, 1963 年 3 月出生于江苏省盐城市, 教授, 博导, 主要研究领域为并行计算, 网络系统, 组合数学.