

移动自组网选播路由协议 ARAD^①

彭革刚^{②**} 李新宇^{**} 宋 鹰^{**} 向黎生^{**} 沈 清^{**} 李仁发^{*}

(^{*} 湖南大学计算机与通信学院 长沙 410082)

(^{**} 拓维信息系统股份有限公司博士后科研工作站 长沙 410205)

摘 要 针对移动自组(ad hoc)网的特点,提出了一种适合 ad hoc 移动网络的选播路由协议——ARAD 协议。该协议在充分考虑选播路由特点的基础上,将 ad hoc 按需距离矢量路由(AODV)协议的路由发现机制与动态源路由(DSR)协议的缓存源路由技术相结合,在动态变化的 ad hoc 网络中实现了一种高效的选播路由。模拟结果表明,在动态的网络环境下,ARAD 协议能够取得良好的性能。

关键词 移动自组(ad hoc)网,选播,单播,路由协议

0 引 言

移动自组(ad hoc)网是一种没有基础设施支持的移动网络,具有无中心、自组织、可快速部署、可移动和多跳等特点^[1-3],特别适合应用于某些需要临时快速建立一个通信网络的场合。由于自组网中主机的移动将带来不可预测的网络拓扑结构变化,因而路由的寻找和维持变得更加困难。针对这个问题,研究人员提出了许多 ad hoc 路由协议^[4-6],但是这些路由协议主要用于解决 ad hoc 网络中的单播(unicast)和组播(multicast)路由问题,而没有涉及到选播(anycast)路由。选播协议是一种网络服务,它能够让一组提供相同服务的服务器使用单一的选播地址,服务访问者使用该选播地址就可以访问到满足某种路由策略的该选播组中的某一台服务器。随着越来越多的应用需要选播服务,在 IPv6 中,选播服务已经被定义为一种标准服务模型^[7-9]。在有线网中,提供选播服务能够提高网络的性能,增强网络的恢复能力。在无线 ad hoc 网络中,选播服务是实现分布式服务的一种有效方法,它简化了分布式服务的配置和管理过程。同时在拓扑结构高度动态变化的 ad hoc 网络中使用选播服务,能够进一步提高 ad hoc 网络的抗毁性。本文针对移动自组网的特点和应用需求,提出了一种适合 ad hoc 移动网络

的选播路由协议——基于 AODV(ad hoc 按需距离矢量路由)和 DSR(动态源路由)的选播路由(anycast routing)协议,简称 ARAD 协议。该协议在充分考虑选播路由特点的基础上,将 AODV 协议的路由发现机制与 DSR 协议的缓存源路由技术相结合,在动态变化的 ad hoc 网络中实现了一种高效的选播路由,取得良好效果。

1 网络模型

用网络图 $G = \langle V, E \rangle$ 表示一个 ad hoc 网络,其中 $V = \{v_1, v_2, \dots, v_n\}$ 是 ad hoc 网络 G 中的移动节点集合, $E = \{e_1, e_2, \dots, e_m\}$ 是 ad hoc 网络 G 的链路集合。由于 ad hoc 网络拓扑的动态性,链路集合 E 将随着时间而变化。每个移动节点 $i(i \in V)$ 根据无线信号传播模型具有大小为 R_i 的传播半径,信道传输率为 B_i 。如果某节点 $j(j \in V)$ 在节点 i 的无线传播半径 R_i 内,则表示节点 i 到节点 j 之间存在一条有向链路 $e_{ij}(e_{ij} \in E)$ 。选播地址 A 是一个目的节点的集合,并且 $A \subset V$ 。对于选播地址 A ,采用 $Gr(A)$ 表示一组目的主机: $Gr(A) = \{D_1, D_2, \dots, D_w\}$, 其中 $D_1, D_2, \dots, D_w \in V$ 。无线 ad hoc 网络中的选播传输路径是指在 ad hoc 网络 G 中从源节点至 $Gr(A)$ 中任何一个节点符合要求的通路。ad hoc 网络的选播路由主要研究如何在网络 G 中找到并

① 国家科技支撑计划(2007BAH14B00)和国家科技重大专项(2012ZX01045-004-005)资助项目。

② 男,1975 年生,博士,博士后;研究方向:网络通信与信息工程;联系人,E-mail: penggegang@hotmail.com
(收稿日期:2012-03-26)

维护这样一条选播传输路径。

2 ARAD 协议描述

2.1 ARAD 协议的基本思想

Ad hoc 按需距离矢量路由 (Ad hoc On-demand Distance Vector routing, AODV) 协议是一个建立在目的序列距离矢量路由 (DSDV) 上的按需路由协议。AODV 协议的特点在于它采用逐跳转发分组方式, 同时加入了组播路由协议扩展。当源节点需要与某目的节点进行通信时, AODV 协议将首先在本节点所维护的路由表中, 查找是否有到该目的节点的路由, 如果路由表已经包含到该目的节点的路由, 则它立刻使用这条路由发送数据分组, 否则, 它将发起一个路由发现过程, 来找到一条从它到该目的节点的可用路由。但 AODV 协议的这种路由表查找方式是面向目的节点的, 这意味着这种查找只能仅仅查找到通往特定目的节点的路由, 而不能查找到通往该目的节点所在的活动路径 (active route) 上的其它节点的路由。这样将导致当该源节点需要同该活动路径上的其它中间节点进行通信时, 由于 AODV 协议是面向目的节点来查找路由表, 因此尽管该活动路径中包含到这些节点的路由信息, 但它仍然不能找到通往这些节点的路由, 最后不得不发起一个新的路由发现过程来发现路由。

图 1 是一个包含 9 个节点的移动自组网拓扑, 图中节点 N_1 到节点 N_5 之间存在一条活动路径 $N_1-N_2-N_3-N_4-N_5$ 。当节点 N_1 需要与目的节点 N_5 进行通信时, 节点 N_1 将首先在路由表中查找到目的节点为 N_5 的路由, 由于节点 N_5 是该活动路径上的端节点, 因此节点 N_1 能够迅速找到这条到目的节点 N_5 的路由 $N_1-N_2-N_3-N_4-N_5$, 然后使用这条路由进行数据传输, 而无需重启一个新的路由发现过程来查找所需路由。同样, 当节点 N_2 、 N_3 和 N_4 需要同节点 N_5 进行通信时, 它们也可通过查找各自的路由表来找到通往目的节点 N_5 的路由。但当节点 N_1 还需同这条活动路径上的中间节点 (N_2 、 N_3 、 N_4) 通信时, 由于它的路由表中该活动路径是面向目的节点 N_5 的, 因此节点 N_1 并不能找到通往该中间节点的路由, 尽管这些路由信息实际上已经被包含在该活动路径中。这将导致 N_1 启动一个新的路由发现过程来发现到这些中间节点的路由。这对于活动路径上的其它节点 N_2 、 N_3 、 N_4 也同样存在这种情况。

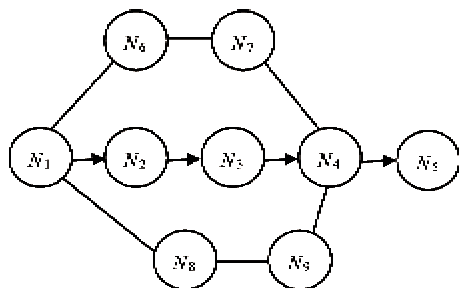


图 1 AODV 路由查找示例

从以上分析可看到, AODV 协议面向目的节点查找路由表的方式在一定程度上降低了源节点发现路由的能力, 引起了额外的路由发现过程, 而过多的路由发现过程的启动, 会带来更多的路由开销, 从而降低路由协议性能。对于这个问题, 可使用类似动态源路由 (DSR) 协议的缓存源路由技术来解决它。按 DSR 协议, 每个节点的路由表中存放的是到每个目的节点的源路由, 因此, 当源节点需要同某目的节点通信时, 它不仅能够查找到通往该目的节点的路由, 而且也能够很方便地查找到到该目的节点对应的活动路径上的各中间节点的路由, 当源节点有到这些中间节点的通信需求时, 不必与 AODV 协议一样启动一个路由发现过程来发现新路由, 从而可避免 AODV 中由于面向目的节点的查找路由而带来的不必要的路由发现过程, 提高协议的性能。

基于这种思想, 我们提出了基于 AODV 和 DSR 的按需路由协议 ARAD, 此协议不必周期性的交换路由信息。同 AODV 的一样, ARAD 协议使用序列号来保证路由的更新, 每个节点需要维护两个序列号: 单播 (unicast) 和选播 (anycast) 序列号, 节点为每个选播组维护一个选播序列号,

选播组节点通过周期性的发布选播组 Hello 分组来维护每个组的选播序列号, 协议使用选播序列号来进行选播通信。同时 ARAD 使用 DSR 的缓存源路由技术来进一步提高协议性能。中间节点 (包括源节点) 在转发路由应答分组时, 需要缓存从目的节点到本节点的路由。这样, 节点将获得更多的路由信息, 在处理随后的路由请求分组时, 如果节点能够在已缓存的源路由信息中发现相应的目的节点, 节点可立刻使用该路由来结束路由请求分组的进一步广播, 以减少路由协议开销。ARAD 协议主要分为路由发现和路由维护两部分。路由发现负责寻找一条 anycast 路由, 路由维护负责对找到的路由进行检查, 确定路由是否因节点移动等原因失效。一旦失效, 负责通知相应的源节点删除失效路由。

2.2 ARAD 路由发现

当源节点希望进行选播通信时,如果发现没有去往该选播地址的路由,源节点则向网络广播一个 ANYRREQ (ANYcast Route REQuest) 的分组。我们将这个请求分组记为 ANYRREQ (Source _ Addr, Anycast _ Addr, ANYRREQ _ ID, Source _ Seq, Anycast _ Seq, Hop _ Cnt), 其中 Source _ Addr 和 Anycast _ Addr 分别表示源节点的地址和选播路由地址, ANYRREQ _ ID 为 ANYRREQ 分组的顺序号, 每次源节点产生新的 ANYRREQ 时将增加分组的顺序号, 该顺序号和 Source _ Addr 一起在协议中唯一标识 ANYRREQ 分组。Source _ Seq 和 Anycast _ Seq 则分别表示源节点序列号和对应选播组的序列号, Hop _ Cnt 记录源节点到选播目的节点的跳数。

在广播 ANYRREQ 分组后, 源节点将等待选播目的节点的应答分组, 如果在 ROUTE _ SETUP _ TIME 时间内没有收到任何 ANYRREQ 分组的回答, 源节点将重发 ANYRREQ 请求分组, 直到收到回答或请求超过给定的次数, 若源节点在给定数次后仍没有收到回答, 则认为目前网络中没有到选播组的路由, 将丢弃要发给选播目的主机的数据。

中间节点收到 ANYRREQ 分组后, 先检查是否该分组以前已收到过, 对于新的 ANYRREQ 分组, 中间节点按条件产生或更新相应的反向路径, 以后应答分组将沿着这些反向路径到达源节点, 如果该中间节点就是目的节点, 或者它有一条到指定选播目的节点的路由, 且该路由所对应的选播序列号不小于 ANYRREQ 分组中相应的选播序列号, 那么该节点将产生 ANYRREP 应答分组, 否则中间节点重广播该 ANYRREQ 分组。中间节点处理 ANYRREQ 分组的具体过程见算法 1。算法描述中用 P 和 RT 分别表示控制分组和节点所维护的路由表, 控制分组和路由表中的域值用对象式的方法表示, 如 P. Source _ Addr 表示控制分组的 Source _ Addr 域的值。

算法 1: 中间节点 N 处理 ANYRREQ 分组算法

```

if (IsRepetitionPkt (P. Source _ Addr, P. ANYRREQ _ ID))
{
    Discard(P);
    Exit;
}
if (ReversePathNotExist (P. Source _ Addr))
    CreateReversePath (P. Source _ Addr, P. Source _

```

```
Seq, P. Hop _ Cnt)
```

```

Else if ((P. Source _ Seq > RT. Source _ Seq) ∨ ((P. Source _ Seq == RT. Source _ Seq) ∧ (P. Hop _ Cnt < RT. Hop _ Cnt)))

```

```

    UpdateReversePath (P. Source _ Addr, P. Source _ Seq, P. Hop _ Cnt);

```

```

    if ((P. Anycast _ Addr = N) ∨ (ExitPathToAnycastDest (P. Anycast _ Addr) ∧ (RT. Anycast _ Seq ≥ P. Anycast _ Seq)))
    {

```

```

        CreateANYRREP();

```

```

        Exit;
    }

```

```

Else
{

```

```

    P. Hop _ Cnt = P. Hop _ Cnt + 1;

```

```

    ReBroadcast(P);
}

```

应答节点在进行应答时, 首先生成一个应答分组 ANYRREP (Source _ Addr, Anycast _ Addr, Anycast _ Seq, Hop _ Cnt, Node _ list, lifetime), 其中 Node _ list 用来记录 ANYRREP 分组所途经节点的信息, 以便中间节点在转发 ANYRREP 的过程中来缓存从目的节点到本节点的路由, 然后应答节点将沿反向路径方向向源节点 Unicast 这个 ANYRREP 分组。反向路径上的中间节点收到 ANYRREP 分组后, 按条件产生或更新相应的转发路径。中间节点只有在 ANYRREP 分组中的选播序列号大于本节点已记录的相应选播序列号, 或者等于已记录的相应选播序列号但具有更小的跳数时, 才更新相应的路由信息。在转发 ANYRREP 分组之前, 中间节点根据 ANYRREP 分组中的 Node _ list 信息来缓存从目的节点到本节点的路由, 以便节点能够获得更多的路由信息。同时将本节点加入 Node _ list 并修改 Hop _ Cnt 值。最后将 ANYRREP 分组转发到反向路径上的下一跳节点。中间节点处理 ANYRREP 分组的具体过程见算法 2。一旦源节点收到这个 ANYRREP 后, 它就成功地建立了一条选播传输路径, 可以立刻使用该路径来发送数据。当随后的 ANYRREP 达到时, 如果 ANYRREP 分组中的选播序列号大于本节点已记录的相应选播序列号, 或者等于已记录的相应选播序列号但具有更小的跳数, 源节点将更新选播路由信息, 选择更好的选播传输路径。

算法 2: 中间节点 N 处理 ANYRREP 分组算法

```

if(ForwardPathNotExist(P. Source _ Addr))
    CreateForwardPath(P. Anycast _ Addr, P. Anycast _
Seq, P. Hop _ Cnt)
    Else if((P. Anycast _ Seq > RT. Anycast _ Seq) ∨
((P. Anycast _ Seq == RT. Anycast _ Seq) ∧ (P. Hop _
Cnt < RT. Hop _ Cnt)))
        UpdateForwardPath(P. Anycast _ Addr, P. Anycast _ Seq,
P. Hop _ Cnt);
    else
        {
            Discard(P)
            Exit;
        }
CacheRouteInfo(P. Node _ list)
AppendNodelist(P. Node _ list, I)
P. Hop _ Cnt = P. Hop _ Cnt + 1;
Forward(P)
    
```

2.3 ARAD 路由维护

由于移动自组网网络拓扑结构的高度动态性, 节点间的链路由于主机移动等一些原因而经常断开, 因此有效的路由维护变得十分重要。ARAD 协议使用 Anycast _ Hello 分组实现局部链路管理和检测链路状态。对于选播传输路径上的链路 $(ni, nj) \in E$, 如果节点 ni 在 ANYCAST _ HELLO _ INTERVAL 时间内没有向下游节点发送任何数据分组, 将向它的邻节点广播一个的 Anycast _ Hello 分组, 为了防止收到该分组的邻节点继续广播, 将 Anycast _ Hello 分组的 TTL 值设置为 1。因此如果在 MAX _ ANYCASTHELLO _ PERIOD 时间内, 节点 ni 仍没有收到 nj 的 Anycast _ Hello 分组, 它将认为链路 (ni, nj) 已经断开, 这时 ni 向选播传输路径上的上游节点发送 AnycastRoute _ Error 分组, 上游节点收到该分组后, 根据分组提供的信息来更新路由表, 然后继续将该分组向上游转发直到源节点。源节点收到 AnycastRoute _ Error 分组后, 如果仍然需要该选播通信, 它可以重新发起路由发现过程。

在路由维护期间, 每个节点需要维护一个选播组信息表来记录相关的选播组信息, 选播组信息表主要包括目的节点、选播组 ID 和有效时间, 其中有效时间主要用来维护选播组信息表中的记录产生和更新的时间。每个节点需要负责处理各种选播消息并更新本地选播组信息表, 这些选播消息主要包括选播创建、选播加入以及选播删除。当一个节点收到一条创建新选播组的消息时, 它向本地选播组信

息表中插入一条新的记录, 同时广播该信息以向网络告知新的选播组的存在。同样当一个节点收到一条加入选播组的消息时, 它向本地选播组信息表中插入一条新的记录, 同时负责广播该消息以向网络告知信息更新。而当一个节点收到离开一个选播组的消息时, 它负责删除本地选播组信息表中的相应项, 并广播更新信息。

3 协议的正确性证明

定理 ARAD 协议能够在移动自组网中确保不出现选播路由环路。

证明: 假设移动自组网中存在一条到选播地址 A 的选播路由环路, 将这个选播路由环路上的节点序列记 $N_i^A, N_i^A \mapsto N_{i+1}^A$ 表示在节点 N_i^A 的路由表中, 对应于选播地址 A 的路由项以节点 N_{i+1}^A 为下一跳节点。因此有 $N_i^A \mapsto N_{i+1}^A, i = 1, 2, \dots, n$ 和 $N_n^A \mapsto N_1^A, AS_i^A$ 表示节点 N_i^A 路由表中, 对应于选播地址 A 的选播序列号 Anycast _ Seq, HOP_i^A 表示节点 N_i^A 到选播目的 A 的跳数。根据算法 1, 当 $N_i^A \mapsto N_{i+1}^A$ 时, 有 $AS_i^A \leq AS_{i+1}^A$, 即

$$AS_1^A \leq AS_2^A \leq AS_3^A \leq \dots \leq AS_n^A \leq AS_1^A \quad (1)$$

由式(1), 可知这条选播路由环路上的所有节点选播序列号相等, 进一步根据算法 2, 可知在每个节点 N_i^A 中的下一跳信息必定是来自于同一个 ANYRREQ 分组, 因此当 $N_i^A \mapsto N_{i+1}^A$ 时, 有 $HOP_i^A = HOP_{i+1}^A + 1$, 即

$$HOP_1^A = HOP_2^A + 1 = HOP_3^A + 2 = \dots = HOP_n^A + (n - 1) \quad (2)$$

由 $N_n^A \mapsto N_1^A$, 有

$$HOP_n^A = HOP_1^A + 1 \quad (3)$$

由式(2)、式(3)可得 $n = 0$, 这显然不成立, 因此并不存在一条到选播地址 A 的选播路由环路, 证毕。

4 模拟和性能分析

4.1 模拟环境

本节我们使用 GloMoSim 网络模拟器来评价和分析 ARAD 的性能。在模拟中采用自由空间模型, 50 个节点随机地分布在 $1000\text{m} \times 1000\text{m}$ 的矩形区域内, 每个节点的无线传输半径为 250m, 信道容量为 2Mbps, 使用 CBR 作为数据源。选择随机路标 (waypoint) 模型作为节点的移动模型, 节点的每次

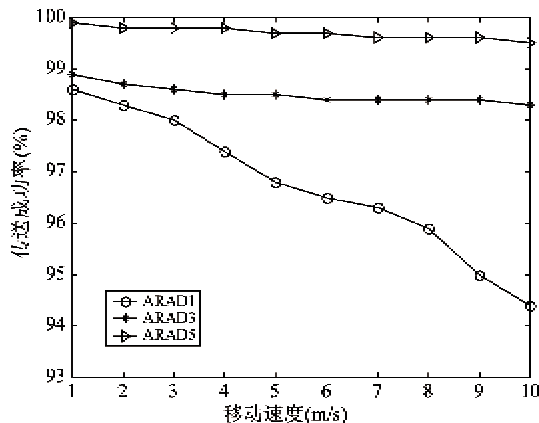
停顿时间为 3s,最小移动速度为 0,最大移动速度为在 1m/s 到 10m/s 之间变化。每次模拟试验执行 600s。所有模拟试验结果均为多次试验的平均值。

4.2 性能评价参数

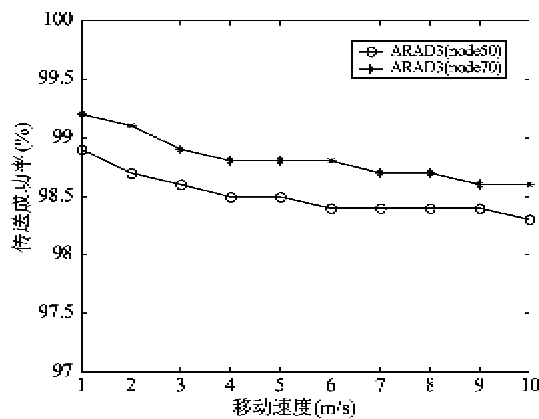
性能评价参数主要包括:(1)分组传送成功率(SDR)=接收端接收的分组数/发送端发送的分组数;(2)路由协议开销(OH)=协议发送的控制分组个数;(3)路由的最优性(PO)=协议实际使用的平均跳数与最优的平均跳数(即使用最短路径的跳数)之差。

4.3 模拟结果与性能分析

图 2(a)给出了 ARAD 协议在不同的节点移动速度下,在传送成功率方面的性能比较。图中曲线 ARAD1、ARAD3 以及 ARAD5 分别表示选播组成员数目为 1、3 和 5 时 ARAD 协议的性能,其中 ARAD1 可看做单播(unicast)。从图中可以看到,随着节点的移动速度增加,数据传输过程中链路断开的概率加大,从而导致协议的传送成功率有所下降,同时可以看到 ARAD5 和 ARAD3 的性能要优于 ARAD1,



(a)



(b)

图 2 分组传送成功率比较

ARAD5 和 ARAD3 的传送成功率要高于 ARAD1,并且随着节点的移动速度增加,传送成功率并没有明显下降。这主要是因为随着选播组大小的增加,源节点能够有更多的可选路由,可以在多条传输路径上实现通信负载的平衡,使得即使在动态的网络环境下,协议也能够找到一条传输路径传输数据分组,从而提高了分组的传送成功率。图 2(b)是 ARAD3 在不同的节点数情况下的传送成功率,在图中 70 个节点时的 ARAD3 传送成功率要高于 50 个节点时的 ARAD3,这是由于随着网络中节点数的增加,各节点之间的可用链路也随之增多,使从源节点到目的节点之间建立有效链路的可能性增大,传送成功概率也相应增加。

图 3 给出了 ARAD 协议在不同的节点移动速度下,路由协议开销的性能比较。可以看到,当节点的移动速度增加时,链路发生破裂的概率也随之增加,使得协议的路由重构次数增加,从而导致路由协议的协议开销不断增加。同时可以看到 ARAD5 和 ARAD3 的性能要优于 ARAD1,ARAD5 和 ARAD3 的路由协议开销要低于 ARAD1。这主要是由于随着选播组大小的增加,ARAD 协议能获得更好负载平衡效果,使得网络中的大部分路由信息是更新的,这样源节点能够使用更准确的路由信息来选择传输路径,这样的传输路径具有更高的可靠性,在传输过程中能够减少链路发生破裂的概率,从而减少了协议的路由重构次数,使得路由协议开销减少。

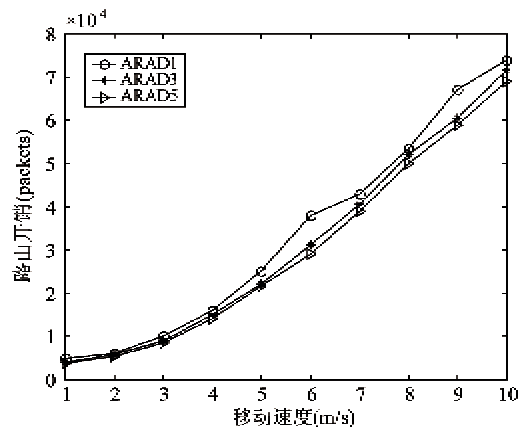


图 3 路由协议开销的比较

图 4 是 ARAD 协议在不同的节点移动速度下,路由最优性的性能比较。从该图中,可以看到在路由最优性方面,ARAD5 和 ARAD3 的性能也要优于 ARAD1,这是由于在选播组大小增加的情况下,源节点在 ROUTE_SETUP_TIME 内能够收到更多的

ANYRREP 分组,它可以在多个的可选路由中选择最短的路由来完成数据分组的发送。因此相对于 ARAD1, ARAD5 和 ARAD3 能够获得更短的传输路径,从而提高了协议的路由最优性。

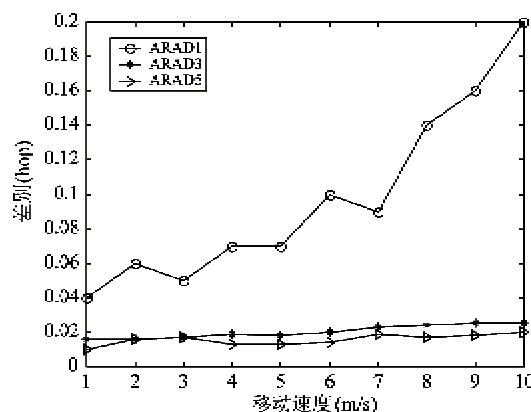


图 4 路由最优性的比较

5 结 论

针对移动自组网的特点,本文提出了一种适合 ad hoc 移动网络的选播路由协议 ARAD,该协议在充分考虑选播路由特点的基础上,将 AODV 协议的路由发现机制与 DSR 协议的缓存源路由技术相结合,在动态变化的 ad hoc 网络中实现了一种高效的选播路由。模拟结果表明,在动态的网络环境下,ARAD 协议能够取得良好的性能,并且协议性能比较稳定。

参考文献

- [1] Poongkuzhali T, Bharathi V, Vijayakumar P. An optimized power reactive routing based on AODV protocol for Mobile Ad-hoc network. In: Proceedings of the International Conference on Recent Trends in Information Technology, Chennai, India, 2011. 194-199
- [2] Zhang Z K, Wang Y. An Iterative Multi-path Routing Protocol in Wireless Ad Hoc Networks. In: Proceedings of the IEEE 10th International Conference on Trust, Security and Privacy in Computing and Communications, Changsha, China, 2011, 1630-1637
- [3] Jasani H. Quality of Service Evaluations of On Demand Mobile Ad-Hoc Routing Protocols. In: Proceedings of the 5th International Conference on Next Generation Mobile Applications, Services and Technologies, Cardiff, England, 2011. 123-128
- [4] Putta C S R, Prasad K B, Ravilla D. Performance of ad hoc network routing protocols in IEEE 802.11. In: Proceedings of the International Conference on Computer and Communication Technology, Allahabad, India, 2010. 371-376
- [5] Kuppusamy P, Thirunavukkarasu K, Kalaavathi B. A study and comparison of OLSR, AODV and TORA routing protocols in ad hoc networks. In: Proceedings of the 3rd International Conference on Electronics Computer Technology (ICECT), Kanyakumari, India, 2011. 143-147
- [6] SreeRangaRaju M N, Mungara J. Performance evaluation of ZRP in adhoc mobile wireless network using Qualnet simulator. *IEEE International Symposium on Signal Processing and Information Technology*, 2010, 26(2): 457-466
- [7] Li T S, Zhao Z G. An anycast routing algorithm based on simplified particle swarm optimization and diversity strategy. In: Proceedings of the 6th International Conference on Computer Science & Education, Singapore, 2011. 1075-1079
- [8] Shaikh A, Buysse J, Jaumard B. Anycast Routing for Survivable Optical Grids: Scalable Solution Methods and the Impact of Relocation. *IEEE/OSA Journal of Optical Communications and Networking*, 2011, 3(9): 767-779
- [9] Tan X G. The QoS Routing for Anycast Message. In: Proceedings of the International Conference on Intelligence Science and Information Engineering, Wuhan, China, 2011. 41-44

ARAD: an anycast routing protocol for mobile ad hoc networks

Peng Gegang^{**}, Li Xinyu^{**}, Song Ying^{**}, Xiang Lisheng^{**}, Shen Qing^{**}, Li Renfa^{*}

(^{*} School of Computer and Communication, Hunan University, Changsha 410082)

(^{**} Talkweb Information System CO. LTD, post-doctoral scientific research workstation, Changsha 410205)

Abstract

According to the features of ad hoc networks, this paper proposes the ARAD protocol, a new anycast routing protocol for mobile ad hoc networks. This protocol combines the route discovery mechanism of the Ad hoc On-demand Distance Vector routing (AODV) protocol with the cache technique for source routing of the dynamic source routing (DSR) protocol based on the full consideration of the characteristics of anycase routing, and achieves an effective anycast routing in dynamic ad hoc networks. The simulation results show that the protocol can get good performance in dynamic network environments.

Key words: ad hoc mobile network, anycast, unicast, routing protocol