

(12) 发明专利

(10) 授权公告号 CN 102036337 B

(45) 授权公告日 2013.02.27

(21) 申请号 201010588334.7

(22) 申请日 2010.12.15

(73) 专利权人 山东大学

地址 250100 山东省济南市历城区山大南路 27 号

(72) 发明人 曹叶文 王宁宁

(74) 专利代理机构 济南圣达知识产权代理有限公司 37221

代理人 张勇

(51) Int. Cl.

H04W 40/14 (2009.01)

H04W 80/00 (2009.01)

(56) 对比文件

CN 101394356 A, 2009.03.25, 说明书第 3 页第 1 行, 第 13 行至第 18 行, 第 6 页第 12 行至最后一行.

CN 101150518 A, 2008.03.26, 全文.

CN 101420364 A, 2009.04.29, 全文.

US 2005135360 A1, 2005.06.23, 全文.

陈晨. 在两跳拓扑环境下对 AODV 协议的改进. 《中国优秀硕士学位论文全文数据库》. 2009, 第 26 页第 4.3.3 第 1 段至第 3 段.

蔡一兵, 李海波, 李忠诚, 谢高岗. 移动自组网基于邻居变化率稳定路径选择方法. 《软件学报》. 2007, 摘要, 第 2 页第 1.1 节第 1 段至第 5 段, 第 4 页第 2 节第 3 段至第 4 段.

审查员 尤媛

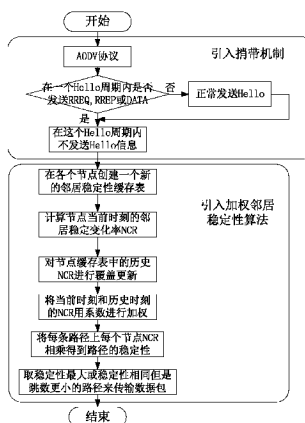
权利要求书 2 页 说明书 7 页 附图 6 页

(54) 发明名称

基于改进 AODV 协议的通信方法

(57) 摘要

本发明公开了一种基于改进的 AODV 协议的通信方法, 该方法首先判断在一个 Hello 周期内节点是否发送了 RREQ, RREP 控制包或 DATA 数据包, 若没有发送则正常传输 Hello 信息, 否则在这个周期内不发送 Hello 信息; 其次求出网络中路径的稳定性, 选择最稳定或稳定性相同但跳数更少的路径来传输数据包。本发明将稍带机制和邻居稳定性算法结合起来进行协议的优化, 融合了稍带机制和加权邻居稳定性算法两者的优势从而使路由开销和端到端时延性能得到很大的提高。



CN 102036337 B

1. 基于改进 AODV 协议的通信方法,其特征是,该通信方法包括两部分:

首先判断在一个 Hello 周期内节点是否发送了 RREQ, RREP 控制包或 DATA 数据包,若没有发送则正常传输 Hello 信息,否则在这个周期内不发送 Hello 信息;

其次求出网络中路径的稳定性,这个过程是在各个节点处要创建一个新的邻居稳定性缓存列表,将当前时刻的邻居稳定变化率存储并覆盖更新有限的缓存列表中最老的邻居稳定性信息,只保留最近的稳定性信息,然后对当前时刻和历史时刻的稳定性由系数 λ 进行加权,计算出节点的加权邻居稳定性进而根据相乘的方法求出网络中路径的稳定性,最后选择最稳定或稳定性相同但跳数更少的路径来传输数据包;

所述加权的邻居稳定性的实现方法,具体步骤如下:

(1) 选择与 Hello 信息发送周期相同的时间间隔 T , 设定邻居集观察周期参数 α , 设定包括当前时刻在内的时刻数为 K , 且在 $0 \sim 1$ 范围内均匀的选取加权系数 λ ;

(2) 对于节点 i , 设定邻居集的观察周期为 $\alpha * T$, 并定义 $[t_r - \alpha * T, t_r]$ 时间段内所有出现过的邻居节点即邻居集为 S_{i,t_r} , 这样相隔时间间隔为 T 的两段时间就可得到

$$NCR_i(t) = \frac{|S_{i,t-T} \cap S_{i,t}|}{|S_{i,t-T} \cup S_{i,t}|}, \text{其中, } S_{i,t-T} \text{ 为 } t-T \text{ 时刻节点 } i \text{ 的邻居集, } S_{i,t} \text{ 为 } t \text{ 时刻节点 } i \text{ 的邻居集, } T \text{ 为发送 Hello 分组的时间间隔};$$

邻居集, T 为发送 Hello 分组的时间间隔;

(3) 利用递推公式可得第 K 个 NCR 值 $NCR_i(t_k, t_{k+1}) = NCR_i(t_1 + (K-1)T, t_2 + (K-1)T)$, 对于这 K 个 NCR 都要缓存在节点新建的链表中, 超过 K 个时刻则进行覆盖更新, 当前时刻对应的 NCR 和这 K 个时刻 NCR 的平均值前面各乘以系数 β 和 $(1-\beta)$, 且满足 $0 \leq \beta \leq 1$; 在 t_{k+1} 时刻节点 i 的 NCR_i 为:

$$NCR_i(t_{k+1}) = \lambda * NCR_i(t_k, t_{k+1}) + (1-\lambda) * \left[\frac{1}{K-1} * \sum_{j=1}^{K-1} NCR_i(t_j, t_{j+1}) \right]$$

其中, $NCR_i(t_{k+1})$ 为当前时刻和历史时刻邻居变化率的加权, 加权系数分别是 λ 和 $1-\lambda$,

$$\lambda = \beta + \frac{1-\beta}{K}, \frac{1}{K} \leq \lambda \leq 1;$$

(4) 将沿每条路径上节点的加权邻居变化率累乘得到每条路径上的稳定性, 选择稳定性数值最大或稳定性相等但是跳数最小的路径来发送数据包。

2. 如权利要求 1 所述的基于改进 AODV 协议的通信方法,其特征是,所述发送 RREQ, RREP 控制包时,如节点需要向特定目的节点发送数据但是路由表中又没有相应的路由信息时就需要进行路由探寻,在分组中增加一个新的字段,这个字段用来存储节点的邻居稳定性度量 NCR_{path} ,目的节点接收到分组之后就相应 RREP,若目的节点离源节点较远,没有在源节点的一跳范围之内,就需要中间节点的转发来实现通信,若中间节点第一次接收到 RREQ 分组,则根据 RREQ 分组 NCR_{path} 字段的值在路由表中记录下来,并建立反向路由同时转发 RREQ,若接收到多个邻居转发的重复 RREQ,则与路由表中邻居稳定性度量相比较,若 RREQ 分组 NCR_{path} 字段数值更大,则更新路由表中邻居稳定性度量值和反向路径,以保证选用最稳定路由,之后继续转发出去,一直到达目的节点;对于到达目的节点的 RREQ,目的节点将 RREQ 中的信息复制到 RREP 沿着反向链路单播传送到源节点,同时建立前向路径,若以后收

到序列号更大更新鲜或序列号相等但是 NCR_{path} 值更大的 RREQ 分组才需要响应 RREP 并更新路由 ;RREQ 和路由表中存储的都是链路沿途节点的邻居变化率的乘积值。

基于改进 AODV 协议的通信方法

技术领域

[0001] 本发明涉及一种基于改进 AODV 协议的通信方法,属于数字通信领域。

背景技术

[0002] 无线自组网按需距离矢量路由协议 AODV (Ad hoc On-Demand Distance Vector Routing, AODV) 作为一种路由选择协议在无线网状网络中有着极其重要的地位,兼具实现单播和组播的功能,是一种典型的按需生成路由的协议。按需的表现就是当源节点向目的节点发送包时,源节点才在网络中发起路由探寻过程,以寻找合适的路由,相较于很多先验式的网络路由协议来说,发送包不是周期性的,这样可以避免很大一部分的冗余开销。在没有路由存在的情况下,源节点才发起路由请求 (Route Request, RREQ),其它的中间节点转发此 RREQ,记录源节点,并建立反向路径,当到达目的节点或直到到达目的节点路由的中间节点时,把路由响应 (Route Reply, RREP) 沿着反向路径回发至源节点,源节点选择跳数最少的路径来传送信息。当链路断裂时,断点的上游节点向源节点发送路由错误 (RouteError, RRER),然后源节点就重新发起路由探寻过程。为了避免节点重复转发 RREQ 以及回路的产生,在 RREQ 中设置了一个序列号,序列号机制也是 AODV 协议的一个重要特征。AODV 协议是在动态源路由协议 (Dynamic Source Routing, DSR) 和目标序列距离路由矢量算法 (Destination Sequenced Distance Vector, DSDV) 协议的基础上发展起来的,其性能得到了很大的提高,但是仍然有其不足之处,如最终选用的最短跳数路径可能并不是最稳定的,因此链路断裂的概率加大;协议发送信息存在冗余路由开销较大等等。

[0003] 对于链路的稳定性研究,在通信过程中一般采用两种方式来选择稳定路径以减少链路断裂情况的发生,一种就是采用一些辅助的通讯设备如 GPS 定位系统,这种方法在思路上还是比较简单的,但是具体实现起来需要 GPS 硬件功能支持,还需要考虑成本等多方面实际问题,这种方法一般很少投入民用通信领域。另一种方式就是采用跨层的思想将通过底层协议获取节点的信号强度信息来预测节点稳定性,这种方式需要多层协议之间进行协作和交叉支持,并且由于底层各种干扰有时难以获取节点的信号强度信息,这就使得这种稳定性检测方法的实际效用大打折扣。之后,逐渐研究出了 NCR-AODV,这种方法的一个最大的突破在于脱离了以上两种传统方式的束缚。这种方法的思想在于节点根据节点邻居变化率来感知节点局部变化程度,最终选择稳定性最好的路由来发送数据。

[0004] 邻居稳定性算法的核心在于邻居变化率的计算,节点 i 的邻居变化率如下所示:

$$[0005] \quad NCR_i(t) = \frac{|S_{i_{t-T}} \cap S_{i_t}|}{|S_{i_{t-T}} \cup S_{i_t}|} \quad (1)$$

[0006] 其中, $S_{i_{t-T}}$ 为 $t-T$ 时刻节点 i 的邻居集, S_{i_t} 为 t 时刻节点 i 的邻居集, T 为发送 Hello 分组的时间间隔。(1) 式定义节点 i 的邻居变化率为相隔 T 的时间点邻居集相同的邻居与这两个时间点所有邻居的比值。比值越大,表明局部拓扑结构越稳固;反之,表明节点局部拓扑变化越剧烈。对于整条路径来说,邻居变化率的路由度量为:

$$[0007] \quad NCR_{path} = \prod_{i \in path} NCR_i(t) \quad (2)$$

[0008] 其中 NCR_{path} 是路径上所有节点邻居变化率的乘积, 最终期望选择此值最大的路径来转发传送 DATA 分组。采用累乘法而不是累加法可以避免局部拓扑剧烈变化情况的发生, 若路径 NCR_{path} 相等时, 则选择跳数较少的路径。

[0009] 通过 NCR-AODV 协议我们可以选取较为稳定的链路来传输数据, 从而可以很大程度上减少断路出现的概率, 也就减少了路由开销, 又可以进一步的减少端到端时延。但是这种稳定性路由选择方法只是考虑了当前时刻的邻居变化率, 稳定性还有待提升, 而且算法是在传统的 AODV 协议的基础上实现的, 因此仍然存在信息冗余的问题。

发明内容

[0010] 本发明的目的是针对上述现有技术的不足, 即路由开销过大, 稳定性偏低的问题, 提供一种实现低开销、低时延的融合稍带机制和加权稳定性算法的 AODV 协议。

[0011] 为实现上述目的, 本发明采用如下设计方案:

[0012] 基于改进 AODV 协议的通信方法, 该通信方法包括两部分:

[0013] 首先判断在一个 Hello 周期内节点是否发送了 RREQ, RREP 控制包或 DATA 数据包, 若没有发送则正常传输 Hello 信息, 否则在这个周期内不发送 Hello 信息;

[0014] 其次求出网络中路径的稳定性, 这个过程是在各个节点处要创建一个新的邻居稳定性缓存列表, 将当前时刻的邻居稳定变化率存储并覆盖更新有限的缓存列表中最老的邻居稳定性信息, 只保留最近的稳定性信息, 然后对当前时刻和历史时刻的稳定性由系数 λ 进行加权, 计算出节点的加权邻居稳定性进而根据相乘的方法求出网络中路径的稳定性, 最后选择最稳定或稳定性相同但跳数更少的路径来传输数据包。

[0015] 所述加权的邻居稳定性的实现方法, 具体步骤如下:

[0016] (1) 选择与 Hello 信息发送周期相同的时间间隔 T , 设定邻居集观察周期参数 α , 设定包括当前时刻在内的时刻数为 K , 且在 $0 \sim 1$ 范围内均匀的选取加权系数 λ ;

[0017] (2) 对于节点 i , 设定邻居集的观察周期为 $\alpha * T$, 并定义 $[t_r - \alpha * T, t_r]$ 时间段内所有出现过的邻居节点即邻居集为 S_{i,t_r} , 这样相隔时间间隔为 T 的两段时间就可得到

$$NCR_i(t) = \frac{|S_{i,t-T} \cap S_{i,t}|}{|S_{i,t-T} \cup S_{i,t}|}, \text{ 其中, } S_{i,t-T} \text{ 为 } t-T \text{ 时刻节点 } i \text{ 的邻居集, } S_{i,t} \text{ 为 } t \text{ 时刻节点 } i \text{ 的邻居集, } T \text{ 为发送 Hello 分组的时间间隔;}$$

居集, T 为发送 Hello 分组的时间间隔;

[0018] (3) 利用递推公式可得第 K 个 NCR 值 $NCR_i(t_k, t_{k+1}) = NCR_i(t_1 + (K-1)T, t_2 + (K-1)T)$, 对于这 K 个 NCR 都要缓存在节点新建的链表中, 超过 K 个时刻则进行覆盖更新, 当前时刻对应的 NCR 和这 K 个时刻 NCR 的平均值前面各乘以系数 β 和 $(1-\beta)$, 且满足 $0 \leq \beta \leq 1$; 在 t_{k+1} 时刻节点 i 的 NCR_i 为:

$$[0019] \quad NCR_i(t_{k+1}) = \lambda * NCR_i(t_k, t_{k+1}) + (1-\lambda) * \left[\frac{1}{K-1} * \sum_{j=1}^{K-1} NCR_i(t_j, t_{j+1}) \right]$$

[0020] 其中, $NCR_i(t_{k+1})$ 为当前时刻和历史时刻邻居变化率的加权, 加权系数分别是 λ 和 $1-\lambda$,

$$[0021] \quad \lambda = \beta + \frac{1-\beta}{K}, 1/K \leq \lambda \leq 1;$$

[0022] (4) 将沿每条路径上节点的加权邻居变化率累乘得到每条路径上的稳定性,选择稳定性数值最大或稳定性相等但是跳数最小的路径来发送数据包。

[0023] 所述发送 RREQ, RREP 控制包时,如节点需要向特定目的节点发送数据但是路由表中又没有相应的路由信息时就需要进行路由探寻,在分组中增加一个新的字段,这个字段用来存储节点的邻居稳定性度量 NCR_{path} ,目的节点接收到分组之后就相应 RREP,若目的节点离源节点较远,没有在源节点的一跳范围之内,就需要中间节点的转发来实现通信,若中间节点第一次接收到 RREQ 分组,则根据 RREQ 分组 NCR_{path} 字段的值在路由表中记录下来,并建立反向路由同时转发 RREQ,若接收到多个邻居转发的重复 RREQ,则与路由表中邻居稳定性度量相比较,若 RREQ 分组 NCR_{path} 字段数值更大,则更新路由表中邻居稳定性度量值和反向路径,以保证选用最稳定路由,之后继续转发出去,一直到达目的节点;对于到达目的节点的 RREQ,目的节点将 RREQ 中的信息复制到 RREP 沿着反向链路单播传送到源节点,同时建立前向路径,若以后收到序列号更大更新鲜或序列号相等但是 NCR_{path} 值更大的 RREQ 分组才需要响应 RREP 并更新路由;RREQ 和路由表中存储的都是链路沿途节点的邻居变化率的乘积值。

[0024] 本发明的详细技术方案如下:一种基于稍带机制和加权稳定性算法的改进 AODV 协议,该算法与原始协议相比具有更低的路由开销和更高的链路稳定性,从而得到在路由开销和端到端时延各方面的优化。这个改进算法分两个部分组成:首先在 AODV 协议基础上引进了稍带机制,这样节点在一个 Hello 时间间隔内若向其邻居发送或转发了控制包或控制包等包含节点信息的载体则不必再发送 Hello 信息;其次就是针对链路稳定性方面算法的改进,将邻居变化率作为链路稳定性的决定性因素,并且将当前时刻的邻居变化率和前几个相同间隔的时间段的历史邻居变化率综合考虑,按照适当的加权因子进行加权处理,从而使传输数据包的路径更加稳定,使网络拓扑的鲁棒性更强。最终将这两方面的改进在 AODV 协议实现,对于协议各个方面的性能得以很大程度的改进。

[0025] 一种加权邻居稳定性的实现方法,它的方法为:

[0026] 将邻居变化率 (NCR) 作为链路稳定性的判断因素,根据选择的适当邻居集观察周期以及适当的时间间隔来计算当前时刻和恰当数量的历史时刻的 NCR,最后选用合理的加权系数对这些 NCR 进行加权。

[0027] 它的具体步骤为:

[0028] (1) 选择与 Hello 信息发送周期相同的时间间隔 $T = 1s$,为避免信息在 MAC 层的不可靠传输设定邻居集观察周期参数 $\alpha = 3$,为保证动态稳定的反映网络拓扑设定包括当前时刻在内的时刻数为 $K = 4$,最后在 $0 \sim 1$ 范围内均匀的选取加权系数 λ ;

[0029] (2) 由当前时刻及其之前相隔 T 时刻的邻居集交集与并集的比值来确定当前时间的邻居变化率 NCR,并将其保存在节点缓存邻居变化率的相应的路由表中,且继续缓存,直到缓存数目超过路由表中设置的历史时间点的数目 K 时再从最久的时间点起依次覆盖;

[0030] (3) 计算某个时刻的加权邻居变化率,将当前时刻和之前的 $K-1$ 个历史时刻的邻居变化率按照加权系数 λ 进行加权;

[0031] (4) 将沿每条路径上节点的加权邻居变化率累乘得到每条路径上的稳定性,选择

稳定性数值最大或稳定性相等但是跳数最小的来发送数据包。

[0032] 在协议中计算节点邻居集时,不只是计算某一个瞬时时刻节点邻居集,而是适当的延长了节点邻居集的观察周期,这样可以有效的避免邻居消息的携带分组在 MAC 层的不可靠传输对邻居变化率的影响。因此,对于节点 i ,设定邻居集的观察周期为 $\alpha * T$,并定义 $[t_r - \alpha * T, t_r]$ 时间段内所有出现过的邻居节点即邻居集为 S_{i,t_r} (α 选取的值为 3 时最佳)。这样相隔时间间隔为 T 的两段时间就可以得到一个 NCR,如在 t_2 时刻就可以计算出节点的一个邻居稳定性的值 $NCR_i(t_1, t_2)$,同样在 $t_3 = t_2 + T$ 处,根据相同的方法可以得到第二个 $NCR_i(t_2, t_3) = NCR_i(t_1 + T, t_2 + T)$,以此类推我们就可以得到第 K 个 NCR 值 $NCR_i(t_k, t_{k+1}) = NCR_i(t_1 + (K-1)T, t_2 + (K-1)T)$ 。

[0033] 对于这 K 个 NCR 都要缓存在节点新建的链表中,超过 K 个时则进行覆盖更新,当前时刻对应的 NCR 和这 K 个时刻 (包括当前时刻和历史时刻) NCR 的平均值前面各乘以系数 β 和 $(1-\beta)$,且满足 $0 \leq \beta \leq 1$ 。在 t_{k+1} 时刻节点 i 的 NCR_i 为:

$$\begin{aligned}
 [0034] \quad NCR_i(t_{k+1}) &= \beta * NCR_i(t_k, t_{k+1}) + (1-\beta) * \left[\frac{1}{K} * \sum_{j=1}^K NCR_i(t_j, t_{j+1}) \right] \\
 [0035] \quad &= \left(\beta + \frac{1-\beta}{K} \right) * NCR_i(t_k, t_{k+1}) + \frac{1-\beta}{K} * (K-1) * \left[\frac{1}{K-1} \sum_{j=1}^{K-1} NCR_i(t_j, t_{j+1}) \right] \quad (3) \\
 [0036] \quad &= \lambda * NCR_i(t_k, t_{k+1}) + (1-\lambda) * \left[\frac{1}{K-1} * \sum_{j=1}^{K-1} NCR_i(t_j, t_{j+1}) \right]
 \end{aligned}$$

[0037] 开始制定公式 (3) 第一行最右侧的邻居变化率平均值时,将当前时刻的邻居变化率也考虑进去的目的在于在当前时刻邻居变化率影响最小,也就是 $\beta = 0$ 时也要保证当前时刻与历史时刻的邻居变化率拥有同样的地位。通过公式变形我们最终可以得到当前时刻和历史时刻邻居变化率的加权,加权系数分别是 λ 和 $1-\lambda$,其中 $\lambda = \beta + \frac{1-\beta}{K}$, $\frac{1}{K} \leq \lambda \leq 1$ 。这样通过调整 λ 的值就可以改变当前时刻和历史时刻邻居变化率对节点稳定性的影响程度了, λ 的值越大,节点受当前时刻影响越大,反之,节点受历史因素的影响越大。特别地,当 $\lambda = 1$ 时改进的 NCR-AODV 等同于 NCR-AODV 中节点邻居变化率的定义。

[0038] 和 NCR-AODV 协议一样,我们在整个链路上邻居变化率的度量值也定义为这条路径上所有节点的邻居变化率的乘积,虽然改进的 NCR-AODV 协议中 NCR_i 是 0 的概率很小,但是采用乘积的方式和求和方式相比更能够如实的反映节点邻居变化率的波动程度。

[0039] 对 ns-2.31 版本中的 AODV 协议进行的改进扩展,对于 AODV 协议的改进如下:

[0040] (1) 在 AODV 协议中添加捎带机制的改进,对 Hello 消息的发送条件进行了适当的限制,而原始 AODV 却是周期性的广播 Hello 分组;

[0041] (2) 与 NCR-AODV 协议相同,在 RREQ, RREP 分组中都添加了邻居稳定性路由度量 NCR_{path} 字段,中间节点转发时也要据此进行路由的更新,且中间节点不能直接回复 RREQ;

[0042] (3) 每个时间段内节点的邻居变化率都要缓存在节点新建了 NCR 存储链表中,保留最近的 K 个值,进行覆盖更新;

[0043] (4) 最终选择目的节点序列号最大或序列号相等但是稳定性路由度量最大的路径,更进一步,若这两者都相等,则选跳数较小的来传输 DATA 分组。

[0044] 与 AODV 协议一样,改进 NCR-AODV 协议中节点需要向特定目的节点发送数据但是

路由表中又没有相应的路由信息时就需要进行路由探寻,即广播 RREQ 分组,不同之处在于分组中增加了一个新的字段,这个字段用来存储节点的邻居稳定性度量 NCR_{path} (不是公知技术, NCR-AODV 中提出的),目的节点接收到分组之后就相应 RREP,若目的节点离源节点较远,没有在源节点的一跳范围之内,就需要中间节点的转发来实现通信,若中间节点第一次接收到 RREQ 分组,则根据 RREQ 分组 NCR_{path} 字段的值在路由表中记录下来,并建立反向路由同时转发 RREQ,若接收到多个邻居转发的重复 RREQ,则与路由表中邻居稳定性度量相比较,若 RREQ 分组 NCR_{path} 字段数值更大,则更新路由表中邻居稳定性度量值和反向路径,以保证选用最稳定路由,之后继续转发出去。按照上面的策略一直到达目的节点。对于到达目的节点的 RREQ,目的节点将 RREQ 中的信息复制到 RREP 沿着反向链路单播传送到源节点,同时建立前向路径,若以后收到序列号更大更新鲜或序列号相等但是 NCR_{path} 值更大的 RREQ 分组才需要响应 RREP 并更新路由。在整个过程中节点邻居稳定性按照 (3) 式计算, RREQ 和路由表中存储的都是链路沿途节点的邻居变化率的乘积值。

[0045] 有益效果:本发明方法具有以下优势:(1) 综合考虑当前时刻和历史时刻的邻居变化率因素,提高了稳定性评价标准的精确度;(2) 可以通过调整节点的加权系数使得节点稳定性更好的反映网络局部拓扑变化情况。本发明将稍带机制和邻居稳定性算法结合起来来进行协议的优化,融合了稍带机制和加权邻居稳定性算法两者的优势从而使路由开销和端到端时延性能得到很大的提高。

附图说明

[0046] 图 1 是各个时间段内节点邻居稳定性定义;

[0047] 图 2 是本发明的实现流程图;

[0048] 图 3(a) 是在不同的移动速度,本发明提出的路由开销性能;

[0049] 图 3(b) 是在不同的暂停时间下,本发明提出的路由开销性能;

[0050] 图 4(a) 是在不同的移动速度,本发明提出的时延性能;

[0051] 图 4(b) 是在不同的暂停时间下,本发明提出的时延性能;

具体实施方式

[0052] 本发明针对现有 AODV 协议、稍带机制以及当前链路稳定性算法的现状分析,设计出了一种能够将上述三个方面进行适当改进融合的方案,不仅可以使路由开销明显减少同时在很大程度上降低了时延,提高了 AODV 协议在无线网络的通信性能。

[0053] 下面结合附图与实施例对本发明做进一步说明。

[0054] 图 1 是定义的几个时间段内节点邻居稳定性定义。

[0055] 在协议中,不只是计算某一个瞬时时刻节点邻居集,而是适当的延长了节点邻居集的观察周期,这样可以有效的避免邻居消息的携带分组在 MAC 层的不可靠传输对邻居变化率的影响。因此,对于节点 i ,设定邻居集的观察周期为 $\alpha * T$,并定义 $[t_r - \alpha * T, t_r]$ 时间段内所有出现过的邻居节点即邻居集为 S_{i-tr} (α 选取的值为 3 时最佳)。这样相隔时间间隔为 T 的两段时间就可以的到一个 NCR,如在 t_2 时刻就可以计算出节点的一个邻居稳定性的值 $NCR_i(t_1, t_2)$,同样在 $t_3 = t_2 + T$ 处,根据相同的方法可以得到第二个 $NCR_i(t_2, t_3) = NCR_i(t_1 + T, t_2 + T)$,以此类推我们就可以得到第 K 个 NCR 值 $NCR_i(t_k, t_{k+1}) = NCR_i(t_1 + (K-1)T,$

$t_2 + (K-1)T$ 。

[0056] 图 2 是本发明的实现流程图。

[0057] 这个改进算法是在 AODV 协议上实现的,具体的改进分引入捎带机制和引入加权的稳定性算法两部分组成。

[0058] 具体的做法为:首先引入捎带机制要判断在一个 Hello 周期内节点是否发送了 RREQ, RREP 等控制包或 DATA 数据包,若没有发送则正常传输 Hello 信息,否则在这个周期内不发送 Hello 信息以避免冗余信息的传输;其次引入加权的邻居稳定性算法,这个过程是在各个节点处要创建一个新的邻居稳定性缓存列表,将当前时刻的邻居稳定变化率存储并覆盖更新有限的缓存列表中最老的邻居稳定性信息,只保留最近的稳定性信息,然后对当前时刻和历史时刻的稳定性由一适当的系数 λ 进行加权,最终计算出节点的加权邻居稳定性进而根据相乘的方法求出网络中路径的稳定性,最后选择最稳定或稳定性相同但跳数更少的路径来传输数据包。

[0059] 本发明所获得的效果可以通过图 3(a) (b)、图 4(a) (b) 来很好的说明。

[0060] 为了说明本发明相对于传统的 AODV 和 NCR-AODV 能够很好的改进路由开销和端到端时延的优越之处,图 3 和图 4 给出了分别在不同的移动速度和暂停时间下所提出改进协议的路由开销和端到端时延的性能。

[0061] 仿真条件:在 NS2.31 版本下进行仿真,在以 250m 作为发送范围的 1500m*900m 的仿真区域内随机放置 50 个节点,有 30 对源到目的节点 CBR 数据流,每秒发送 2 个 512byte 大小的数据包,选用 Random way-point 模型作为节点的移动模型,最大移动速度分别为:0m/s,5m/s,10m/s,15m/s,20m/s,暂停时间分别为:0s,100s,200s,300s,400s,500s,600s,700s,800s,900s,Hello 信息的发送间隔 $T = 1s$,邻居集观察周期系数 α 设为 3,缓存的总的邻居变化率个数 $K = 4$,邻居稳定性的加权系数 λ 分别取较为均匀的三个值:0.475、0.7、1,以 900s 的仿真时间仿真 6 次取平均值。

[0062] 首先,关于网络的路由开销。图 3(a) 统计了协议路由开销随速度的变化趋势,从图中我们可以看到,随着速度的增大,路由开销由于断链增加出现递增趋势。进一步可以看到我们的改进算法在路由开销方面优势明显,最主要的原因在于引入捎带机制大大减少了 Hello 分组的发送数目,使得整体路由开销降低。而 NCR-AODV 协议选择稳定链路,避免频繁断链,路由开销自然就少。由于上面提到的两个方面的原因,Improved NCR-AODV 在路由开销上取得了更大的优势,但是路由开销随速度的变化和 λ 值的不同略有变化,当速度较小时 λ 的值较小时路由更加稳定,因此路由开销就小,反之,速度较大时, λ 值越大路由开销性能反而越好。图 3(b) 仿真出了各协议路由开销随暂停时间的变化趋势,各协议的路由开销随着暂停时间的增加表现出整体减小的趋势。与图 3(a) 表现出来的情况一致,NCR-AODV 协议的路由开销性能优于 AODV 协议,但由于捎带机制 Improved AODV 协议和 ImprovedNCR-AODV 路由开销性能更好,同样的,在暂停时间较小时,节点移动性较强,ImprovedNCR-AODV ($\lambda = 1$) 取得了更好的性能,反之暂停时间在 400s ~ 900s,系统移动性较弱时,Improved NCR-AODV ($\lambda = 0.475$) 稳定性最佳,路由开销最小。

[0063] 其次,关于端到端时延。图 4(a) 显示了各协议端到端时延随速度的变化曲线,从图中我们首先可以看出随节点移动速度增加各协议端到端时延呈现递增趋势,这是由于移动速度增加引起的断链和网络剧烈变化导致路由修复的增加,同时又进一步加重了

网络拥塞状况从而增加时延。ImprovedAODV 相比于 AODV 来说时延性能有了很大提高,原因还是在于捎带机制大大减少了网络的路由负载,因此减少了由于链路中控制分组和数据分组冲突以及排队过程中引起的时延。与路由开销的仿真结果相呼应,节点在低速时 ImprovedNCR-AODV ($\lambda = 0.475$) 在端到端时延这一性能方面表现的更为突出,而在高速时,对于实时稳定性要求较高,因此 Improved NCR-AODV ($\lambda = 1$) 协议端到端时延最小,性能最好。图 4(b) 显示各协议端到端时延随暂停时间的变化情况,各协议随着节点暂停时间的延长端到端时延也越来越小。我们提出的 ImprovedNCR-AODV 由于基于 ImprovedAODV 做的改进,因此继承了 Improved AODV 协议 Hello 分组在受限条件下发送的时延优势,并且,与图 4(a) 一样,暂停时间较短时, λ 值大的协议性能好,反之,暂停时间较长时则 λ 较小协议可以在时延方面性能达到最优。

[0064] 因此,我们可以很清晰的看出,我们提出的改进方案(包括不考虑链路稳定性和考虑链路稳定性的捎带机制方案)相较于传统的 AODV 协议和 NCR-AODV 协议来说路由开销和时延方面有很大的改进。并且在不同的应用环境下(高低速,长短暂停时间)可以选择适合的加权系数来获得更好的性能。

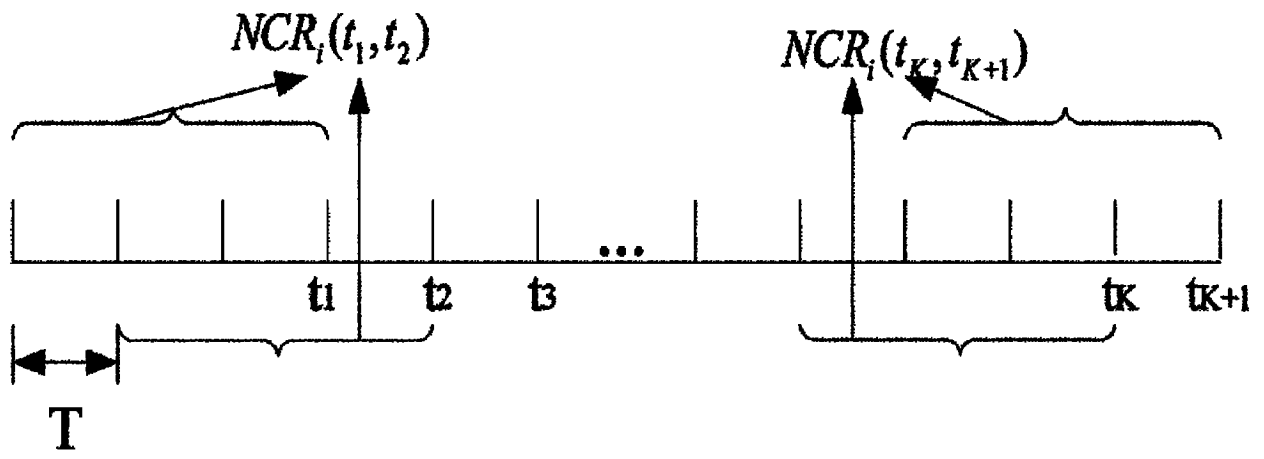


图 1

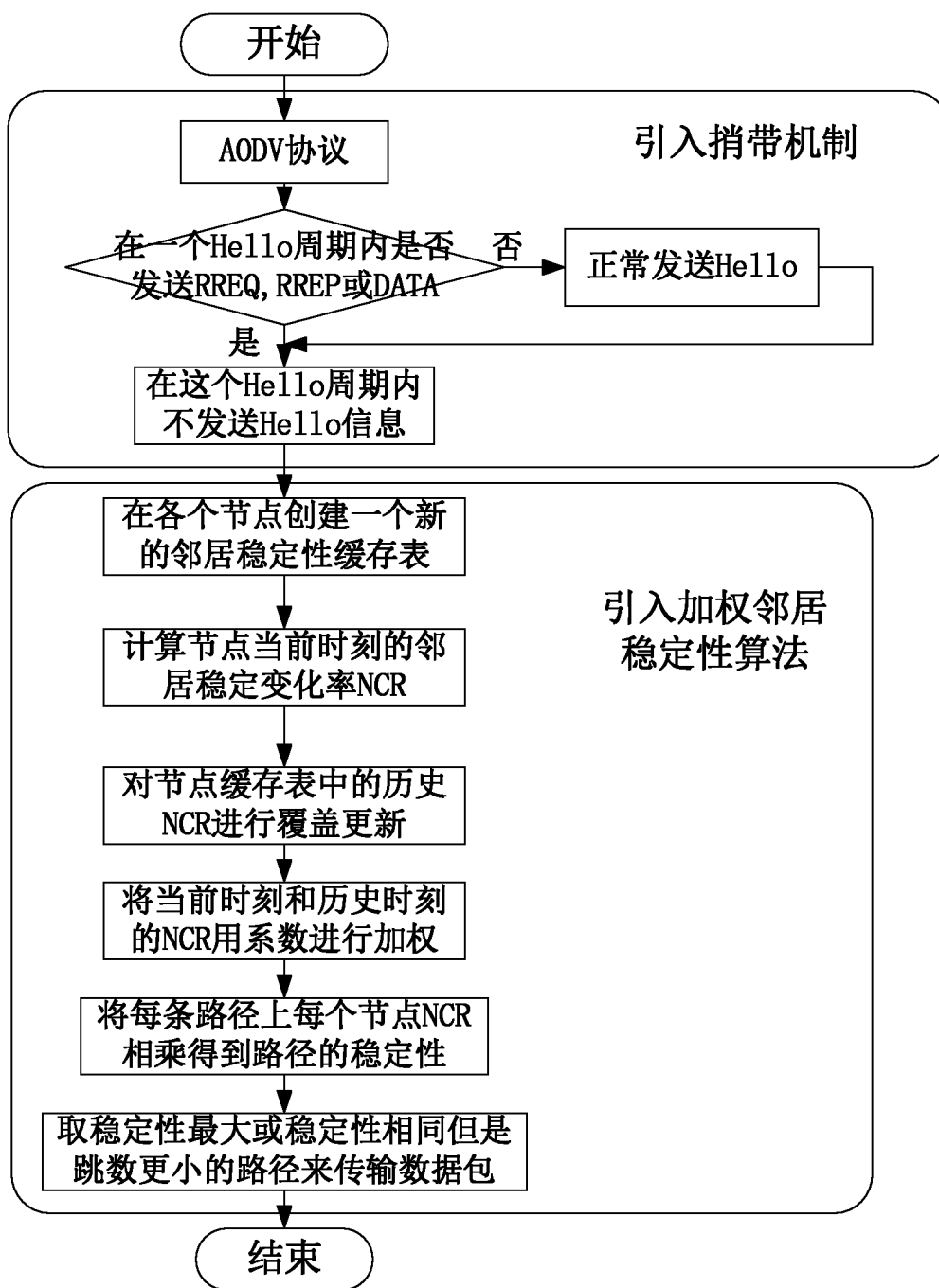


图 2

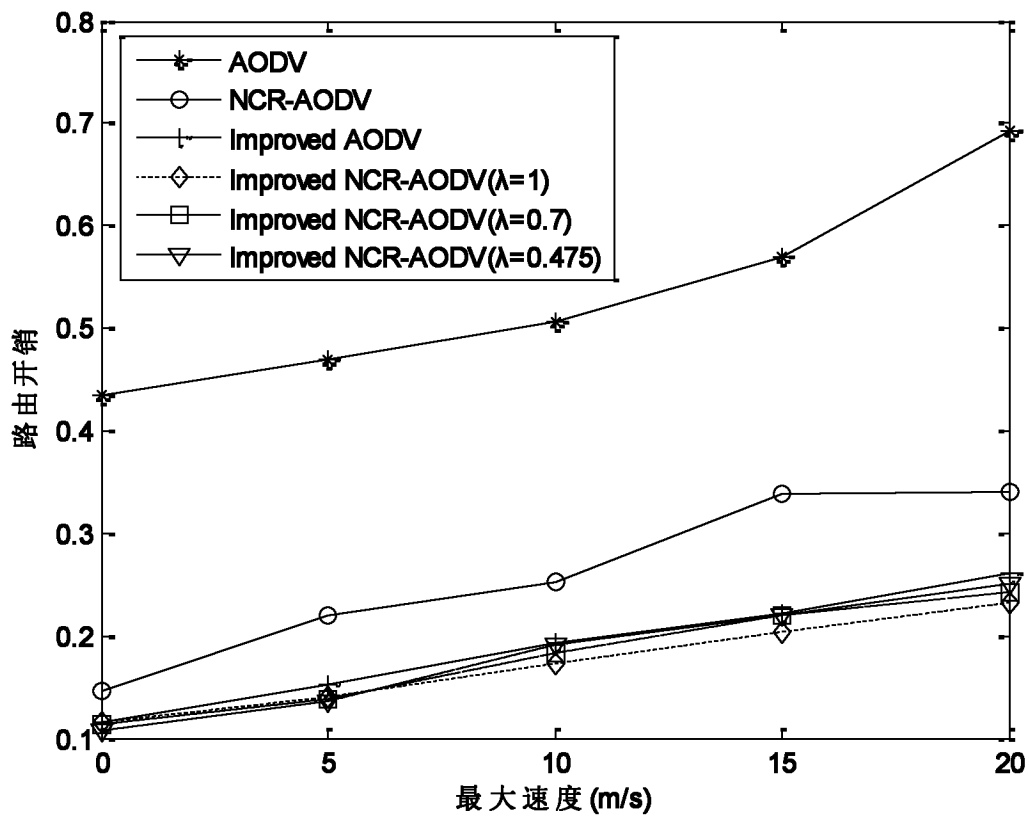


图 3(a)

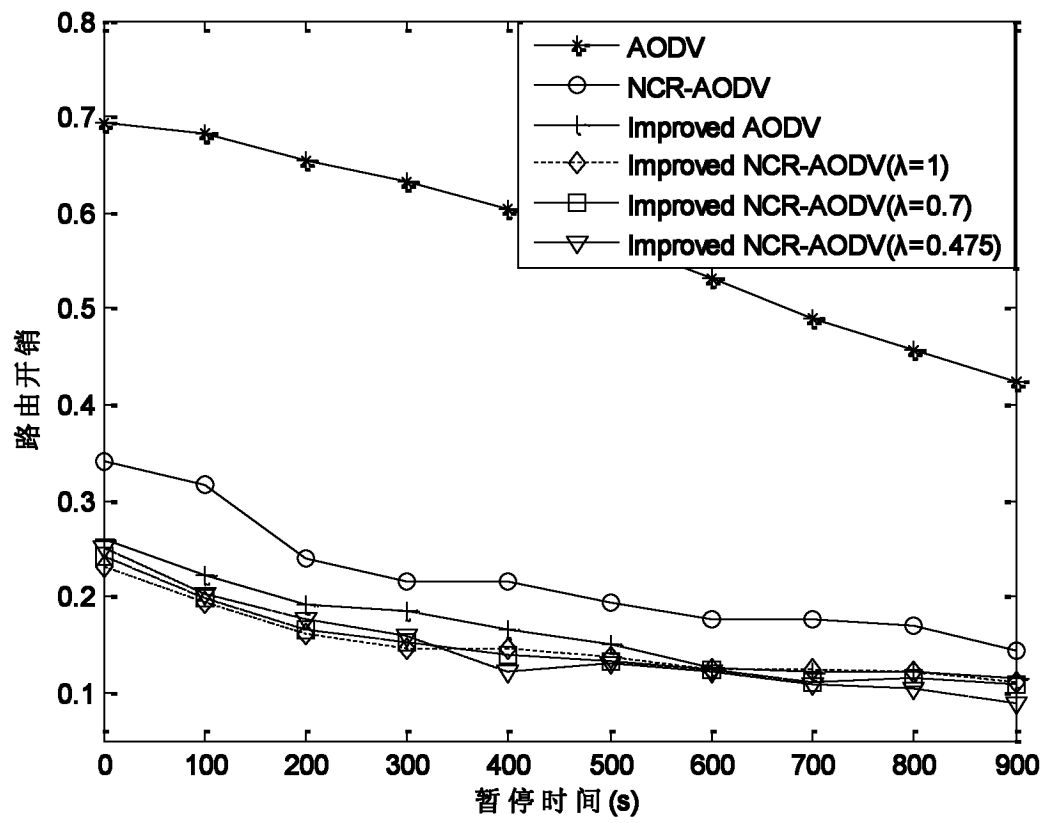


图 3(b)

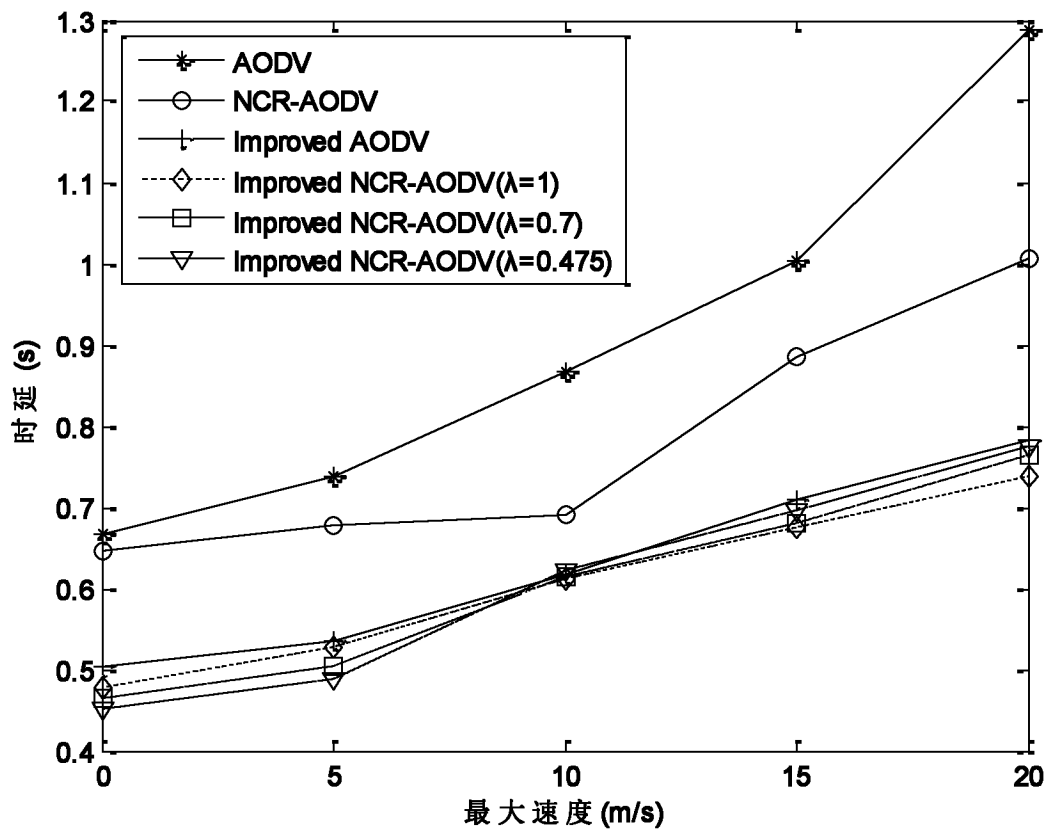


图 4(a)

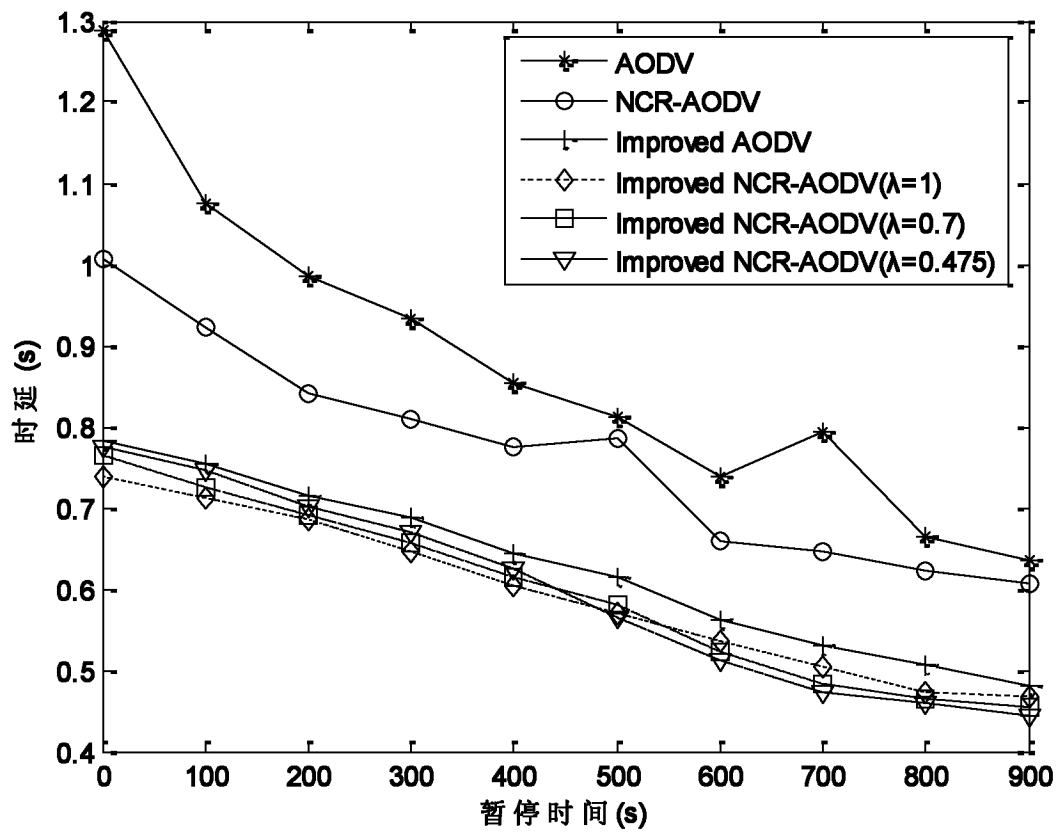


图 4(b)