

一种 Ad hoc 网络中分布式无环路由算法设计

康 凯 郭 伟 吴诗其

(电子科技大学通信抗干扰国防重点实验室 成都610054)

A Novel Distributed Loop-free Routing Algorithm in the Ad hoc Networks

KANG Kai GUO Wei WU Shi-Qi

(National Communication Lab, UESTC, Chengdu 610054, China)

Abstract The routing algorithm is one of the important problems in the Ad hoc networks. The performances of the networks are heavily depended on the property of the routing algorithm. In this paper, we present a new loop-free routing algorithm in the Ad hoc networks. By virtue of the correlation of routing information among neighbor nodes, the formation of the temporary loops during the topological changes can be prohibited. The new algorithm can avoid the waste of the wireless bandwidth, and accelerate the convergence of the routing during the network transitions.

Keywords Ad hoc network, Routing, Loop-free, Temporary loop

1 引言

自组网,又称为 Ad hoc 网,是指一组带有无线收发装置的移动节点组成的一个多跳的临时性自治系统^[3]。自组网不需要依赖于任何预先架设的网络设施,能够实现临时快速自动组网和支持节点的移动性,并具有动态网络管理和动态网络优化的功能,主要用于在高度机动和无线通信的环境下为移动节点提供通信服务。在军事通信、抢险救灾、计算机支持协同工作(CSCW)等方面都具有广泛的应用^[1~3]。

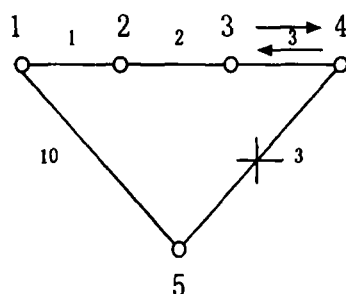


图1 3、4节点间暂态环路的形成

自组网的拓扑结构具有动态变化的特性,因此路由算法是自组网中最为重要的问题之一。自组网的路由协议用于监控网络拓扑变化,交换路由信息,产生、维护和选择路由,并根据选择的路由转发数据包。选择一个好的路由算法是保证网络性能的主要因素^[3,4]。自组网的路由算法通常可以分为以下几种:固定式路由算法;集中式路由算法;孤立式路由算法;分

布式路由算法,其中以分布式路由算法应用得最为普遍。本文集中讨论自组网中的分布式路由算法。

自组网中的路由算法设计中必须考虑移动性造成的网络拓扑频繁变化的问题,包括链路的增加、减少和链路代价的变化等,在相关的更新消息的扩散达到收敛过程中,会造成网络中的路由出现暂态环路。环路会消耗有限的无线带宽资源。对于暂态环路问题,可见图2、3示例。

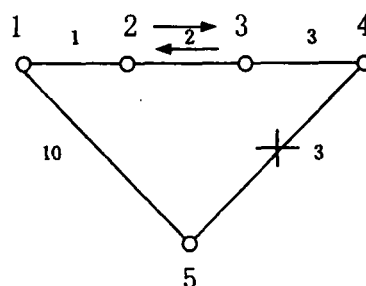


图2 2、3节点间暂态环路的形成

当因某种原因链路 l_{45} 断开时,节点4发往其他节点的数据包以3节点为下一跳,而3节点在未接收到有关状态更新消息时,仍将1、2、3节点发往5节点的数据包以4节点作为下一跳,因此在链路 l_{34} 上会形成3、4节点之间的暂态环路。当3节点从4节点接收到有关状态更新消息后,会修改自己的路由表。此时 l_{34} 上的暂态环路会消除,但在 l_{23} 上会形成同样的环路。

解决移动环境下的暂态环路抑制的最优路由算法是自组

- 4 Braun T, Gunter M, Khail I. Virtual Private Network Architecture. April 1999. URL: <http://www.tik.ee.ethz.ch/~cati/>
- 5 Blake S, Black D, Carlson M, et al. An Architecture for Differentiated Services. IETF RFC 2475, Dec. 1998
- 6 林闯,单志广,盛立杰,等. Internet 区分服务及其几个热点问题的研究. 计算机学报, 2000, 23(4): 419~433
- 7 Duffield N G, Goyal P, Greenberg A, et al. A flexible Model for Resource Management in Virtual Private Networks
- 8 Kent S, Atkinson R. Security Architecture for the Internet Protocol. IETF RFC 2401, Nov. 1998
- 9 Braden R, Clark D, Shenker S. Integrated Services in the

- Internet Architecture: an Overview. IETF RFC 1633, June 1994
- 10 Clark D. The Design Philosophy of the DARPA Internet Protocols. ACM SIGCOMM'88, Aug. 1988
- 11 Berner Y, Ford P, Yavatakr R, et al. A Framework for Integrated Services Operation over Diffserv Networks. IETF RFC 2998. Nov. 2000
- 12 Nichols K, Blake S, Baker F, et al. Definition of the Differentiated Services Field (DS Field) in the IPv4 and IPv6 Headers. IETF RFC 2474, Dec. 1998
- 13 Nichols K, Jacobson V, Zhang L. A Two-bit Differentiated Services Architecture for the Internet. IETF RFC 2638, July 1999

网中的一个关键问题^[5,6],目前在这一方面的研究较少。本文中提出了一种新的分布式暂态环路抑制路由算法,能够有效地抑制网络拓扑变化时暂态环路的形成。

2 分布式无环路由算法的设计原理

本文所提出的算法是基于表驱动方式的链路状态(link-state)方法。将网络抽象成为一个带权图 $G=(V,L)$, V 表示节点的集合, L 表示边的集合。每个节点代表网络中的一个节点,每条边代表一条链路,赋予一个权值。权值可以是跳数、信道带宽、平均通信量或根据其他一些因素计算出来的。节点之间交换拓扑信息,建立连接矩阵,根据所维护的网络连接矩阵,利用 Dijkstra 算法计算路由。多路径 Dijkstra 算法可以完成对于每个目的节点,从本节点的邻节点的集合中,找出一个或一组节点,使得以此作为下一跳至目的节点的路径距离最短。

通常的路由算法中,每个节点独立地根据自己对网络拓扑的认识,即所维护的网络连接矩阵,计算各自的路由表。在稳态时的各节点对网络的认识是一致的。此时根据 Dijkstra 算法计算出的最短路径是一个有向树;对于维护多个等效路径的多路径 Dijkstra 算法,计算出的到各个节点的最短路径是一个有向非循环图,均不存在环路。在网络拓扑变化时,由于链路状态更新消息扩散过程中的随机延迟,或者同时在网络中扩散不同的链路状态更新消息,各节点对网络拓扑认识,即维护的连接矩阵,会出现暂态的不一致。由于各节点独立地计算路由表,因此可能会出现暂态的路由环路。我们称网络中没有链路变化时的状态为稳态,在链路变化时,当链路状态通知在网络中扩散,且未达到收敛时的状态称为暂态。收敛指网络中各节点维护的网络中路由达到一致。

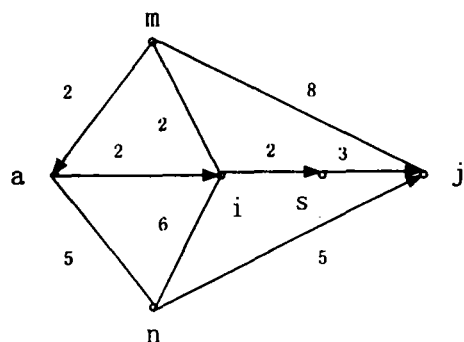


图3 网络路由中的偏序关系

本文中所提出的无环路由算法是基于偏序的概念。如果节点 i 在节点 p 到目的节点 j 的最短路径上,节点 i 会接收到节点 p 发往目的节点 j 的数据包,并转发到下一跳,则定义节点 p 为节点 i 关于目的节点 j 的一个上游节点;相应地,称节点 i 为节点 p 的一个下游节点。我们称这种上游节点和下游节点间的顺序关系为一个偏序关系($\rightarrow$)。偏序关系反映了网络路由中的顺序关系。在稳态的情况下,下游节点将接收到上游节点的数据包转发到下一跳,而上游节点不会接收到下游节点的数据包,这时候称偏序关系是一致的。而当网络拓扑结构发生变化时,由于节点对网络的认识是异步更新的,因此节点间的偏序关系可能是不一致的。这种各节点维护的路由顺序的不一致就可能会导致环路。这可以通过以下的一个示例来加以说明。

图3中对于目的节点 j 有偏序关系 $m \rightarrow i \rightarrow j$ 和 $n \rightarrow j$,

节点 m 为节点 a, i 关于 j 的一个上游节点。当链路 L_{ni} 断开时,节点 i 更新后到节点 j 的最短路径如图4中虚线所示。节点 i 选择节点 m 为到节点 j 的下一跳,而在节点 m 未对路由表做一致性修改时,会在节点 i, m, a 之间形成一个环路。

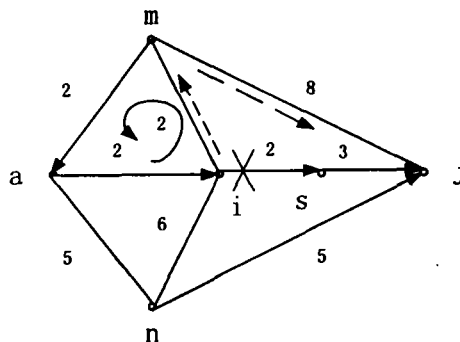


图4 链路 L_{ni} 断开时 i, m, a 节点间环路的形成

文[5,6]中指出,对于目前几种已有的无环路由算法中,要求维护全部路径信息,不仅开销很大,而且在分布式系统中实现困难。在本文所提出的算法中,只限于根据所维护的不同相邻节点之间的相关性信息,进行暂态环路检测。相邻节点间交换有关信息,建立起节点之间偏序关系的相关性,只有在满足一定约束条件时,才能选择某个节点作为下一跳。

相邻节点间的偏序关系利用与节点间到目的节点的距离关系得到。下面先介绍本文中将要用到的一些符号及其意义。

n : 网络中的节点数。 L_{ij} : 表示网络中 i, j 节点之间的链路。 d_{ij} : 表示链路 L_{ij} 的等效距离,即链路的权值。设不相邻的两个节点之间的距离为 ∞ 。 r'_i : 表示节点 i 的路由表中记录的到目的节点 j 的最短距离。 $neib(i)$: i 节点的邻节点的集合,即 $neib(i) = \{j | d_{ij} \neq \infty, d_{ij} \neq 0, j \in V\}$ 。

对于偏序关系 $p \rightarrow i \rightarrow j$,存在节点间如下的距离关系式:

$$r'_i = r'_p + r'_{ij} \quad (1)$$

特别地,对于 p, i 是相邻节点的情况,有:

$$r'_i = d_{pi} + r'_j \quad (2)$$

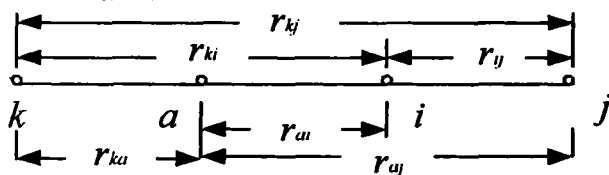


图5 偏序关系的距离示例

如图5所示,节点 k 为节点 a, i 关于目的节点 j 的上游节点,节点 k, a 为节点 i 关于目的节点 j 的上游节点。偏序关系有 $k \rightarrow a \rightarrow i \rightarrow j$,且存在如下的关系式:

$$r'_i = r'_k + r'_{ij} \quad (3)$$

$$r'_i = r'_a + r'_{ij} \quad (4)$$

$$r'_j = r'_a + r'_{ij} \quad (5)$$

在本文所提出的无环路由算法中,网络中的每个节点维护的数据结构包括一个网络的连接矩阵,一个网络的路由表,一个标志矩阵,一个距离矩阵,分别定义如下:

连接矩阵: 节点 i 的连接矩阵记为 $D_i = (d_{ij})_{n \times n} (i, j \in V)$ 。 d_{ij} 表示链路 L_{ij} 的权值,设不相邻的两个节点之间的距离为 ∞ 。本文讨论的是对称链路,即有 $d_{ij} = d_{ji}$ 。

路由表: 节点 i 的路由表记为 $\langle R', Next \rangle (i \in V)$ 。 $R' = (r'_j)_{j \in V}$, r'_j 表示节点 i 到节点 j 的距离。 $Next = (next'_j)_{j \in V}$ 。

($j \in V$), $next'_i$ 表示节点 i 到节点 j 的下一跳节点, 当存在多个等效路径时为节点的集合。

距离矩阵: 节点 i 的距离矩阵记为 $C = (C_a)_{|neib(i)|} = (c_{a,j})_{|neib(i)| \times n}$, ($a \in neib(i)$, $j \in V$)。距离矩阵 C 中, 节点 i 对每个邻节点 a 都建立一个距离向量 $C_a = (c_{a,j})_n$ ($j \in V$)。 $c_{a,j}$ 为节点 i 所记录的邻节点 a 到目的节点 j 的距离。在稳态时, 有 $c_{a,j} = r_j^i$; 而在暂态期内, 两者可能是不相等的。 C 初始化时均置为 $c_{a,j} = \infty$ ($j \in V, j \neq a$), $c_{a,a} = 0$ 。

标志矩阵: 节点 i 的标志矩阵记为 $T = (T_a)_{|neib(i)|} = (t_{a,j})_{|neib(i)| \times n}$, ($a \in neib(i)$, $j \in V$)。标志矩阵 T 中, 节点 i 对每个邻节点 a 都建立一个标志向量 $T_a = (t_{a,j})_n$ ($j \in V$)。 T_a 反映了节点 i 所维护的与邻节点 a 之间的偏序关系。如果对目的节点 j , 节点 i 与邻节点 a 之间存在偏序关系 $a \rightarrow i \rightarrow j$, 则 $t_{a,j} = 1$, 否则 $t_{a,j} = 0$ 。 T 初始化时均置为 $t_{a,j} = 0$ ($j \neq a, j \in V$), $t_{a,a} = 1$ 。

每个节点根据网络的连接矩阵 D' 计算出路由表 $\langle R', Next' \rangle$, 根据 R_i 和距离矩阵 D' 或 S' 来对标志矩阵 T 进行修改, 根据 T 和 $Next'$ 检测或者消除与相邻节点之间的暂态环路。

节点间的相关性信息通过相邻节点间交换网络状态通知 (NSA) 消息实现。节点 i 的网络状态通知消息 (NSA) 包括了距离状态通知 (DSA) 和链路状态通知 (LSA) 两部分, 以及序列号。其中距离状态通知消息包括了 i 节点需要通知邻节点的一项或多项更新了 r_j^i 的值。链路状态通知包括了节点 i 认识到的网络中新的链路变化情况。如果没有新的链路变化情况, 则网络状态消息只有距离状态消息部分。网络中的节点根据检测到相邻链路的变化和接收到邻节点 NSA 消息通知的 r_j^i 的值, 更新 $c_{a,j} = r_j^i$, 并判断邻节点 a 是否是一个上游节点, 并对相应的标志位 $t_{a,j}$ 进行定义:

$$\begin{cases} \text{if}(c_{a,j} \geq r_j^i + d_{a,i}) \text{ then } t_{a,j} = 1 \\ \text{if}(c_{a,j} < r_j^i + d_{a,i}) \text{ then } t_{a,j} = 0 \end{cases} \quad (j \in V) \quad (6)$$

每个节点的数据结构中维护了两种不同的偏序关系:

偏序关系 1: 节点 i 的路由表项 $next'_i = a$ 表示了与邻节点 a 之间关于目的节点 j 的偏序关系为 $i \rightarrow a \rightarrow j$, 即节点 i 是节点 a 关于目的节点 j 的一个上游节点。

偏序关系 2: 节点 i 的标志向量 T_a 表示了节点 i 所记录的邻节点 a 维护的与节点 i 之间关于目的节点 j 的偏序关系:

如果 $t_{a,j} = 1$, 则有偏序关系: $a \rightarrow i \rightarrow j$, 即节点 a 是节点 i 关于目的节点 j 的一个上游节点。

如果 $t_{a,j} = 0$, 则表明不存在偏序关系 $a \rightarrow i \rightarrow j$, 即节点 a 不是节点 i 关于目的节点 j 的一个上游节点。

环路的检测是通过对这两种偏序关系之间的判断来实现的。环路检测包括了环路生成检测和环路消除检测两种。

(1) 环路生成检测 是判断所选择的下一跳节点是否满足无环约束条件:

对于路由表项 $next'_i = a$, 如果标志位为 $t_{a,j} = 0$, 则 $next'_i = a$ 为一个可行的转发表项。

如果标志位 $t_{a,j} = 1$, 则节点 i 以节点 a 为下一跳发往目的节点 j 的数据包, 会被节点 a 重新返回节点 i , 因此在 i, a 节点之间形成一个暂态环路, $next'_i = a$ 不能作为下一跳。此时节点 i 应封锁路由表中 $next'_i = a$ 表项。

(2) 环路消除检测 对于封锁的路由表项 $next'_i = a$, 如果更新后的 $t_{a,j} = 0$, 则表明已经不存在 $a \rightarrow i \rightarrow j$ 的偏序关系, 成为一个可行的转发表项。

本文所提出的分布式无环路由算法可以描述如下:

(1) 初始组网 (节点 i):

1. 与邻节点之间交换可达性信息, 建立连接矩阵 D' 。

2. 根据连接矩阵 D' 计算路由表 $\langle R', Next' \rangle$ ($i \in V$)。

3. 生成 NSA 消息, 在 DSA 部分通知 r_j^i ($j \in V, j \neq i$) 的值。初始组网时的 NSA 消息只有 DSA 部分。

4. 根据接收到邻节点 a 的 NSA 消息, 更新 $C_a: c_{a,j} = r_j^i$ ($j \in V, j \neq i$), $c_{a,a} = 0$ 。

5. 根据 (2) 式更新 T_a 。

(2) 检测到相邻链路变化:

1. 更新连接矩阵 D' , 记录下新的链路变化。

2. 如果相邻链路 l_a 断开, 则删除 C_a, T_a 向量; 如果增加了邻节点 a 和相邻链路, 则增加 $C_a: c_{a,j} = \infty$ ($j \in V, j \neq i$), $C_{a,a} = 0$; $T_a: t_{a,j} = 1$ ($j \in V, j \neq i$), $t_{a,a} = 0$ 。

3. 调用 Dijkstra 算法更新路由表 $\langle R', Next' \rangle$ 。

4. 根据 (2) 式更新 T_a 。

5. 进行环路检测。

(3) 接收到邻节点 a 的 NSA 消息:

(3.1) 如果 LSA 通知了新的链路变化时:

1. 更新连接矩阵 D' , 记录下新的链路变化。

2. 调用 Dijkstra 算法更新路由表 $\langle R', next' \rangle$ 。

3. 根据邻节点 a 的 NSA 消息中通知的 r_j^i , 更新 $c_{a,j} = r_j^i$ 。

4. 根据 R', C_a 更新 T_a 。

5. 进行环路检测。

(3.2) 如果 LSA 中通知的链路变化已经在所记录的范围中, 则不必重新计算路由表和产生新的 NSA 消息, 进行环路消除检测。

(4) 生成 NSA 消息, 并通知邻节点:

在 DSA 部分通知更新的 r_j^i , 在 LSA 部分通知所记录的新链路变化。

结束语 路由算法是 Ad hoc 网中最为重要的问题之一。本文主要对 Ad hoc 网中分布式路由算法进行了研究, 提出了一种新的分布式无环路由算法。新算法具有以下特点:

1) 利用相邻节点间建立的路由相关性信息, 使得只有满足无环约束条件的邻节点才能作为下一跳, 从而抑制了网络拓扑变化时暂态环路的形成。

2) 节点间的相关性信息通过交换网络状态通知 (NSA) 实现。NSA 消息包括了路由更新通知和链路更新通知两部分。节点根据对自己路由表的修改, 针对不同邻节点生成相应的网络状态通知。

3) 在所提出的路由算法中, 同时采用了距离向量和路径状态两种方法的结合。节点根据维护的连接矩阵, 使用路径状态算法计算出网络的路由表; 采用邻节点的距离向量信息确定节点之间的偏序关系, 据此进行环路检测和环路消除。

参考文献

- 1 Sanchez, et al. Networking on the Battlefield: Challenges in Highly Dynamic Multi-hop Wireless Networks. INFOCOM, 1999
- 2 Haas, et al. On Some Challenges and Design Choices in Ad-Hoc Communications. IEEE GLOBECOM, 1998. 187~192
- 3 史美林. 自组网路由协议综述. 通信学报, 2001, 11: 93~103
- 4 Royer. A Review of Current Routing Protocols for Ad Hoc Mobile Wireless Networks. IEEE Personal Communications, 1999, 17 (8): 47~55
- 5 Gracia-Lunes-Aveces. A Path-Finding Algorithm for Loop-Free Routing. IEEE Trans. on Networking, 1997, 5(1): 148~159
- 6 Vutukury. Distributed Algorithm for Multipath Computation. IEEE GLOBECOM, 1999
- 7 严蔚敏, 吴伟民. 数据结构. 清华大学出版社, 2000