

基于自组网的多路由冗余传输算法研究

刘宪磊^{1,2}, 韩军^{1,2}, 倪光华^{1,2}, 张春辉^{1,2}

(1. 中国电子科技集团公司第五十四研究所, 石家庄 050081;

2. 微波散射通信专业部, 石家庄 050081)

摘要: 随着“网络中心战”、“网电一体战”概念的提出, 自组网在组网通信中扮演着越来越重要的角色; 在实际应用中, 自组网常常面临着电磁频谱环境复杂, 无线设备侦听干扰频繁等问题, 对网络的抗干扰、抗截获性能提出了更高的要求; 针对上述问题, 提出多路由冗余传输算法, 源节点对原始数据进行冗余编码, 利用多路由协议发现多条路由并行传输编码包, 目的节点只需接收到一定数量的编码包即可恢复出全部原始信息; 该算法有效融合了多路由和冗余编码的特点, 具有抗干扰、抗截获和无需重传的优势; 利用 OPNET Modeler 仿真平台进行仿真建模, 测试了多路由冗余传输算法下通信节点对之间收发包数量以及监听节点截获数据包情况; 试验表明, 多路由冗余传输算法能有效提高网络的抗干扰、抗截获性能。

关键词: 自组网; 多径路由; 冗余编码; 抗干扰; 抗截获

Multipath Redundant Transmission Algorithm Research for MANET

Liu Xianlei^{1,2}, Han Jun^{1,2}, Ni Guanghua^{1,2}, Zhang Chunhui^{1,2}

(1. The 54th Research Institute of CETC, Shijiazhuang 050081, China;

2. Microwave and Troposcatter Communications Department, Shijiazhuang 050081, China)

Abstract: With the proposing of “Network Center Warfare” and “Integrated Electricity Network Warfare”, MANET plays an important role in network communication. The practical application environment of MANET is usually filled with complex electromagnetic spectrum, frequent hostile monitor and hostile interference, which resulting in higher requirements about the performance of anti-jamming and anti-interception. According to the above questions, the multipath redundant transmission algorithm is proposed: source node redundant codes the original data into coding packets, parallelly transports the coding packets by the paths calculated by the multipath routing protocol, and destination node can recover the original data by receiving a certain number of coding packets. Combining the multipath routing protocol with redundant coding, the algorithm proposed has the advantage of anti-jamming, anti-interference and without retransmission. Based on OPNET Modeler platform, the number of the packets between the communication nodes under the scenario of lost packet caused by interference and the packets intercepted by monitor node is tested. The test shows that the multipath redundant transmission algorithm can effectively improve the network performance of anti-jamming and anti-interception.

Keywords: MANET; multipath routing; redundant coding; anti-jamming; anti-interception

0 引言

随着“网络中心战”、“网电一体战”^[1]概念的提出, 自组网在组网通信中扮演着越来越重要的角色, 逐步成为当前数字化战场通信的首选装备, 是各国战术互联网的重要技术^[2]。

在早期通信过程中, 抗干扰多为设备级的抗干扰, 一般认为只要设备的抗干扰性能达到某种程度就自然解决了抗干扰问题。而在自组网应用环境中, 客观或人为的电磁干扰会直接影响传输路径的建立和数据分组的发送, 直观地导致丢包现象的出现, 并且干扰范围和强度的不规则性常常导致通信节点对之间路径传输受到限制甚至产生路径失败。网络层的入侵能够在底层设备检测正常的情况下造成传输路径中断、网络瘫痪, 进而无法满足上层业务对网络的需求, 具有很强的隐蔽性和破坏性。因此, 在传统物理层、链路层的抗干扰技术之外, 研究网络层相应的抗干扰措施^[3]具有十分重要的意义。

本文基于自组网实际应用中的抗干扰、抗截获需求, 提出

一种多路由冗余传输算法, 融合多径路由协议和冗余编码的特点, 源节点处对数据包进行冗余编码, 利用多路由发现的 多条传输路径并行传输编码包, 中间节点负责编码包的转发, 目的节点只需收到足够数量的编码包即可译码恢复出全部原始信息。

本文利用 OPNET Modeler 平台对实际网络环境进行仿真建模, 充分模拟通信过程中因电磁干扰而产生的丢包现象, 以及在无线空间对链路的侦听, 测试通信节点对之间的数据传输情况与侦听节点的数据包侦收数量。试验表明, 多路由冗余传输算法能有效降低干扰丢包带来的影响, 提高信息传输的可靠性, 增强网络的抗干扰和抗截获性能。

1 多路由冗余传输算法的设计

本文提出的多路由冗余传输算法首先利用 MP-OLSR (Multipath Optimized Link State Routing)^[4]路由协议计算得到源节点到目的节点的多条传输路径; 当有通信需求时, 源节点处对原始信息进行冗余编码, 再利用计算得到的多条传输路径并行传输编码包, 中间节点只负责数据包的转发; 由于采用了冗余编码, 目的节点只需要接收到足够数量的编码包即可译码恢复出全部信息, 无需接收全部编码包, 能有效抵抗干扰丢包给信息传输带来的影响; 另外, 由于采用了多径路由机制, 即

收稿日期: 2017-12-05; 修回日期: 2018-01-05。

基金项目: 国家自然科学基金(61504124)。

作者简介: 刘宪磊(1994-), 男, 硕士研究生, 主要从事计算机网络与无线自组网方向的研究。

使敌方在无线空间内的侦听到了某条链路的传输信息,也只能获得部分编码包,不足以恢复出完整的原始信息。多路由冗余传输算法框架图如图 1 所示。

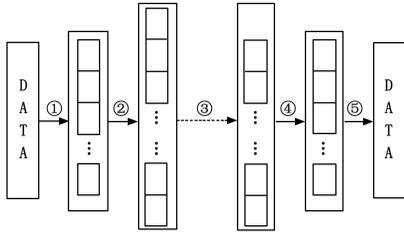


图 1 多路由冗余传输算法框架图

其中: ①源节点将原始信息分为 K 个原始数据包; ②对 K 个原始数据包进行冗余编码得到 N 个编码包; ③无线空间传输, 出现丢包; ④目的节点处根据接收到的数据包恢复出 K 个原始数据包; ⑤获取全部原始信息。

1.1 MP-OLSR 路由协议

目前的多径路由协议大多数是在典型单径路由协议的基础上针对数据传输可靠性、路由容错能力和拥塞控制等需求进行改进得到的。MP-OLSR 路由协议最早由易甲子等人提出^[5], 继承了 OLSR 协议适用于大规模网络的特点, 支持进行分簇设计。作为 OLSR 路由协议的扩展版本, MP-OLSR 协议保留了 OLSR 协议中基本的路由控制消息和拓扑信息处理机制, 核心机制在于采用多径 Dijkstra 算法计算传输路径^[5]。

多径 Dijkstra 算法:

多径 Dijkstra 算法的基本思想是多次迭代使用 Dijkstra 算法, 在每次迭代中发现一条源节点 s 到目的节点 d 的路径, 每次迭代之后增加已使用过的链路和节点的权值, 避免求出已有的路径。与标准的 Dijkstra 算法不同, 多径 Dijkstra 算法使用了两个成本函数以产生节点分离 (Node Disjoint) 或者链路分离^[3] (Link Disjoint) 的多条链路, 基本算法流程如下:

```

Multiple-Path Dijkstra(s,d,G,N)
C1 = C;
G1 = G;
for i = 1 to N do
  SourceTreei = Dijkstra(Gi,s);
  Pi = GetPath(SourceTreei,d);
  for all arcs e in E
    if e is in Pi or Reverse(e) is in Pi then
      Ci+1}(e) = fp(Ci}(e));
    else if the vertex Head(e) is in Pi then
      Ci+1}(e) = fe(Ci}(e));
    end if
  end for
  Gi+1} = (V,E,Ci+1});
End for
Return(P1,P2,...,PN);

```

图 2 多径 Dijkstra 算法流程

该算法基于图 $G = (V,E,C)$ 点 $(s,d) \in e^2$ 和一个正整数 N 。数组 (P_1, P_2, \dots, P_N) 是由标准 Dijkstra 算法得到的从点 s 到达点 d 的最短路径。GetPath (SourceTree, n) 从源树 SourceTree _{i} 中得到的到达节点 n 的最短路径; Reverse (e) 返回链路 e 的反向链路; Head (e) 返回链路 e 的两个顶点。当使用跳数 HopCount 作为权值时, C 取默认值 1^[6]。

f_p 和 f_e 是两个成本函数: f_p 增加上一条最短路径中的每一段链路的权值, f_e 增加上一条路径所经过的节点的权值,

二者共同作用决定下一条路径的计算。不同的 f_p 和 f_e 组合决定所得路径是链路相异还是节点相异。

多径 Dijkstra 算法的举例如图 3 所示, 其中设置成本函数 $f_p = 3c, f_e = 2c, c = 1$:

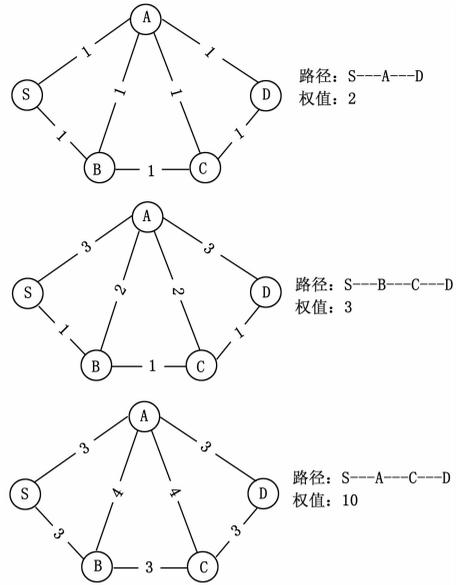


图 3 多径 Dijkstra 算法举例

- 1) 第一次计算时, 得到最短路径 $S \rightarrow A \rightarrow D$, 路径总权值为 2;
- 2) 第一次计算完成后, 成本函数 f_p 增加链路 $S \rightarrow A$ 与 $A \rightarrow D$ 的权值, f_e 增加链路 $A \rightarrow B$ 和 $A \rightarrow D$ 的权值;
- 3) 第二次计算时, 得到最短路径 $S \rightarrow B \rightarrow C \rightarrow D$, 路径总权值为 3;
- 4) 第二次计算完成后, 成本函数 f_p 增加链路 $S \rightarrow B$ 、 $S \rightarrow C$ 和 $C \rightarrow D$ 的权值, f_e 增加链路 $B \rightarrow A$ 和 $C \rightarrow A$ 的权值;
- 5) 第三次计算时, 得到最短路径 $S \rightarrow A \rightarrow C \rightarrow D$, 路径总权值为 10。

1.2 冗余编码

本文提出的多路由冗余传输算法中, 冗余编码负责将 K 个原始数据包编码为 N 个编码数据包 ($N > K$)。在此介绍喷泉码的基本概念^[6]: 该种编码可以将 K 个原始数据分组生成任意数量的编码分组, 接收端只要收到其中任意 M ($M > K$) 个编码分组即可译码以较高的概率成功恢复出全部原始分组。考虑到喷泉码与本文提出多路由冗余传输算法中冗余编码算法具有高度的契合性, 本文中即以喷泉码作为冗余编码算法进行分析研究。

LT (Luby Transform) 码是一种实用型喷泉码, 编译码方法简单, 编译码复杂度和译码开销较低, 编码冗余度可以根据需要设置, 灵活可控。LT 码编码分组的产生流程如下:

- 1) 根据设计的度函数分布 Ω 随机选取编码分组的度数 d ;
- 2) 从 K 个原始分组中等概率地随机选取 d 个;
- 3) 将这 d 个原始分组模二和, 生成一个编码分组;
- 4) 重复 (1) (2) (3) 步骤, 直到产生所需数量的编码分组。

LT 码性能的关键在于选择合理的度分布函数 Ω 以产生编码矩阵^[7]。目前公认的在仿真性能中比较好的两种度分布函数

分别为理想孤波分布 (Ideal Soliton Distribution, ISD) 和鲁棒孤波分布 (Robust Soliton Distribution, RSD)^[8]。

理想孤波分布: 定义 $\rho(d)$ 为:

$$\rho(1) = \frac{1}{K} \quad (1)$$

$$\rho(i) = \frac{1}{i(i-1)}, i = 2, \dots, K \quad (2)$$

其中:

$$\sum_{d=1}^K \rho(d) = 1 \quad (3)$$

鲁棒孤波分布:

$$\text{定义}(R = \theta \cdot \ln\left(\frac{K}{\epsilon}\right) \sqrt{K}, \theta > 0) \quad (4)$$

$$\tau(i) = \frac{R}{(iK)}, i = 1, \dots, \frac{K}{R} - 1 \quad (5)$$

$$\tau(i) = \frac{R \ln\left(\frac{R}{\epsilon}\right)}{K}, i = \frac{K}{R} \quad (6)$$

$$\tau(i) = 0, i = \frac{K}{R} + 1, \dots, K \quad (7)$$

把函数 $\tau(i)$ 与 $\rho(i)$ 相加并作归一化处理即可得到鲁棒孤波分布函数 $\mu(i)$:

$$\beta = \sum_i (\rho(i) + \tau(i)) \quad (8)$$

$$\mu(i) = \frac{(\rho(i) + \tau(i))}{\beta}, i = 1, 2, \dots, K \quad (9)$$

其中为接收端接收到 M 个编码数据分组时译码失败的概率; $M = K\beta$ 为以至少以 $1-\epsilon$ 的概率成功译码时所需要接收到的数据编码分组数量; θ 为小于 1 的任意常数。

当以鲁棒孤波分布作为 LT 码的编码度分布函数时, 能够以接近最少数量的编码数据分组恢复所有的原始数据分组。在 M. Luby 的经典理论^[9]中, 研究讨论了以鲁棒孤子分布作为度函数时, 若需要较好地还原信源信息, 接收端需要接收到 M 个已编码的信息数据包^[8]。基于此, 本文中即以鲁棒孤波分布作为 LT 码的编码度分布函数, 同时以目的节点接收到数据包数量来判定目的节点是否能够正确恢复全部原始信息。

2 仿真平台选择

目前主流的网络仿真软件主要有 OPNET、NS2、OMNET、Qualnet 等^[10]。OPNET 软件最初由麻省理工大学的两个博士创立, 作为一款商业软件其操作界面方便友好, 内置丰富的库函数和模型库, 既有各种基本模块以供调用, 也支持用户根据需要自定义模块。另外, OPNET 采用离散时间驱动的模拟机制, 较之于时间驱动, 计算效率有很大的提高。

本文中提出的多路由冗余传输算法拟基于 OPNET 软件系列中的 OPNET Modeler 平台进行建模分析。

3 模型建立

3.1 网络域模型

本文中网络模型及主要网络参数配置如图所示如图 4 和表 1 所示。

3.2 节点域模型

各节点的节点域模型设计如图 5 所示, 主要包括应用层模块 (src、sink), 序列号分配模块 (assign), 数据包编码模块 (code_buffer), 路由控制层 (route), 数据链路层模块 (mac_

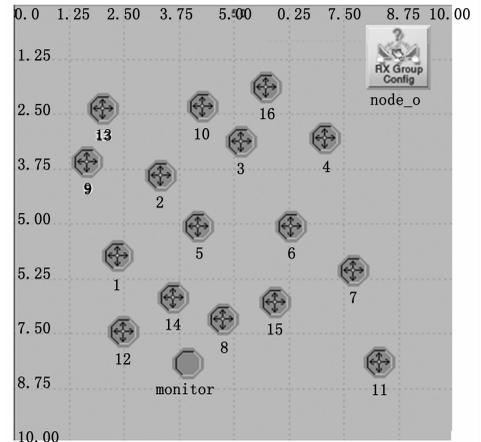


图 4 网络域模型

表 1 网络主要参数配置

参数	参数值
网络区域	10 km × 10 km
节点数	16
源节点	1
目的节点	7
监听节点	monitor
编码参数 θ	0.03
编码参数 ϵ	0.01
路由协议	OLSR、改进 MP-OLSR
路由条目	3
单条路由最大跳数	5
成本函数 f_c	$f_c(c) = 2c$
成本函数 f_p	$f_p(c) = 3c$
数据包速率	1 包/s
数据包大小	512 Bytes
比特率	11 Mbps
MAC 层协议	TDMA
网络丢包率	3.00%

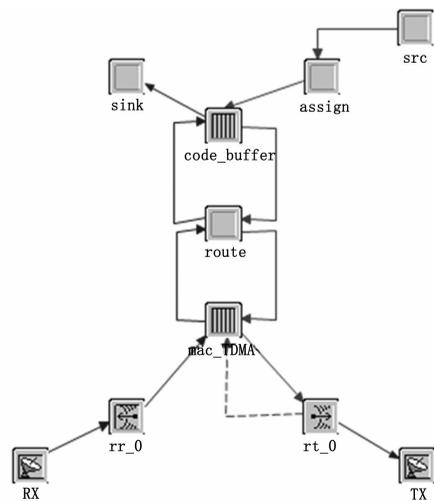


图 5 节点域模型

TDMA), 和物理层模块 (rr_0、rt_0、RX、TX), 其中:

- 1) src、sink: 模拟通信过程中的上层业务;

2) assign: 对 src 产生的数据包封装序列号, 以及添加控制域字段;

3) code_buffer: 对原始数据包进行冗余编码;

4) route: 路由由协议控制报文处理模块, 负责计算数据分发路径;

5) mac_TDMA: 数据链路层协议模块, 采用 TDMA 协议;

6) rr_0、rt_0、RX、TX: 无线收发信机组及收发天线, 实现信号的接收和发送。

3.3 数据包编码冗余度分析

引入冗余度参数 σ :

$$\sigma = \frac{N}{K} - 1 \quad (10)$$

其中: N 为编码生成包数量, K 为原始数据包数量。考虑到丢包数量与实际网络规模和原始数据包数量有关, 试验中统计了图 4 所示的 16 节点网络中采用多路由冗余传输算法时源节点发送不同原始数据包数量对应目的节点接收到的数据包数量, 如图 6 所示。

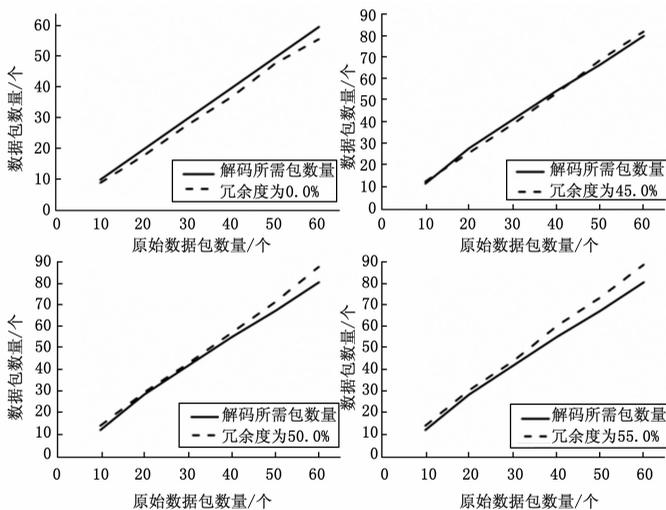


图 6 冗余度与收包数量关系

由图 6 可知, 当冗余度 $\sigma=0$ 即不采用冗余编码时, 目的节点接收到的数量小于解码所需包的数量, 即由于干扰丢包现象的存在, 目的节点无法恢复全部原始数据。当冗余度 σ 分别为 45.0%、50.0%、55.0% 时, 对比可知: 当 $\sigma=45.0\%$ 时, 只有当原始数据包数量较大时, 才能保证接收包的数量大于解码所需的数据包数量, 不具有稳定的解码恢复数据能力; 当 $\sigma=50.0\%$ 或 $\sigma=55.5\%$ 时, 对于不同数量的原始包, 目的节点接收包数量均大于解码所需的数据包数量, 即目的节点能够完整地恢复出全部原始信息。在下文多路由冗余传输算法仿真过程中的冗余度 σ 即设置为 50.0%, 此时原始数据包数量与解码所需数据包数量对应关系如表 2 所示。

表 2 解码所需编码包与原始数据包数量对应关系

数据包数量(个)	值					
原始数据包 K	10	20	30	40	50	60
编码生成包 N	15	30	45	60	75	90
解码所需编码包 M	12	28	41	54	66	79

4 仿真结果分析

在仿真过程中, 设置原始数据包数量分别为 $K=20$ 和 $K=50$, 统计对比了传统单径 OLSR 协议与多路由冗余传输算法下通信节点之间收发包数量与监听节点侦听包数量情况。

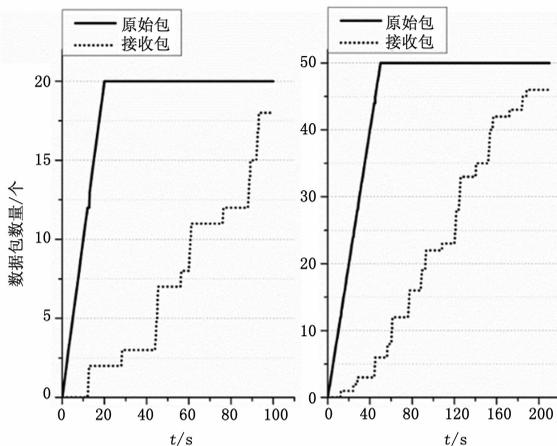


图 7 OLSR 协议中收发包情况

由图 7 可知, 采用传统单径 OLSR 协议情况下, 当原始数据包数量 $K=20$ 和 $K=50$ 时, 目的节点接收到的数据包数量分别为 18 包和 48 包, 两种情形下因干扰产生的丢包数为 2 包。由于源节点不采用任何冗余编码措施直接发送原始数据包, 目的节点必须接收到全部数据包才能正确恢复全部原始信息。因此由于丢包现象的存在, 目的节点不能完全恢复出全部原始信息。

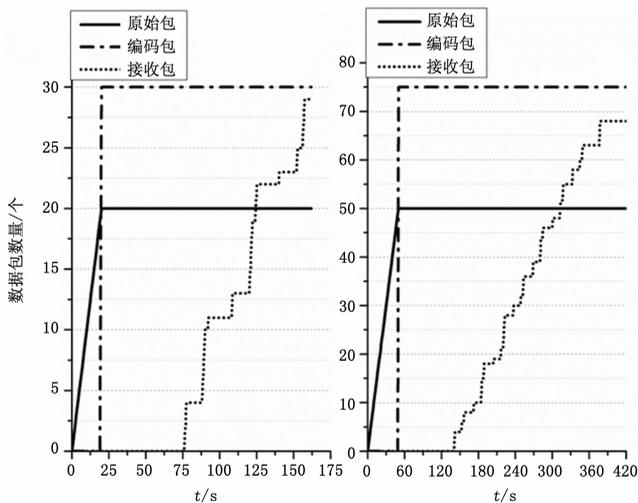


图 8 多路由冗余传输算法中收发包情况

由图 8 可知, 采用多路由冗余传输算法情况下, 当原始数据包数量 $K=20$ 和 $K=50$ 时, 编码包数量分别为 $N=30$ 和 $N=75$ ($\sigma=50.0\%$), 目的节点接收到的数据包数量分别为 29 包和 69 包, 产生的丢包数分别为 1 包和 6 包。由于当所有原始数据包产生完毕后才开始编码包的产生与发送, 因此编码包发送时间略滞后于原始数据包的产生。由表 2 可知, 即使存在丢包现象, 目的节点仍能够收到足够的编码包恢复出原始信息, 验证了多路由冗余传输算法的抗干扰性能。

(下转第 199 页)