



(12)发明专利

(10)授权公告号 CN 103475453 B

(45)授权公告日 2017.03.08

(21)申请号 201310420089.2

(22)申请日 2013.09.02

(65)同一申请的已公布的文献号

申请公布号 CN 103475453 A

(43)申请公布日 2013.12.25

(73)专利权人 中山大学

地址 510275 广东省广州市海珠区新港西路135号

(72)发明人 刘星成 龚信仁

(74)专利代理机构 广州粤高专利商标代理有限公司

44102

代理人 邱奕才

(51)Int.Cl.

H04L 1/00(2006.01)

H04W 84/18(2009.01)

(56)对比文件

CN 102164369 A,2011.08.24,

US 2009201899 A1,2009.08.13,

罗建中.无线传感器网络中基于网络编码的协作通信研究.《中国优秀硕士学位论文全文数据库信息科技辑》.2012,

Xiaohua (Edward) Li,et.al.Application of STBC-Encoded Cooperative Transmissions in Wireless Sensor Networks.《IEEE SIGNAL PROCESSING LETTERS》.2005,第12卷(第2期),

李莎等.基于网络的无线传感器网络节能路由算法.《计算机工程》.2011,第37卷(第9期),

Xingcheng Liu,et.al.A novel cluster-chain channel adaptive routing protocol in wireless sensor networks.《QShine'08 Proceedings of the 5th International ICST Conference on Heterogeneous Networking for Quality》.2008,

审查员 舒思

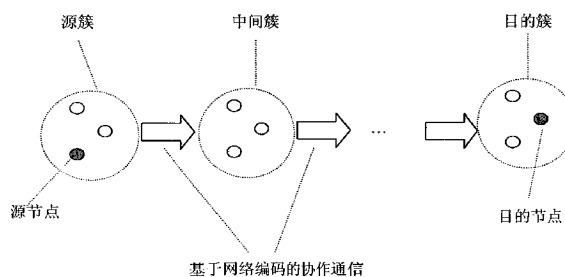
权利要求书3页 说明书5页 附图7页

(54)发明名称

基于随机线性网络编码的协作通信方法

(57)摘要

本发明针对无线传感器网络(Wireless Sensor Networks,WSNs)通信的可靠性,提出了基于网络编码的协作通信方法(Network Coding Based Cooperative Communication Scheme,NCCC)。NCCC结合了协作通信和网络编码各自的优点,通过对发送数据包进行网络编码而获得一定的编码正增益,同时利用协作通信降低因个别节点失效而造成通信失败的概率。在NCCC中,每个节点独立编码并发送数据包,不需要几个节点同时发送数据,因此节点间不需要严格的时钟同步。同时,在有限域上随机产生网络编码的编码向量,从而降低了实现的复杂度。本发明具有实现复杂度低,并可以大大提高WSN中的通信可靠性的特点。



1. 一种基于随机线性网络编码的协作通信方法,其特征在于:由若干个节点组成簇协作传输数据包,在发送数据包前,每个正常工作的节点均向簇内其他节点广播自己的状态及收到的数据包个数,同时接收其他节点的广播信息并将节点号及其接收到的数据包数量存储在一个发送队列TransQueue里,之后对TransQueue队列按节点接收到的数据包个数从大到小按顺序进行排序,使得簇内每个正常工作的节点均可知道簇内其他正常工作节点的状态及其接收到的数据包数量,当有节点不能正常工作时,其他正常工作的节点会接替失效节点的发送任务;

在每轮发送数据过程中,每个簇均需发送N个编码数据包,中间簇的节点根据TransQueue里自己所在的位置确定自己的发送时隙,并在特定的时隙对收到的数据包进行随机网络编码,之后再发送,若队列TransQueue的长度为 $n=N$,则簇内N个节点的每个节点均按TransQueue里的顺序各发送一个编码数据包;若 $n<N$,则簇内n个节点的每个节点按TransQueue里的顺序各发送一个编码数据包,然后按TransQueue里的第1个节点发送第n+1个编码数据包,第2个节点发送第n+2个数据包,以此类推,直到发送N个编码数据包结束,其中, $n\leq N$;

三种处理:源簇的处理、中间簇的处理、目的簇数据处理,及其相协调的处理过程;

源簇的处理,分为三个阶段:源节点数据包广播阶段、簇内各节点状态广播及发送队列构建阶段和簇内各节点编码和数据发送阶段,具体处理的实现如下:

阶段一:源节点数据包广播阶段——源节点将需要发送给目的节点的源数据包通过广播的方式发送给簇内其他节点;

阶段二:簇内各节点状态广播及发送队列构建阶段——源簇内所有正常工作的节点在特定的时隙进行状态广播通告其他节点自己工作状态,在其他时隙则监听并接收其他节点的状态广播;根据收到的状态通告将发出状态广播的节点的节点号依次存入TransQueue队列中,轮到自己广播时也同时将自己的节点号存入TransQueue;

阶段三:簇内各节点编码和数据发送阶段——假设源数据包长度可以进行M等分,源簇内正常工作的节点将源数据包分成M个等长的数据段(P_1, P_2, \dots, P_M);每个簇发送数据至下一个簇时,都需要进行N次发送,当发送簇正常工作的节点个数n小于N时,由于TransQueue是循环队列,在TransQueue的n个节点按先后顺序发送完后,发送队列又回到TransQueue的开头进行剩下时隙的发送任务,不断循环直至发送完N个数据包;每个节点在一轮传输过程中可能不止发送一次,因此每个节点需要计算本节点在数据传输需要传输的次数及确定传输的时隙,具体做法如下:

令 N/n 的商为Q,余数为r,本节点在TransQueue中所处的位置为I,则在本次传输过程中本节点需要进行的传输次数为:

$$c = Q + \left\lfloor \frac{r}{I} \right\rfloor, \text{其中符号} \left\lfloor \cdot \right\rfloor \text{表示对“} \cdot \text{”下取整的操作运算;}$$

即本节点需要分别第I, n+I, $\dots, (c-1) \times n + I$ 个数据时隙将数据段(P_1, P_2, \dots, P_M)进行编码发送,每一次都要进行一次编码封包后发送,簇内总共分为N个发送时隙,每个时隙发送一个编码数据包,簇内所有节点总共要发出N个编码数据包;

编码方法如下:先在 $GF(q)$ 域上随机选取域元素 v_i ,组成一个 $M \times 1$ 维的列向量 $v = (v_1, v_2, \dots, v_M)^T$,将(P_1, P_2, \dots, P_M)和v进行 $GF(q)$ 域上的相乘运算,得到编码后的数据C:

$$C=(P_1 P_2 \cdots P_M) \cdot v$$

将得到的C及v进行封包发送。

2. 根据权利要求1所述的基于随机线性网络编码的协作通信方法,其特征在于中间簇的处理过程,分为三个阶段:数据接收阶段、状态广播阶段和数据包再编码及数据发送阶段,具体的实现如下:

阶段一:数据接收阶段——每个节点维护一个数据接收列表RecvTable和接收数据包数量J;当节点从上一跳簇收到一个数据包时,将数据包存入RecvTable,并使 $J=J+1$;

阶段二:状态广播阶段——第一阶段的数据接收结束之后,簇内所正常工作的节点在特定的广播时隙广播自己的接收数据包的情况,同时在其他节点的广播时隙监听并接收其他节点广播,广播数据包中包含自己接收到的数据包的数量;本阶段的处理流程具体如下:

在本次的状态广播中,每个节点除了通告各自的工作状态外还通告自己接收到的数据包数量J,簇内所有在本轮正常工作的节点通过接收簇内其他节点发送的状态广播来获知其他节点接收数据包的情况;每个节点以收集到的各个节点的接收到的数据包的数量J从大到小的顺序来对TransQueue进行排序,J值相等的节点则以节点号从小到大的顺序进行排序;

阶段三:数据包再编码及数据发送阶段——设节点收到的m个数据包: $P_R=(P_{R1}, P_{R2}, \cdots, P_{Rm})$,其中 $m \leq N$,对 P_R 进行一次再编码过程如下:

先从 $GF(q)$ 域中随机选取m个域元素 v_{Ri} 组成列向量 $v_R=(v_{R1}, v_{R2}, \cdots, v_{Rm})^T$,对 P_R 所有数据包中v和C进行线性组合组成新的 v_T 和 C_T ,再封装成新的数据包进行发送;数据发送过程为:根据本节点在TransQueue中所处的位置计算本节点在本轮传输中需要进行数据发送的次数c并确定自己的发送时隙,在自己的发送时隙进行数据发送。

3. 根据权利要求2所述的基于随机线性网络编码的协作通信方法,其特征在于目的簇数据处理,分为四个阶段:数据接收阶段、状态广播阶段、数据包汇集阶段和目的节点解码阶段,具体实现如下:

阶段一:数据接收阶段——本阶段与中间簇数据处理过程的数据接收阶段相同;

阶段二:状态广播阶段——本阶段与源簇数据处理过程中的簇内各节点状态广播阶段相同;

阶段三:数据包汇集阶段——在本阶段,目的簇中除目的节点外,其他正常工作的节点均将接收到的数据包转发给目的节点,目的节点将收集到的数据包与自己接收的RecvTable中数据包的序列号进行对比,若有相同,说明收集到的数据包已存在目的节点中,则丢弃所述收集到的数据包;

阶段四:目的节点解码阶段——当目的节点收集簇内其他节点发来的所有数据包并丢弃重复数据包后,就开始对收到的数据包进行解码;设收到的数据包的个数为m,所有数据包中的编码向量的集合为 $V_D=(V_{D1}, V_{D2}, \cdots, V_{Dm})$,其中 V_{Di} 是第i个数据包携带的数据向量,数据段集合为 $C_R=(C_1, C_2, \cdots, C_m)$,则 V_D 、 C_R 和原始数据段 $P=(P_1, P_2, \cdots, P_M)$ 的关系有:

$$P \cdot V_D = C_R$$

设 $V_{Di}=(v_{Di1}, v_{Di2}, \cdots, v_{DiM})^T$,上式可写成

$$(P_1 \ P_2 \ \cdots \ P_M) \cdot \begin{pmatrix} v_{D11} & v_{D21} & \cdots & v_{Dm1} \\ v_{D12} & v_{D22} & \cdots & v_{Dm2} \\ \vdots & \vdots & \ddots & \vdots \\ v_{D1M} & v_{D2M} & \cdots & v_{DmM} \end{pmatrix} = (C_1 \ C_2 \ \cdots \ C_m)$$

转换成线性方程组形式,得

$$\begin{cases} v_{D11}P_1 + v_{D12}P_2 + \cdots + v_{D1M}P_M = C_1 \\ v_{D21}P_1 + v_{D22}P_2 + \cdots + v_{D2M}P_M = C_2 \\ \vdots \\ v_{Dm1}P_1 + v_{Dm2}P_2 + \cdots + v_{DmM}P_M = C_m \end{cases}$$

由以上方程组可知,当 $m < M$ 时,方程组有多解,则不能解出原始信息 P ;当 $m \geq M$ 时,若 $\text{rank}(V_D) = M$,则方程组有唯一解;此时可通过高斯消元法解出原始信息 P 。

基于随机线性网络编码的协作通信方法

技术领域

[0001] 本发明针对的是无线传感器网络(Wireless Sensor Networks, WSNs)及其协作通信方法,具体地说,是涉及到无线传感器网络中的网络编码和协作通信方法及其可靠性等性能。

背景技术

[0002] 无线传感器网络是近十年来兴起的一项网络技术,它集嵌入式系统、无线通信技术、计算机网络技术等高新技术于一身,有着广阔的应用前景,因此在学术界吸引了世界各国高校及研究机构的目光;在工业界也逐步得到工业巨头们的重视,目前已有无线传感器网络产品被广泛应用于军事、工业、交通、农业、医疗等多个领域。

[0003] 无线传感器网络由大量分布于监测区域的传感器节点组成,这些传感器节点通过无线自组织的方式构成网络。传感器节点通过安装在其上面的各类传感器(如温度传感器、湿度传感器、光照传感器等)探测相应的环境参数数据后,经过本地初步处理之后通过网络路由协议找出到达汇聚节点的路径,然后将数据发送至下一跳节点。下一跳节点经过存储转发的方式(或将数据与本地数据进行数据融合后)将数据发送到再下一跳节点,直至将数据传送到汇聚节点。汇聚节点将收到的数据进行汇总处理,然后通过公共通信网络,如卫星、互联网和移动通信网络等途径将数据发送至管理终端。

[0004] 由于监控区域一般位于环境恶劣的地方,不便或不可能提供持续性电源或持续性电源的提供机制复杂有待解决,因此,传感器节点一般采用电池供电。另外,由于成本和体积的限制,传感器节点的硬件配置很低,数据处理和存储能力、无线通信能力有限,通信可靠性较低。由此可以看出,一些在计算机网络和无线自组网(Ad-hoc Network)中常用的通信协议并不能直接用在无线传感器网络中。为了提高WSN中的通信可靠性,很有必要设计适用于无线传感器网络的简单高效通信协议。

发明内容

[0005] 本发明设计了一种可靠的基于随机网络编码的协作通信方法(Network Coding Based Cooperative Communication Scheme, NCCC),以较低的实现复杂度提高无线传感器网络中的通信可靠性。为了达到以上目的,本发明采用的技术方案如下:

[0006] NCCC协作通信是基于簇的传输方法。该NCCC方法充分利用了无线通信的广播特性,通过一发多收的方式降低因单条无线链路问题而导致通信失败的概率。同时,在每次进行数据发送前每个正常工作的节点进行状态广播,这样可以使得每个正常工作的节点掌握其他节点的工作状态。当存在节点失效时,符合条件的节点自动充当失效节点,代为完成发送任务,以降低节点失效对通信可靠性的影响。

[0007] 节点对数据包进行线性网络编码,每次只发送编码后的数据包,接收端只需接收到足够的符合解码条件的数据包即可解码出源数据包,而不需要接收所有数据包。因此,在有个别通信链路或节点失败的情况下仍可通信成功,提高了通信的可靠性。在进行线性网

络编码时,采用随机网络编码的方式,即在有限域上随机产生编码向量并对数据进行线性组合。随机网络编码本身可以大大降低编码时的实现复杂度,使其易于在资源有限的传感器节点上实现。但是,随机产生的编码向量存在一定的概率(记为 $p_{N,M}$)不符合解码条件,从而使得通信误包率(Packet Error Rate,PER)存在一个 $p_{N,M}$ 的下限值,当 M 、 N 值满足一定条件时, $p_{N,M}$ 的值可以做到非常小。

附图说明

- [0008] 图1系统模型
- [0009] 图2源簇状态广播阶段节点的处理流程
- [0010] 图3源簇编码发送处理流程
- [0011] 图4封包格式
- [0012] 图5中间簇状态广播节点的处理流程
- [0013] 图6中间簇节点再编码过程
- [0014] 图7中间簇数据再编码发送流程
- [0015] 图8单跳情况下各种方法(或算法)的性能对比
- [0016] 图9各种方法的PER在不同跳数情况下的性能比较
- [0017] 图10多跳情况下各种方法的时延性能对比

具体实施方式

[0018] NCCC协作通信所用的数据处理方法主要包括源节点所在簇(源簇)的处理过程、中间簇的处理方法以及目的节点所在簇(目的簇)的处理过程。下面结合附图对本发明进一步说明。

[0019] 1.源簇的处理方法分为三个阶段:源节点数据包广播阶段、簇内各节点状态广播及发送队列构建阶段和簇内各节点编码和数据发送阶段。

[0020] 阶段一:源节点数据包广播阶段。

[0021] 源节点将需要发送给目的节点的源数据包通过广播的方式发送给簇内其他节点。

[0022] 阶段二:簇内各节点状态广播及发送队列构建阶段。

[0023] 源簇内所有正常工作的节点在特定的时隙进行状态广播通告其他节点自己工作状态,在其他时隙则监听并接收其他节点的状态广播。根据收到的状态通告将发出状态广播的节点的节点号依次存入TransQueue队列中,轮到自己广播时也同时将自己的节点号存入TransQueue。本阶段的流程图如图2所示,其中 t 表示当前时隙, n 用于记录发送广播的节点个数(即在本轮本簇内正常工作的节点个数, $n \leq N$), I 用于记录本节点在TransQueue中所处的位置。

[0024] 阶段三:簇内各节点编码和数据发送阶段。

[0025] 假设源数据包长度可以进行 M 等分,源簇内正常工作的节点(包括源节点)将源数据包分成 M 个等长的数据段(P_1, P_2, \dots, P_M)。每个簇发送数据至下一个簇时,都需要进行 N 次发送,当发送簇正常工作的节点个数 n 小于 N 时,由于TransQueue是循环队列,在TransQueue的 n 个节点按先后顺序发送完后,发送队列又回到TransQueue的开头进行剩下时隙的发送任务,不断循环直至发送 N 个数据包。每个节点在一轮传输过程中可能不止发送一次,因此

每个节点需要计算本节点在数据传输需要传输的次数及确定传输的时隙,具体做法如下:

[0026] 令 N/n 的商为 Q ,余数为 r ,本节点在TransQueue中所处的位置为 I ,则在本次传输过程中本节点需要进行的传输次数为:

[0027] $c = Q + \left\lfloor \frac{r}{I} \right\rfloor$ (其中符号 $\lfloor \cdot \rfloor$ 表示对“ \cdot ”下取整的操作运算)

[0028] 即本节点需要分别在第 $I, n+I, \dots, (c-1) \times n + I$ 个数据时隙将数据段 (P_1, P_2, \dots, P_M) 进行编码发送,每一次都要进行一次编码封包后发送,簇内总共分为 N 个发送时隙,每个时隙发送一个编码数据包,簇内所有节点总共要发出 N 个编码数据包。

[0029] 编码方法如下:先在 $GF(q)$ 域上随机域元素 v_i ,组成一个 $M \times 1$ 维的列向量 $v = (v_1, v_2, \dots, v_M)^T$,将 (P_1, P_2, \dots, P_M) 和 v 进行 $GF(q)$ 域上的相乘运算,得到编码后的数据 C :

[0030] $C = (P_1 \ P_2 \ \dots \ P_M) \cdot v$

[0031] 将得到的 C 及 v 进行封包发送,本阶段流程图如图3所示,数据封包格式如图4所示。

[0032] 2. 中间簇的处理过程分为三个阶段:数据接收阶段、状态广播阶段和数据包再编码及数据发送阶段。

[0033] 阶段一:数据接收阶段。

[0034] 每个节点维护一个数据接收列表RecvTable和接收数据包数量 J 。当节点从上一跳簇收到一个数据包时,将数据包存入RecvTable,并使 $J = J + 1$ 。

[0035] 阶段二:状态广播阶段。

[0036] 第一阶段的数据接收结束之后,簇内所正常工作的节点在特定的广播时隙广播自己的接收数据包的情况,同时在其他节点的广播时隙监听并接收其他节点广播,广播数据包中包含自己接收到的数据包的数量。本阶段的处理流程与源簇中阶段二中的状态广播处理流程类似,只是多一个TransQueue的排序过程。

[0037] 在本次的状态广播中,每个节点除了通告自己的工作状态外还通告自己接收到的数据包数量 J ,簇内所有在本轮正常工作的节点通过接收簇内其他节点发送的状态广播来获知其他节点接收数据包的情况。每个节点以收集到的各个节点的接收到的数据包的数量 J 从大到小的顺序来对TransQueue进行排序, J 值相等的节点则以节点号从小到大的顺序进行排序。本阶段流程图如图5所示。

[0038] 阶段三:数据包再编码及数据发送阶段。

[0039] 设节点收到的 $m (m \leq N)$ 个数据包: $P_R = (P_{R1}, P_{R2}, \dots, P_{Rm})$,对 P_R 进行一次再编码过程如下:

[0040] 先从 $GF(q)$ 域中随机选取 m 个域元素 v_{Ri} 组成列向量 $v_R = (v_{R1}, v_{R2}, \dots, v_{Rm})^T$,对 P_R 所有数据包中 v 和 C 进行线性组合组成新的 v_T 和 C_T ,再封装成新的数据包进行发送,如图6所示。

[0041] 数据发送过程与源簇数据处理过程中阶段三的数据发送过程相似:根据本节点在TransQueue中所处的位置计算本节点在本轮传输中需要进行数据发送的次数 c 并确定自己的发送时隙,在自己的发送时隙进行数据发送,如图7所示。

[0042] 3. 目的簇数据处理过程分为四个阶段:数据接收阶段、状态广播阶段、数据包汇集阶段和目的节点解码阶段。

[0043] 阶段一:数据接收阶段。

[0044] 本阶段与中间簇数据处理过程中的数据接收阶段同。

[0045] 阶段二:状态广播阶段。

[0046] 本阶段与源簇数据处理过程中的簇内各节点状态广播阶段相同。

[0047] 阶段三:数据包汇集阶段。

[0048] 在本阶段,目的簇中除目的节点外,其他正常工作的节点均将接收到的数据包转发给目的节点,目的节点将收集到的数据包与自己接收的RecvTable中数据包的序列号进行对比,若有相同,说明收集到的数据包已存在目的节点中,则丢弃所述收集到的数据包。

[0049] 阶段四:目的节点解码阶段。

[0050] 当目的节点收集簇内其他节点发来的所有数据包并丢弃重复数据包后,就开始对收到的数据包进行解码。设收到的数据包的个数为 m ,所有数据包中的编码向量的集合为 $V_D = (V_{D1}, V_{D2}, \dots, V_{Dm})$,其中 V_{Di} 是第 i 个数据包携带的数据向量,数据段集合为 $C_R = (C_1, C_2, \dots, C_m)$,则 V_R 、 C_R 和原始数据段 $P = (P_1, P_2, \dots, P_M)$ 的关系有:

$$[0051] \quad P \cdot V_D = C_R$$

[0052] 设 $V_{Di} = (v_{Di1}, v_{Di2}, \dots, v_{DiM})^T$,上式可写成

$$[0053] \quad (P_1 \quad P_2 \quad \dots \quad P_M) \cdot \begin{pmatrix} v_{D11} & v_{D21} & \dots & v_{Dm1} \\ v_{D12} & v_{D22} & \dots & v_{Dm2} \\ \vdots & \vdots & \ddots & \vdots \\ v_{D1M} & v_{D2M} & \dots & v_{DmM} \end{pmatrix} = (C_1 \quad C_2 \quad \dots \quad C_m)$$

[0054] 转换成线性方程组形式,得

$$[0055] \quad \begin{cases} v_{D11}P_1 + v_{D12}P_2 + \dots + v_{D1M}P_M = C_1 \\ v_{D21}P_1 + v_{D22}P_2 + \dots + v_{D2M}P_M = C_2 \\ \vdots \\ v_{Dm1}P_1 + v_{Dm2}P_2 + \dots + v_{DmM}P_M = C_m \end{cases}$$

[0056] 由以上方程组可知,当 $m < M$ 时,方程组有多解,则不能解出原始信息 P ;当 $m \geq M$ 时,若 $\text{rank}(V_D) = M$,则方程组有唯一解。此时可通过高斯消元法解出原始信息 P 。

[0057] 各种方法(NCCC、MRNC、RDLC、MIMO、SISO)在单跳传输过程中的可靠性能如图8所示,从图中可以看出,在同等能量消耗的条件下,RDLC的性能最优但RDLC的实现复杂度较高,在各类资源都极度缺乏的无线传感器网络中应用比较困难。NCCC的可靠性能均比SISO及MRNC的性能要好,当 $N > M$ 时,在低能耗阶段,NCCC的性能略优于协作MIMO。

[0058] 多跳传输中,各种方法的误包性能仿真结果如图9所示,各种方法的PER均随着跳数的增加而呈上升趋势。在相同条件下,NCCC的PER上升速度均比MRNC和SISO要低,原因如下:MRNC通过冗余的路径向目的节点发送经过网络编码后的数据包,每条路径都是独立传输,而且采用的是单点对单点的无线传输。若一条独立路径在传输过程的任何一跳的传输失败,则整条路径传输失败,因此随着传输路径变长(即跳数变大),MRNC的传输失败的概率就增大。RDLC的上升速度最快,RDLC采用旧数据包和新数据包一起编码发送的方式以不增加发送时延而提高可靠性,但也会造成因某一次传输的失败而使得所有一起编码的数据包不能被正确解码,从而出现大量数据传输失败。当传输路径较长时,数据包在传输路径上传输失败的概率就增大,会造成RDLC中大量数据包不能被正确解码的情况,从而其PER也会急剧上升。而本文提出的NCCC方法充分利用无线通信的广播特性,通过一个节点发送多个节

点同时接收的方式,降低了单点对单点通信失败的概率,从而提高了通信的成功率。

[0059] 多跳情况下,各种方法的时延性能如图10所示。从图中可以看出,SISO、RDLC及MRNC的时延相对比较小,因为它们都是纯粹单点对单的通信,没有协作通信中节点间的协作和时隙的分配而产生的时延。协作MIMO在簇间协作传输过程中因所有节点都是在同一频率同时发送,此部份的时延与SISO一样。但是协作MIMO在每一跳的传输过程中每个簇都有一个簇首,发送簇中簇首需要将数据分发给其他协作节点进行协作传输,接收簇中除簇首外其他节点需要将接收到数据汇集给簇首,数据分发阶段簇首只需一个广播时隙就可以完成,但数据汇集阶段则需要分为 $N-1$ 个时隙,因此其相对于SISO需要额外的时延进行数据分发和汇集,且随着 N 的增大而增大。由于在每一跳的过程中都需要进行数据分发,随着跳数的增加,其因数据分发和数据汇集产生的时延也将线性增大。对于NCCC,在源簇源节点需要进行数据的广播分发及在目的簇各节点将收到的数据包汇集给目的节点,在每个簇内有一次状态广播的过程且在数据包传输时还有一部份的报头开销(编码向量),因此NCCC的时延比SISO、RDLC和MRNC都要大,且随着跳数增大而增大。相比于协作MIMO,因NCCC在每一跳都需要进行各节点的状态广播,在跳数较小(如跳数为1或2)时,时延比协作MIMO要稍微大一些。但由于NCCC在中间簇并不需要进行数据的数据分发和汇集,比协作MIMO节省了部份的时延开销,随着跳数的增大节省的时延开销也增大,从图中可以看出,跳数越大,NCCC比协作MIMO节省的时延越多,总时延就越小。另外,NCCC的编码效率 $R(R=M/N)$ 越小,说明发送的数据包冗余信息越多,则需要更多的时间来进行传输,因此时延就越大。

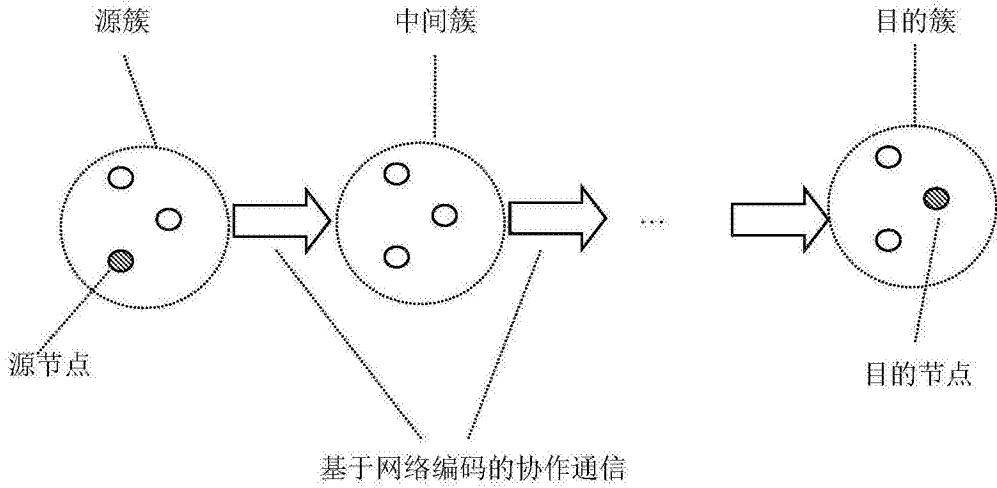


图1

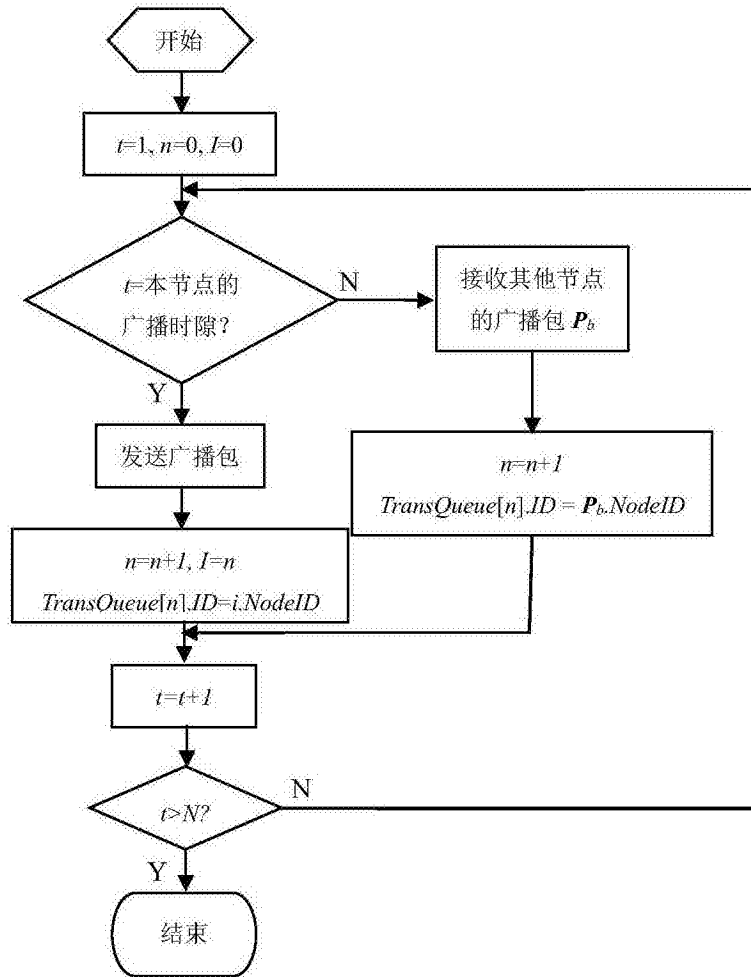


图2

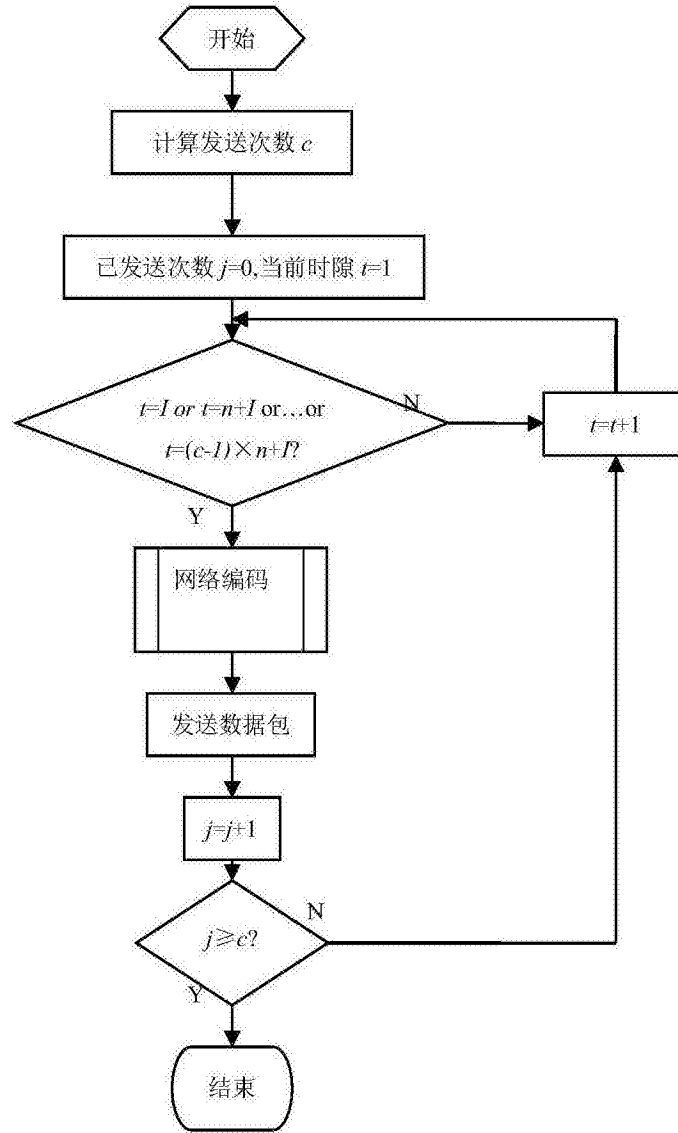


图3

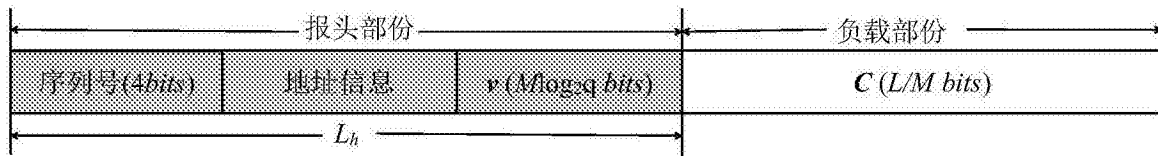


图4

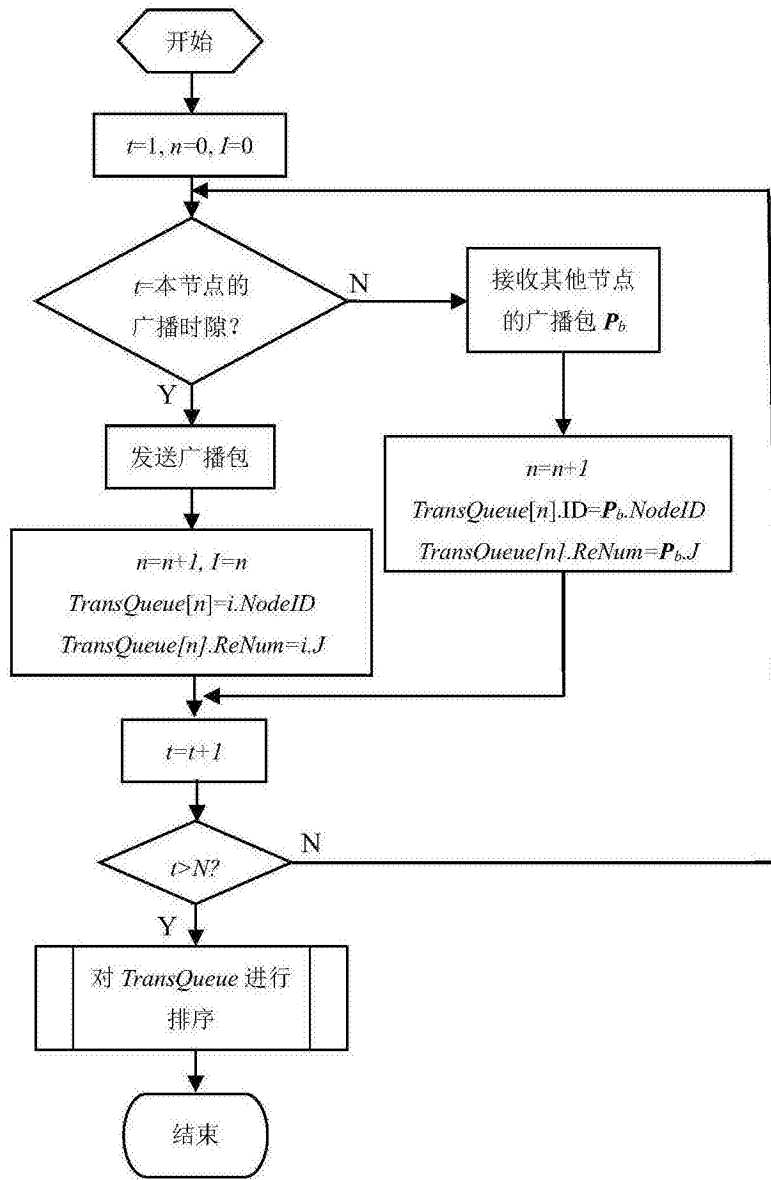


图5

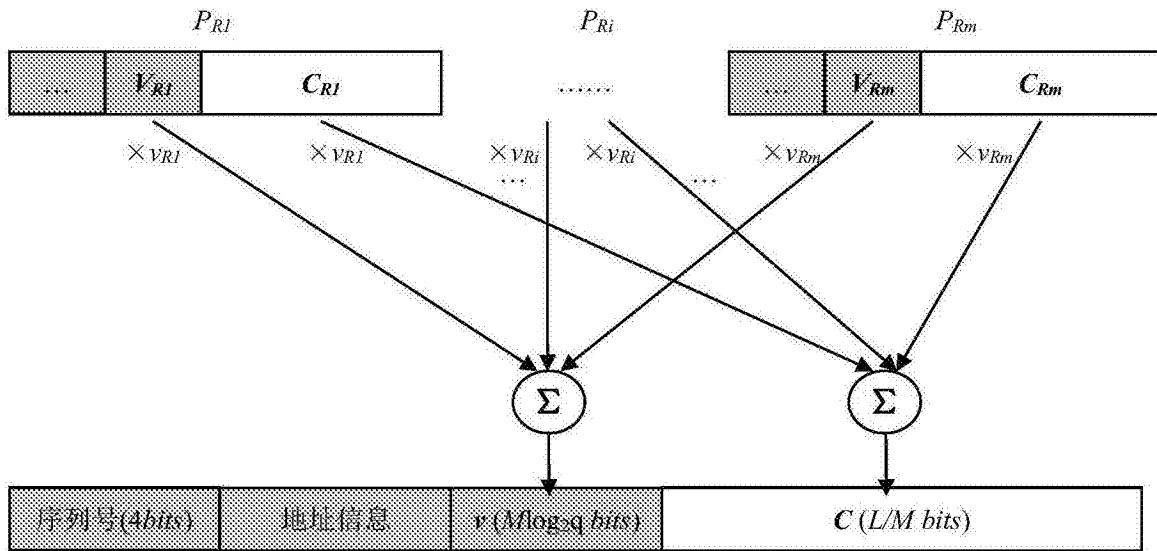


图6

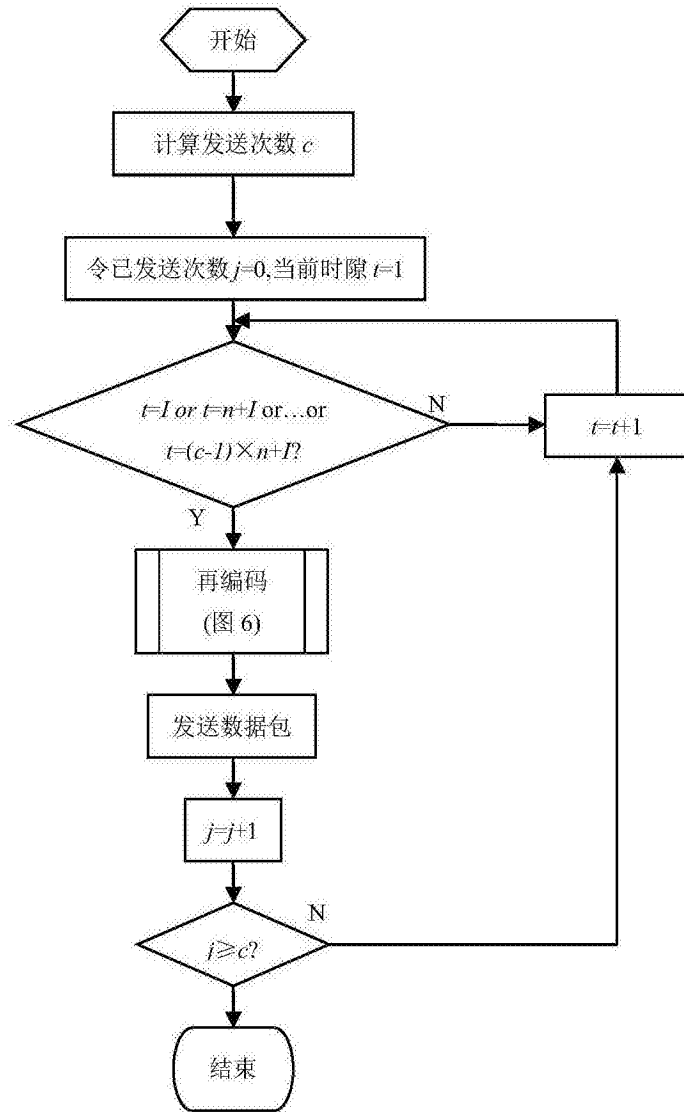


图7

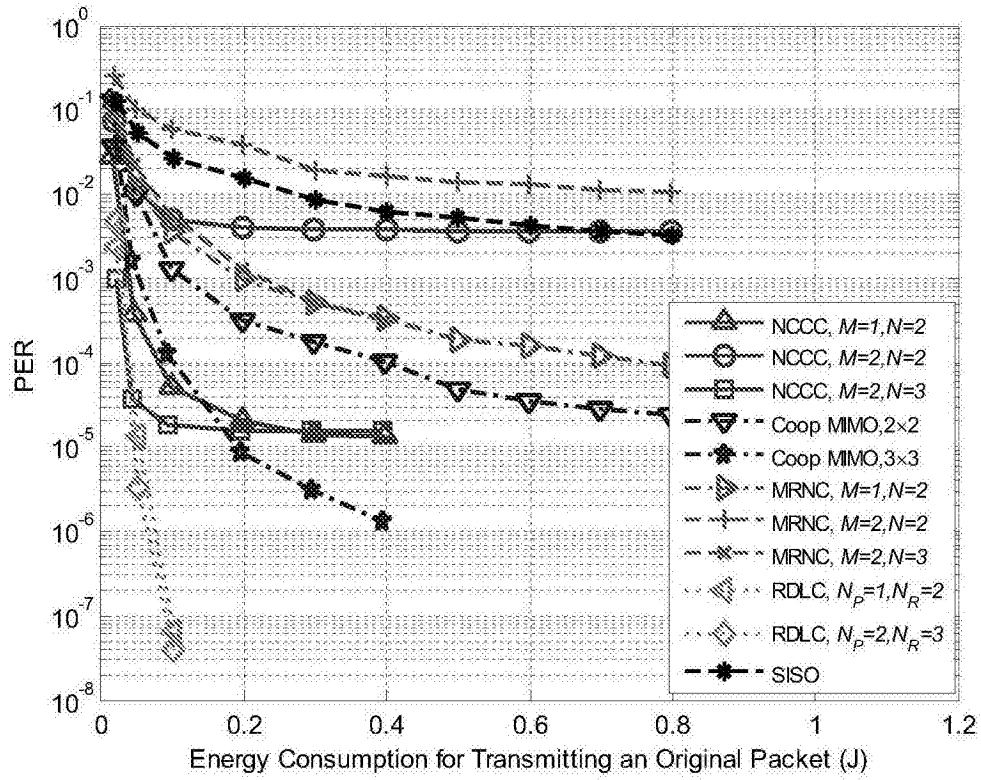


图8

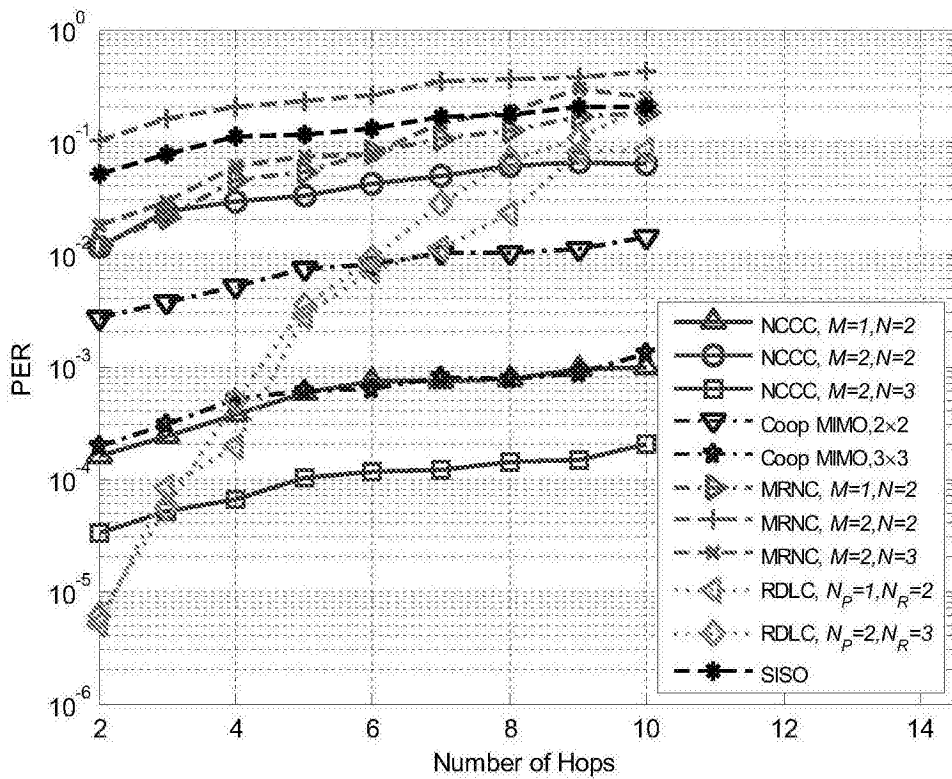


图9

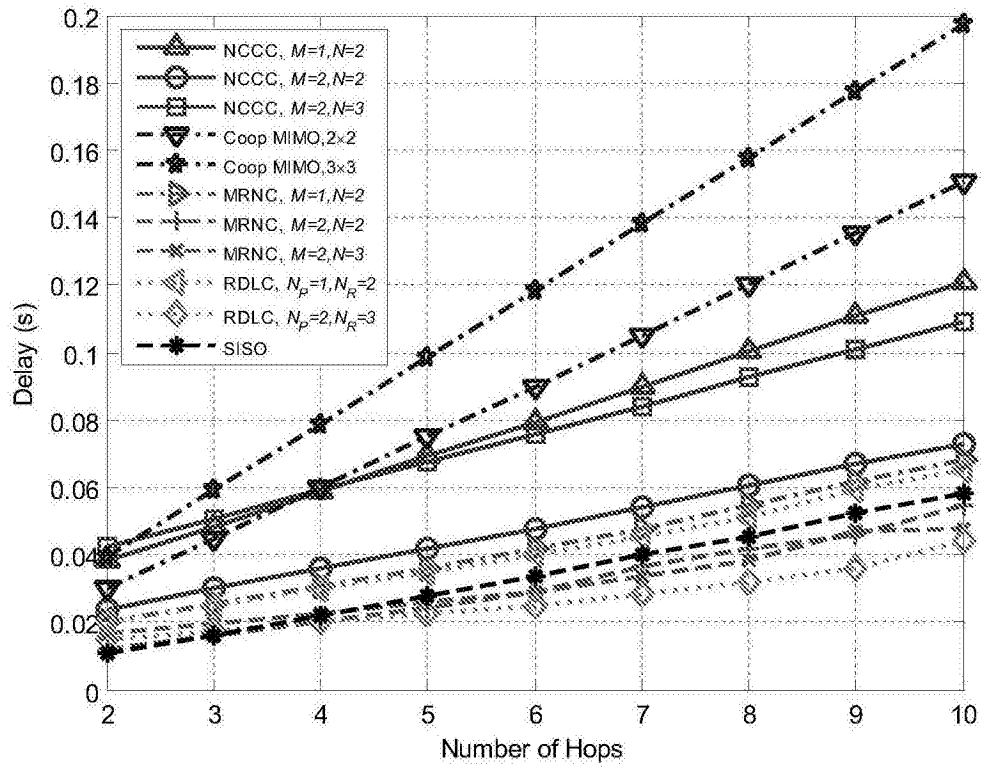


图10