

针对事件驱动型传感器网络的多路径编码路由协议

全杰¹, 钱德沛^{1,2}, 刘轶¹, 李世晗¹

(1. 北京航空航天大学计算机学院, 100191, 北京; 2. 西安交通大学电子与信息工程学院, 710049, 西安)

摘要: 针对事件驱动型传感器网络 Burst-Bulk 的流量特性, 提出了一个基于网络编码和机会转发的多路径单播路由协议——CodePaths. 定义了一个考虑时延和拥塞的路由度量, 以该度量为基础建立了所有节点到 Sink 的梯度场. 当网络中有事件发生时, 节点对数据包进行 Intra-flow 随机线性编码和再编码, 并依据路由梯度和链路质量为多个下游节点分配“转发机会”, 从而形成从源节点到 Sink 的无状态交叉多径 Mesh, 编码包最终汇聚于 Sink, 并解码得到原始数据. 实验结果表明, CodePaths 能够适应事件的时空随机性, 具有较高的吞吐率和良好的实时性, 包传输成功率在 95% 以上, 并在保证可靠性条件下最小化数据传输量, 提高能量效率.

关键词: 无线传感器网络; 网络编码; 机会多径路由

中图分类号: TP393.04 **文献标志码:** A **文章编号:** 0253-987X(2010)06-0039-07

Multipath Coded Routing Protocol for Event-Driven Wireless Sensor Networks

TONG Jie¹, QIAN Depei^{1,2}, LIU Yi¹, LI Shihan¹

(1. School of Computer Science and Engineering, Beihang University, Beijing 100191, China; 2. School of Electronics and Information Engineering, Xi'an Jiaotong University, Xi'an 710049, China)

Abstract: An intra-flow network coding based opportunistic multipath routing protocol called CodePaths is proposed referring to the Burst-Bulk characteristic of traffic in event-driven wireless sensor networks. In CodePaths, a new routing metric, taking delay and congestion factors into account, is defined and implemented for establishing routing gradient field from nodes to sink. When an event occurs in the network area, nodes employ random linear network coding and recoding to mix packets, and assign ‘forwarding opportunities’ to downstream nodes based on the routing gradients and link qualities. Consequently, an interleaving-paths mesh is constructed, along which the encoded packets are routed and finally decoded by the sink node. The extensive simulation results indicate that CodePaths can well adapt to the randomness of events. Compared with other single and multi-path routing, CodePaths with higher throughput and lower latency minimizes the data transmission and energy consumption while keeps the packet delivery rate over 95%.

Keywords: wireless sensor network; network coding; opportunistic multipath routing

机会路由^[1]是一种新颖的无线多跳路由方法, 与传统路由在发包前选择特定下一跳不同, 机会路由允许任何“听到”此次发送的下游节点参与转发,

从而充分利用无线网络物理层的广播特性, 在无线链路不可靠的情况下大大提高了网络吞吐量. 然而, 当多个节点同时听到一个包时, 机会路由需要通过

严格的调度策略减少冗余转发和广播冲突. 引入网络编码机制^[2]后, 数据包在转发前都会与其他包进行随机线性编码, 相同包的输入会产生不同的新编码包的输出, 在保持机会路由特性的同时免去了复杂的调度, 使每一个节点的发送触发其多个下游节点的转发, 从而形成了到目的节点的无状态多径路由. 本文基于网络编码和机会路由方法, 提出了一个多路径的传感器网络路由协议——CodePaths.

现有的传感器网络多路径路由协议多以可靠性设计为目标^[3], 通过各种算法维护路径分离, 选择冗余路径和避免径间干扰^[4]. CodePaths 由“机会转发”而自然形成交叉路径 Mesh, 节点不用显式地交换控制信息来建立和维护路径状态, 因而大大降低了协议复杂度和算法开销. GRAB^[5]采用类似的受限广播建立多路径, 但没有采用网络编码来避免冗余数据的传输. 文献[6]融合网络编码和扩散算法, 选择性地调整多路径拓扑来增加网络容量, 文献[7]提出了一种在传感器网络中使用网络编码时最少路径数的计算方法, 而并非完整的多路径协议, 二者均未使用机会路由来提高链路的空间利用率. 文献[8]采用与本文相似的方法, 但其针对节点密度较低的无线网状网, 也未针对资源受限做优化, 其转发控制算法为集中式启发算法且未考虑节点拥塞.

本文的主要贡献有: ①首次结合网络编码理论和机会多径方法进行传感器网络路由协议的设计; ②定义并实现了一个针对事件驱动型传感器网络的路由度量; ③提出了一个实用的分布式算法来控制编码包的转发, 以保证可靠传输下的能耗最小化.

1 系统模型

(1)网络范型. 我们关注典型的无线传感器网络架构, 即大量资源受限的同构传感节点和一个资源非受限的 Sink 节点部署于目标区域内, 传感节点和 Sink 节点都不具有移动性. 同时, 假设事件触发多个节点产生的数据已在事件信号最强的中心节点进行融合并去除了相关性, 由该“源节点”到 Sink 节点的网络层通信是一个单播路由过程.

(2)应用场景. 传感器网络中的数据传输模式可分为时间驱动(连续)、事件驱动和查询驱动模式. 在事件驱动模式下, 节点的传感模块执行监测任务, 而收发模块大部分时间处于休眠状态或仅交换基本的组网信息. 只有当监测值发生突变或变化值超过预定阈值时, 节点才被触发数据传输行为, 我们称之为事件驱动型传感器网络. 其典型应用包括入侵检测、

目标跟踪、灾难预警、应急响应等.

(3)流量特性. 上述应用模式下的数据流量往往具有如下特性: ①随机性, 事件的发生在时间和空间上都具有不确定性和不可预测性, 因而触发的流量在时空尺度上也具有极大的随机性; ②稀疏性, 事件的发生是低频的, 在空间上也是非均匀的, 因而在全网范围内同时存在多条数据流的概率较低; ③突发性, 事件一旦发生, 通常会在较短时间内产生大量需要报告的数据, 我们称之为 Burst-Bulk 流量, 它对实时性、吞吐量、负载均衡和拥塞避免等都具有较高要求, 这也是本文路由协议的设计目标.

2 协议设计

2.1 方法概述

在 CodePaths 中, 源节点首先将需要报告的数据以每 M 个包为单位分成多个 Bulk, 对每个 Bulk 采用随机线性编码不断生成编码包. 中间节点收到包后对已接收到的同一 Bulk 的编码包重新编码并广播至下游节点, 这样编码包沿多路径源源不断地流向 Sink. 当 Sink 积累了足够的编码包时即可对该 Bulk 解码, 得到 M 个原始数据包, 同时向源节点发送 ACK, 使其转向下一个 Bulk 的传输, 这样的过程持续到所有数据发送完毕.

2.2 路由梯度

由于 Burst-Bulk 数据收集具有严格的实时性要求, 我们以路径时延 T 为指标建立整个网络的路由梯度. 假设一条路由由 n 条链路组成, 节点 i 的路径时延定义为

$$T(i) = \beta T_{\text{BASE}}(i) + (1 - \beta) T_{\text{Event}}(i) \quad (1)$$

$$T_{\text{BASE}}(i) = \min \left\{ \sum_{l=1}^n \left(E_{\text{TX}}(l) \frac{L}{R} \right) : N_i \rightarrow \dots \rightarrow \text{sink} \right\}$$

$$T_{\text{Event}}(i) = HI(i)$$

式中: β 是 0~1 范围内的可调参数; $E_{\text{TX}}(l)$ 为链路 l 上成功发送一个数据包的期望传输次数(包括重传)^[9]; L 是固定包长度; R 是链路带宽; $I(i)$ 为节点 i 中待发送编码包的队列长度; H 是发送一个数据包所花费的时间. 由于 backlog 参数的使用, CodePaths 的路由算法将节点拥塞考虑在内.

T 表述了一个包从节点 i 成功到达 Sink 的期望传输时间, 它是 T_{BASE} 和 T_{Event} 基于 β 因子的指数加权位移平均(EWMA), 可看做是全局资源消耗和本地资源保护的折中.

本文定义 T_{BASE} 为“带宽调整的 E_{TX} ”, 综合了链路质量、带宽和相邻链路干扰等因素, 但没有考虑竞

争退避,仅用 L/R 反映信道发送持续时间. T_{BASE} 反映了无队列情况下的路径端到端最小时延,在整个网络生存周期都进行周期性测量,离 Sink 越远, T_{BASE} 值越大,从而建立了所有节点到 Sink 的“漏斗”状路由梯度场.

T_{Event} 仅在数据传输行为被事件触发时才被更新并加入到 T 中.此时, T_{BASE} 能够防止路由振荡,而 T_{Event} 随发送队列长度而变,直接反映网络负载的动态变化.在网络“宁静”区域,显然有 $T = T_{\text{BASE}}$.

2.3 Intra-flow 网络编码

定义 V 为网络中的节点集合, $S \in V$ 为源节点, $T \in V$ 为 Sink 节点, Intra-flow 网络编码即指节点仅对来自相同的端到端单播流的数据包进行编解码操作,该流由二元组 (S, T) 标识.

编码:源节点以 Bulk 为编码单位,每个 Bulk 包含 M 个原始包.采用随机线性编码(RLC)^[2-3]将当前 Bulk 中的 M 个包 $X_1, X_2, \dots, X_M$ 进行组合

$$Y_j = \sum_{i=1}^M g_{ji} X_i \quad (2)$$

每个结果编码包 Y_j 都在包头携带其全局编码向量 $\mathbf{G}_j = [g_{j1}, g_{j2}, \dots, g_{jM}]$ 作为标签,描述了从原始包到编码包的生成信息.编码系数 g_{ji} 理论上在有限域 F_2^8 以完全独立和分布式的方式随机选取,例如使用 F_2^8 可使每个有限域符号都用 1 B 存储.

再编码:当中间节点听到一个 Innovative 包时,它基于已收到的所有属于相同 Bulk 的编码包生成新的编码包.对编码包的线性再编码仍然是相应原始包的线性组合

$$Y_k = \sum_j g_{kj} Y_j = \sum_{i=1}^M \left(\sum_j g_{kj} g_{ji} \right) X_i \quad (3)$$

所谓 Innovative 包是指其编码向量与 Bulk 内其他包的编码向量线性无关.线性相关性检查即是对这些向量执行高斯消元法^[10].仅对 Innovative 包进行编码有两个原因:首先,无线信道的广播特性和机会路由的“监听”行为都增加了 Non-Innovative 包的接收概率;其次,有限域并不保证随机选取的系数向量 100% 的线性无关^[2].

解码:由于可能的线性相关包的产生,源节点一般会为一个 Bulk 生成多于 M 个编码包,但任何 Bulk 的 Innovative 包数都以 Bulk 的大小,即 M 值为上界.一旦 Sink 收到 M 个 Innovative 包时即可通过高斯消元(需 2 个 $M \times M$ 矩阵空间)解出整个 Bulk

$$\begin{bmatrix} X_1 \\ \vdots \\ X_M \end{bmatrix} = \begin{bmatrix} g_{11} & \cdots & g_{1M} \\ \vdots & & \vdots \\ g_{M1} & \cdots & g_{MM} \end{bmatrix}^{-1} \begin{bmatrix} Y_1 \\ \vdots \\ Y_M \end{bmatrix} \quad (4)$$

式中: $\mathbf{G}^{-1} = (g_{ji})_{M \times M}^{-1}$ 即解码矩阵,是编码矩阵 $\mathbf{G}$ 的逆矩阵,满秩可逆即是要求 $\mathbf{G}$ 中编码向量线性无关的缘故.

2.4 机会多径转发

在 CodePaths 中,中间节点从已收包生成新编码包并广播给潜在的下流节点.由于编码的随机性和无状态性,每个编码包都具有相等的信息熵而无冗余,因而转发节点需生成多少新编码包是基于网络编码进行机会路由的核心问题.同时,最小化必要的发包数对于能量敏感的传感器网络至关重要.

每个节点都维护一个邻居表,表项包括各邻节点的 T 、剩余能量等信息.定义 $N(i)$ 为节点 i 的邻节点集合,即通过物理层广播可达的节点.定义 $D(i) = \{k \mid k \in N(i) \& \& (T(k) < T(i) \parallel T(k) = T(i) \& \& E(k) > E(i))\}$ 为节点 i 的下游节点集合,即 T 值较小,或相等但剩余能量较大的节点.每个节点 $j \in D(i)$ 都应当承担一部分转发任务,节点 i 根据下游节点 T 为其分配的转发机会,称为

$$Q_i(j) = \begin{cases} 1, & \text{if } T(j) = \min\{T(k); k \in D(i)\} \\ \prod_{T(k) < T(j)} [1 - p_i(i, k)], & \text{其他} \end{cases} \quad (5)$$

式中: $p_i(i, k)$ 是节点 i 到节点 k 的包发送成功率,用 3.1 节的方法测得. $Q_i(j)$ 决定了节点 j 收到节点 i 发送时应当转发的概率,其计算思想是最好的下游节点应转发它听到的所有包,次好节点仅转发最好节点可能未听到的部分,依次类推.节点将其分配给每个下游节点的 Q 值放在它发送的编码包的包头.

本地节点的 Q 值存储其所有上游节点为其累积分配的 Q 值总和.当 $Q[i] > 1$ 时,即需要生成和发送新编码包.为保证这些包被下游节点接收到,新编码包数目的计算如下

$$A(i) = \lceil Q[i] \rceil / \left\{ 1 - \prod_{j \in Dw(i)} [1 - p_i(i, j)] \right\} \quad (6)$$

当生成 A 个编码包压入节点的发送队列时,其本地的 Q 值被消耗,队列长度 I 增加.本文提出了一个低复杂度的分布式算法来实现上述解决方案,如图 1 所示.

在 CodePaths 协议中, T 虽然是一个路由度量,但它并不被真正用于下一跳的选择,而是在计算 Q

```

1: 根据 $T$ 对下游节点排序, $T$ 值相等时剩余能量多者优先;
2:  $P_Q=1$ ;
3: For 依次排序的 $j \in D(i)$  do
4:    $Q_i(j) \leftarrow P_Q$ ;
5:    $P_Q \leftarrow P_Q(1-p_f(i,j))$ ;
6: end For
7:  $A(i) \leftarrow \lfloor Q[i] \rfloor / (1-P_Q)$ ;
8:  $Q[i] \leftarrow Q[i] - \lfloor Q[i] \rfloor$ ;  $\Delta$  节点 $i$ 的累积 $Q$ 值被消耗
9:  $I(i) \leftarrow I(i)+A(i)$ ;  $\Delta$  新编码包入队

```

图1 对下游节点转发限额的分配算法

时用于对转发节点的排序。另外, T 将队列“back-pressure”信息显式地包含进来, 节点负载越重, 分配 Quota 时优先级就越低, 得到的 Q 值就越小, 从而有效避免了拥塞。

当选择多个转发节点并分配了转发任务后, 多路径即被隐式地建立起来。不同于传统多径路由, CodePaths 仅使用受控的广播来自然发掘多条机会路径, 不需要计算和维护路径状态。图2示意了一个简化网络的机会多路径转发机制。

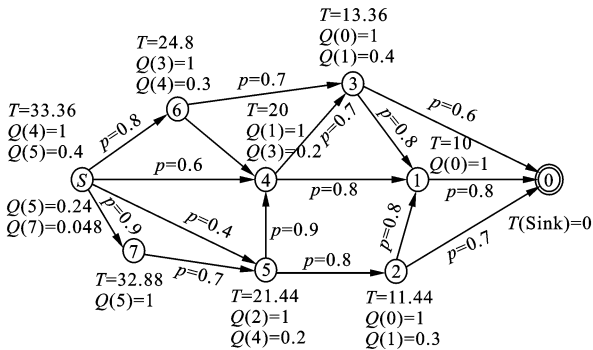


图2 机会多路径转发示意图

3 协议实现

3.1 T 的测量

在 CodePaths 中, 数据包被节点广播发送, 802.11/802.15.4 网络对广播包不进行链路层 ACK, 因而一条链路的 E_{TX} 仅由其单向转发成功率 p_f 计算

$$E_{TX}(l) = 1/p_f \quad (7)$$

式中: p_f 为包成功到达接收端的统计概率, 使用专门的 Beacon 包进行测量。每个节点以平均周期 τ (实现中取 10 s) 广播固定长度的 Beacon, 该广播不会被确认或重传。节点 i 到 j 的转发概率等于节点 j 到 i 的接收概率 p_r , 即

$$p_f(i,j) = p_r(j,i) = \frac{c(t-w,t)}{w\tau^{-1}} \quad (8)$$

节点 i 记录在最近滑动时间窗口 w 内 (取 50 s) 从每个邻节点接收的 Beacon 数 $c(t-w,t)$ 中, 并在它发送的 Beacon 包中携带所有 $p_r(j,i)$ 信息 ($j \in N(i)$), 邻节点可从最新接收的 Beacon 中获得相应的 p_f 。

本文采用类似链路状态路由的方法交换上述链路信息, 每个节点运行 Dijkstra 算法, 计算其 T_{BASE} 代价。 I 为节点本地变量, 后台线程监测编码包队列长度的变化, 提供 I 值为 T 计算所用。

3.2 包结构

CodePaths 的包结构如图3所示。由于目的节点 Sink 的唯一性, 包头中仅有源节点和发送节点地址。节点用其最近更新值填充 T 和剩余能量域, 用 FlowID 和 BulkID 对编码包的归属进行标识, 并在 Forward Block 中按 T 的顺序列出可参加转发的邻节点及其被分配的 Q 值。编码向量携带了该包的全局编码系数。SourceID/FlowID/BulkID 由源节点初始化并被每个转发节点拷贝, 其他域都由转发节点计算和更新。另外, 若该包属于该 Flow 的最后一个 Bulk, 源节点会在 Flag 域中标识。

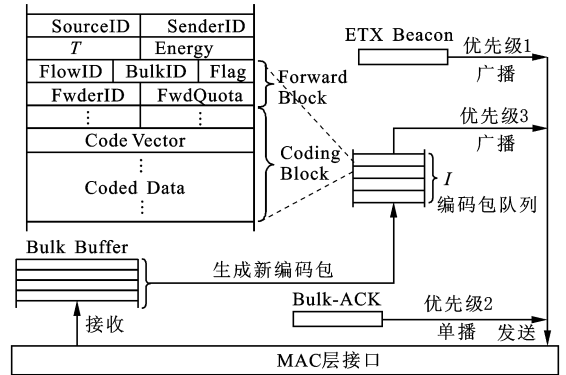


图3 CodePaths 的包结构及收发示意图

传感器网络往往节点密度较大, 具有较低 Q 值的转发节点对路由的贡献很小, 因而为减小信道竞争并降低包头开销, CodePaths 会剪掉 $Q < 0.05$ 的转发节点, 并将转发节点数目限制为 5。Bulk Buffer 缓冲区最大为 $M \times L$ (B), 存放由 MAC 层接口收到的当前 Bulk 的 Innovative 包, 由其产生新编码包并加入编码包队列。

此外, 链路质量的周期性测量与更新是所有先验式路由协议的固有开销, 并非 CodePaths 特有。为保证路由梯度的创建与维护, 系统实现中设置协议数据包, 如 ETX Beacon、Bulk-ACK 等的优先级高于编码数据包。

3.3 编解码开销

编码向量的捎带是包头的主要开销之一。当编

码系数在 F_2^8 内选取时,若 M 取 8 即使编码向量占用 8 B,这对于如 TinyOS 中 46 B 的包长也开销过大. 本文限定第 1 个编码系数恒为 1, 剩余 $M-1$ 个编码系数从 0 至 $r-1$ (r 为一个质数) 的整数中随机选取, 设为 $g_1, g_2, \dots, g_{M-1}$. 在包头中用单个整数来表征这些系数, 即 $\sum g_i r^{M-1-i}$, 其取值范围为 $0 \sim r^{M-1}-1$, 因而需要占用 $\text{lb}r^{M-1}(\text{b})$. 同时, 在进行线性加乘时有可能发生溢出, 若包长为 L , 则其数值不大于 2^L-1 , M 个包线性编码后不大于 $(2^L-1) + (r-1)(M-1)(2^L-1)$, 需要占用 $\text{lb}[1 + (r-1)(M-1)] + L(\text{b})$, 即溢出位为 $\text{lb}[1 + (r-1)(M-1)]$. 综上所述, 编码系数引起的包头开销为

$$L_{\text{code}} = \lceil \text{lb}r^{M-1} \rceil + \lceil \text{lb}[(r-1)(M-1) + 1] \rceil \quad (9)$$

限制编码系数的取值范围自然会增加线性相关包产生的概率. 假设已存在 j 个线性无关向量 $\mathbf{G}_1, \mathbf{G}_2, \dots, \mathbf{G}_j$, 下一个随机生成的编码向量 $\mathbf{G}_{j+1}$ 若与这些向量线性相关, 则一定存在常数 $e_1, e_2, \dots, e_j$ 使 $\sum_{k=1}^j e_k \mathbf{G}_k = \mathbf{G}_{j+1}$. 由于本文设定第 1 个编码系数恒为 1, 故有 $\sum_{k=1}^j e_k = 1$, 即 $(e_1, e_2, \dots, e_j)$ 有 r^{j-1} 种不同选择, 也即存在 r^{j-1} 个向量线性相关于 $\mathbf{G}_1, \mathbf{G}_2, \dots, \mathbf{G}_j$, 那么 $\mathbf{G}_{j+1}$ 线性无关于这些向量的概率为 $1 - r^{j-1}/r^{M-1} = 1 - 1/r^{M-j}$, 亦即在此种系数选择机制下收到第 $j+1$ 个包为 Innovative 包的概率. 设 W_j 为得到 j 个 Innovative 包所期望的平均编码包数, 则有 $W_1=1$ 且 $W_{j+1}=W_j+1/(1-1/r^{M-j})$, 计算得到

$$W_M = 1 + \sum_{j=1}^{M-1} 1/(1-1/r^{M-j}), j \leq M-1 \quad (10)$$

M 和 r 的取值是通信开销和计算效率的折衷. M 越大, 网络编码的收益率越大, r 越大, 编码向量线性相关的概率越低, 但较大的 M 和 r 会增加包头长度和解码操作的复杂度. 通常我们取 $M=8$, $r=5$, 由式(9)、式(10), $L_{\text{code}}=22 \text{ b}$, 少于 3 B, $W_M=W_8=8.30$, 接近于解码 Bulk 所需的最小包数.

在 CodePaths 中, 编解码速率远大于传感器网络数据传输速率. 源节点编码开销最大, 它要对 Bulk 内所有 M 个包编码, 而转发节点的编码开销取决于其收到的 Innovative 包数 ($\leq M$). 同时, 转发节点检查 Innovative 仅对编码向量进行高斯消元, 完全的解码操作只在 Sink 节点运行.

3.4 协议控制流

在 CodePaths 协议中, 流量由源节点注入网络, 当转发节点收到足够多的包且 Quota 值累积大于 1

时才产生流量. 源节点持续发送数据包直至最后一个 Bulk 被 Sink 确认. 为使源节点在最短的时间内转向下一个 Bulk 的传输, Sink 收到第 M 个 Innovative 包时先发送 Bulk-ACK 再解码. ACK 沿路径时延最短的路径路由至源节点, 并在每一跳都使用重传机制以确保可靠传输. 显然, 基于 Intra-flow 网络编码, 每 M 个包仅需要一个 ACK.

除了机会转发, 本文还基于无线信道广播特性, 采用另外两个“监听”来进一步改进 CodePaths 的性能: 当上游节点听到下游节点发送的数据包时, 立即更新其邻居表中相应的 T 和 E 并重新排序; 任何中间节点若听到 Bulk-ACK, 立即停止过期 Bulk 包的发送并清除 Bulk Buffer. 图 4 描述了节点 i 收到节点 j 发送的数据包的详细处理流程.

```

1: If Bulk-ACK then to step 23;
2: Check  $T$  and Energy;
3: If  $j \in D(i)$  then
4:   插入或更新邻居表项并重新排序;
5: Else  $\Delta i \in D(j)$ 
6:   If  $i \in \text{FwdList}(j)$  then
7:     If BulkID BulkID( $i$ ) then
8:       清除过期bulk buffer;
9:        $Q[i] + Q_j(i)$   $\Delta i$ 的 quota 值累积
10:      If innovative then  $\Delta$  检查 innovation
11:        存入当前bulk buffer;
12:        If  $i$  是 sink then
13:          If 已得到  $M$  个编码包 then
14:            发送Bulk_ACK;
15:            对Bulk解码并传递到上层;
16:      Else If  $Q[i] > 1$   $\Delta i$ 应履行转发义务
17:        分配 $Q_j(D(i))$ , 更新 $Q[i], I(i)$   $\Delta$  § 2.4算法
18:        生成  $A(i)$  个新编码包并入队;
19:         $A(i)=0$ ;
20:      Else 丢弃该包;  $\Delta$  非 innovative包
21:      Else 丢弃该包;  $\Delta$  过期包
22:      Else 丢弃该包;  $\Delta i$ 过期包被剪掉
23: 清除已ACK的 bulk;
24: If 是源节点 and 数据流未结束 then 转向下一个 bulk;
```

图 4 包接收触发的协议控制流程

4 性能评估

本文使用 NS2(v2.31) 对 CodePaths 进行仿真实验, 并与两种路由协议进行性能比较与分析.

(1) GRAB^[9]. GRAB 是 UCLA 实现的传感器网络多路径可靠路由协议, 它建立并维护所有节点到 Sink 方向的 Cost 梯度场, 利用类机会转发机制完成数据传输. 实验中用 E_{Tx} 作为其 Cost.

(2) Directed Diffusion^[11]. 我们关注 DD 的加强路径阶段, 使用 E_{Tx} 作为其路径选择度量, 并基于 ns/diffusion3 加入网络编码功能, 使其变成使用网络编码的 E_{Tx} 最短路径路由, 称为 CodeDD.

4.1 仿真模型

本文以 2.4G IEEE802.15.4(LRWPAN)作为 MAC 层协议. 实验中的节点随机均匀分布于矩形监测区域内, Sink 位于区域中下方. 根据系统模型中描述的流量特性, 设定网络中存在一个事件并持续 5 min, 激发源节点以 1 包/s 的速率产生报告包. 每轮实验按照 CodeDD、CodePaths 和 GRAB 的顺序进行, 重复 10 轮取平均值. E_{TX} 、 T 按照 3.1 节的方法实测得到, 其他仿真参数如表 1 所示.

表 1 仿真参数设置

区域尺寸/m×m	200×200
节点数目	200
节点部署	均匀随机
L/B	127
Bulk 大小(M)	8
Radio 发送半径/m	20
$R/\text{kb} \cdot \text{s}^{-1}$	250
$H/\text{kb} \cdot \text{s}^{-1}$	1/115.2
β	0.65
信道访问机制	CSMA-CA/Beacon Enabled
数据报告速率/包 $\cdot \text{s}^{-1}$	1 (300 s CBR 流量)

4.2 仿真结果及分析

定义 1 从事件结束(即从 300 s 开始计时)到 Sink 不再收到新包之间的时间间隔为网络时延, 也表征了完成整个事件报告的网络吞吐率.

定义 2 Sink 成功接收到的总有效包数与应得包数(300)的比值为包传输成功率.

定义 3 以网络中参与数据传输的所有节点的总发包数作为网络能耗的间接度量.

事件的发生具有空间随机性, 分别选取与 Sink 距离不同的节点作为源节点进行实验, 因其具有特定下一跳的单径路由, 该“距离”由 CodeDD 的跳数来衡量. 统计上述 3 个性能指标, 实验结果如图 5~图 7 所示.

由图 5 可知, CodeDD 的吞吐率小于多路径路由的吞吐率. 在跳数较小(≤ 3)时, 因不进行编解码操作, GRAB 的时延小于 CodePaths 的时延. 随着跳数的增加, 基于网络编码的多路非拷贝数据并行传输对吞吐率的贡献越明显, 而且 CodePaths 使用 T 作为转发度量, 避免经过负载较重的瓶颈节点, 进一步优化了时延性能和能耗平衡.

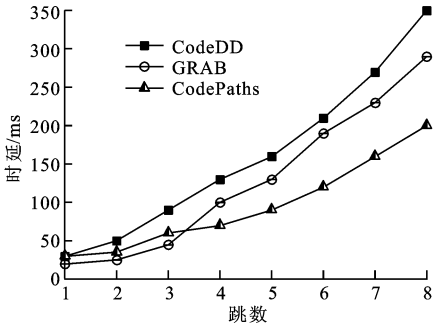


图 5 时延仿真结果比较

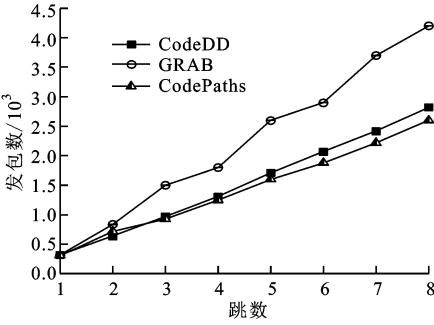


图 6 包发送次数结果比较

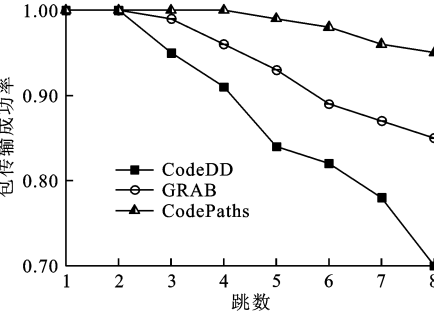


图 7 包传输成功率仿真结果

CodeDD 向下一跳发送 Unicast 数据, 会得到 MAC 层 ACK, 丢包重传是其发包数额外增加的主要原因. CodePaths 利用 Quota 分配算法控制编码包的转发, 并采用了拥塞避免机制, 因而其总发包数甚至低于单路径的 CodeDD. GRAB 由于没有采用网络编码方法, 在广播转发中会产生大量冗余包, 大大降低了网络的传输效率和能量效率.

由图 7 可知, 单路径的 CodeDD 和多路径的 GRAB 都会因跳数增加而遭受更大的丢包率, CodeDD 甚至会因严重的丢包导致 Sink 无法解码. CodePaths 的机会路由保证无线链路的充分利用和编码包的足量发送, 编码机制又减小了其对特定包的依赖, 完整得到所有原始包的概率在 95% 以上.

CodePaths 利用参数 β 和 Q 将负载均衡和拥塞

控制纳入路由算法中. 上述实验都设定 β 为一定值 (0.65), 图 8 考察了 β 的取值对 CodePaths 性能的影响. 由于 802.15.4 在发包速率高于 1 包/s 时丢包率会急剧上升^[12], 因此配置事件区域内多个节点都作为源节点同时发包来增加网络负载和编码包队列长度. 显然, β 的取值反映了上游节点为下游节点分配转发机会时链路质量和拥塞避免两个因素的折中.

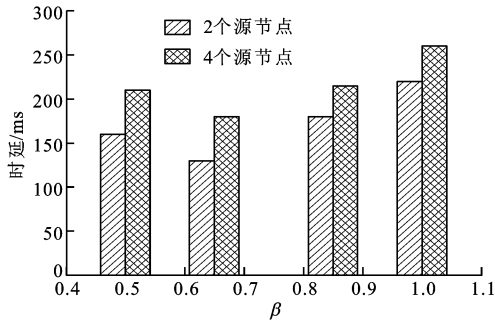


图 8 β 取值对时延的影响

5 结束语

CodePaths 充分利用无线链路的广播特性, 采用机会路由建立了多路径传输机制, 并引入网络编码来降低这种机制的复杂度, 因此可用相对廉价的计算资源换取网络传输效率和网络健壮性. 构建路由梯度和运用分布式转发控制算法使 CodePaths 能够避免拥塞, 平衡网络负载, 并最小化广播发送次数. 通过仿真实验, 发现 CodePaths 的性能优于其他单路径和多路径路由, 具有高吞吐率、高可靠性、低时延和低能耗的特点, 且协议运行开销小, 非常适合事件驱动型传感器网络的数据收集.

参考文献:

- [1] BISWAS S, MORRIS R. ExOR: opportunistic multi-hop routing for wireless networks [C]//Proceedings of ACM SIGCOMM 2005. New York, USA: ACM, 2005: 133-144.
- [2] CHOU P A, WU Yunnan, JAIN K. Practical network coding [C]//Proceedings of the 41st Annual Allerton Conference on Communication, Control, and Computing. Monticello, USA: UIUC, 2003: 40-49.
- [3] GANESAN D, GOVINDAN R, SHENKER S, et al. Highly-resilient, energy-efficient multipath routing in

- wireless sensor networks [J]. ACM SIGMOBILE Mobile Computing and Communications Review, 2001, 5 (4): 11-25.
- [4] VOIGT T, DUNKELS A, BRAUN T. On-demand construction of non-interfering multiple paths in wireless sensor networks [C]//Proceedings of the 2nd Workshop on Sensor Networks at Informatik. Bonn, Germany: SICS, 2005: 96-101.
- [5] YE Fan, ZHONG G, LU S, et al. Gradient broadcast: a robust data delivery protocol for large scale sensor networks [J]. ACM Wireless Networks, 2005, 11(3): 285-298.
- [6] TOLEDO A L, WANG X D. Efficient multipath in sensor networks using diffusion and network coding [C]//Proceedings of the 40th Conference on Information Sciences and Systems. New Jersey, USA: IEEE Press, 2006: 87-92.
- [7] 李姗姗, 廖湘科, 朱培栋, 等. 基于网络编码的无线传感网多路径传输方法 [J]. 软件学报, 2008, 19 (10): 2638-2647.
- LI Shanshan, LIAO Xiangke, ZHU Peidong, et al. A method for multipath routing based on network coding in wireless sensor network [J]. Journal of Software, 2008, 19(10): 2638-2647.
- [8] CHACHULSKI S, JENNINGS M, KATTI S, et al. Trading structure for randomness in wireless opportunistic routing [C]//Proceedings of ACM SIGCOMM 2007. New York, USA: ACM, 2007: 169-180.
- [9] COUTO D, AGUAYO D, BICKET J, et al. A high-throughput path metric for multi-hop wireless routing [C]//Proceedings of the 9th Annual International Conference on Mobile Computing and Networking. New York, USA: ACM, 2003: 134-146.
- [10] KOETTER R, MEDARD M. An algebraic approach to network coding [J]. IEEE/ACM Transaction on Networking, 2003, 11(5): 782-795.
- [11] INTANAGOWIWAT C, GOVINDAN R, ESTRIN D, et al. Direct diffusion NS2 implementation [EB/OL]. [2002-10-02]. <http://www.isi.edu/scadds/software/>.
- [12] ZHENG Jianliang, LEE M J. A comprehensive performance study of IEEE 802.15.4 [M]//Sensor Network Operations; Chapter 4. Piscataway, NJ, USA: IEEE Press, 2006: 218-237.

(编辑 杜秀杰 苗凌)