

基于网络编码的无线网络多路径机会路由算法

韩 莉^{1,2} 钱焕延¹

(南京理工大学计算机科学与技术学院 南京 210094)¹ (安徽大学计算机科学与技术学院 合肥 230039)²

摘 要 提出了一个效用优化模型,以最大化整个网络的吞吐量,并在各个流之间均衡分配带宽资源。基于该模型,结合无线网络的广播特性和机会路由的思想,构造单播多路径路由,并应用网络编码简化数据调度。性能分析与仿真实验表明,算法在稠密网络及存在多个流的条件下,可以取得比同类协议更大的吞吐量、更低的网络整体开销和更合理的带宽资源的分配。

关键词 网络编码,机会路由,多路径,多个流

中图分类号 TP393 **文献标识码** A

Algorithm of Network Coding Based Multipath Opportunistic Routing for Wireless Networks

HAN Li^{1,2} QIAN Huan-yan¹

(School of Computer Science & Technology, Nanjing University of Science & Technology, Nanjing 210094, China)¹

(School of Computer Science and Technology, Anhui University, Hefei 230039, China)²

Abstract We considered wireless mesh networks, and exploited the inherent broadcast nature of wireless by making use of opportunistic multipath routing. We presented an optimization framework that enables us to derive optimal flow control, routing, scheduling schemes, where we use network coding to ease the routing problem. The simulation shows that in the dense network with multi-flows, the algorithm can achieve higher throughput improvement and fairer allocation of bandwidth compared to other protocols of the same genre.

Keywords Network coding, Opportunistic routing, Multipath routing, Multi flow

针对无线信道的广播、时变、丢失特性和确定性路由策略的不足,麻省理工学院(MIT)的 Biswas 等人于 2004 年率先提出了机会路由(也称作机会转发)的概念^[1]。机会路由通过多个潜在中继节点竞争、自主智能判断进行下一跳节点选择,充分利用信道广播特性,提高吞吐量和传输可靠性。研究机会路由算法来提升无线多跳网络的性能,已成为当前无线自组织网络与传感器网络组网协议研究中的一个重要方向。

网络编码(Network Coding)技术由 R. Ahlswede 等人于 2000 年首次提出。该技术可以极大地提高网络的吞吐量和可靠性。

如何同时发挥机会路由和网络编码的优势,文献[3]中的 MORE 协议对此进行了尝试。MORE 是一个与 MAC 无关的协议,它通过引入网络编码解决了数据传输过程中节点协作难的问题,即:转发节点可以不管其他节点发送的是什么数据,而只需要将自己缓存下来的数据进行编码后转发出去即可,目的节点收到一定数量的编码包后解出相应的原数据包。同时它充分利用了网络的空间关系,通过多个节点的协作转发数据包,而不是某一个节点转发。但 MORE 并没有解决多个流的调度问题,随着流数量的增加,协议总体性能下降。

Baolin Sun 等给出了一种基于网络编码的单播多路径路由算法^[4],在中间节点上进行完全编解码。但由于对转发节

点和转发路径没有任何约束,糟糕的情况下可能会形成全网广播,对于信道资源有限的无线链路来说无疑是雪上加霜。

本文提出了一个效用优化的模型,旨在最大化网络吞吐量,并在各个流之间均衡地分配资源。基于该优化模型,提出一个分布的、独立于 MAC 层节点调度的启发式算法,应用可控的要素,如流的选择、传输速率的控制等,以实际发送的分组数量指引流量分配,旨在建立一个实用的单播多路由选择协议。性能分析与仿真实验表明,算法在稠密网络及存在多个流的条件下,可以取得比 MORE 协议和文献[4]中的无控制多路径路由更大的吞吐量、更低的网络整体开销和更合理的带宽资源的分配。

1 模型

1.1 路由及流调度

定义 \underline{N} 为网络上的节点集, $N = |\underline{N}|$, $L(\underline{N})$ 为 \underline{N} 所有子集的集合。定义 \underline{L} 为所有点到点链路 (i, j) 及超边 (i, J) 的集合,其中 $i, j \in \underline{N}$, $J \subset \underline{N}$ 。由于存在大量可能的无线链路,我们仅考虑那些有效链路,比如, J 可以选为信干比大于一定阈值的节点。

定义 C 为网络上端到端单播流的集合,其中每个流 $c \in C$ 有一个源 $Src(c) \in \underline{N}$ 和一个目的节点 $Dst(c) \in \underline{N}$, f_c 为流 c 的

到稿日期:2013-07-05 返修日期:2013-11-03 本文受国家自然科学基金项目(61202227)资助。

韩 莉(1975—),女,博士生,讲师,主要研究方向为计算机网络技术及应用,E-mail:hanli98@ahu.edu.cn;钱焕延(1950—),男,教授,博士生导师,主要研究方向为计算机网络技术及应用。

速率。每个流可以选择满足活动配置的任意一条路径,也可以选择从源节点到目的节点的多条路径。我们假定由转发节点决定每个流的转发流量。

定义流调度配置矩阵 $A, A_k = 1$ 表示节点 i 转发流 c 的数据, $A_k = 0$ 表示节点 i 不转发流 c 的数据。如果对所有 $i \in N$ 仅有一个 $c \in C$, 使得 $A_k = 1$, 则称流调度配置 A 是合法的, 令 \underline{A} 表示所有合法的流调度配置矩阵的集合。

记信道状态向量 S 是取值于有限域 \mathcal{S} 上的独立同分布变量; 速率向量 $R \in \underline{R}$, 其中 \underline{R} 是物理层允许发送速率的集合, 定义 η 是速率 R_i 的上界。

假定信道状态 S 、路由 A 、速率分配 R 取值于一个有限状态空间, 因此联合配置 (S, A, R) 具有一个时间平均概率, 定义为 $\alpha_{S,R,A}$, 其中 $\alpha_{S,R,A} \geq 0$, $\sum_{S,R,A} \alpha_{S,R,A} \leq 1$ 。

定义 $p_{ij}(R_i, S)$ 表示节点 j 从节点 i 成功接收一个分组的概率。定义 $p_Z(P, R_i, S) = \prod_{j \in Z} p_{ij}(R_i, S)$ 表示节点 i 发送的信息被 Z 中所有节点正确接收的概率。其中 $Z \subset L(N)$, $(i, j) \in \underline{L}$ 。

1.2 可行速率集合

为每个流 c 分配一个端到端速率 f_c , 记速率向量 $f = (f_c)_{c \in C}$ 。满足如下 3 个条件的速率向量称为合法的速率向量。

$$f_{Dst(c), Z} = 0 \quad (1)$$

$$\sum_{j \in N} \sum_{Z(j, Z) \in \underline{L}, i \in Z} f_{jZ} + f_c l_{i=Src(c)} \leq \sum_{j \in Z} f_{jZ} = |Z| R_{iZ} \quad (2)$$

$$R_{iZ} = \sum_{S,R,A} \alpha_{S,R,A} A_{i,c} R_Z \quad (3)$$

其中, $R_{i,Z} = R_i \cdot p_Z(R, S)$ 。从式(2)可以看出, 当节点 i 成功传输一个分组到 Z 时, Z 中的每个节点都将获得一个授权。由式(3)可以看出, $\{R_{i,Z}\}_{i,Z} \subseteq \text{Hull}(\{R_{i,Z}(R, S)\}_{i,Z})$ 。

定义 1 如果每个流 c 均以速率 f_c 从源节点 $Src(c)$ 向目的节点 $Dst(c)$ 发送数据, 则称速率向量 f 是可行的。

定理 1 令 \underline{F} 是端到端速率向量 f 的集合, 并满足: 存在向量 $f = (f_{jZ})_{(j,Z) \in \underline{L}, c \in C}$, $R = (R_{iZ})_{(i,Z) \in \underline{L}, c \in C}$, $\alpha = (\alpha_{S,R,A})_{S \in \mathcal{S}, R \in \underline{R}, A \in \underline{A}}$ 满足条件(1)–(3)。向量 f 是可行的当且仅当 $f \in \underline{F}$, \underline{F} 是凸的。

定理 1 表明, 当允许对一个流中所有分组进行编码时, 每个流的目的节点接收独立物理分组的速率等于发送分组的数目。

2 效用最大化模型

我们将网络的效用最大优化问题描述为:

$$\max_{c \in C} \sum U_c(f_c), f \in \underline{F} \quad (4)$$

效用函数 $U_c(f_c)$ 是一个严格凹的、端到端速率 f_c 的递增函数。例如, $U_c(f_c) = \log(f_c)$ 表示比例公平策略, $U_c(f_c) \propto 1/f_c$ 接近 TCP 的效用^[6]。效用最大化的目标是在效率与公平之间进行权衡。

假定效用函数 $U_c(\cdot)$ 满足如下条件:

a) $U_c(\cdot)$ 是一个严格凹的递增函数;

b) $U_c(\cdot)$ 二阶可导, 且对任意 $m, M, 0 < m < M < \infty$, 存在

常数 $0 < c < C < \infty$, 使得对 $\forall x \in [m, M]$, 有 $\frac{c}{c} \leq -1/U_c''(x) \leq \frac{C}{c}$ 。

由于集合 \underline{F} 是凸的且目标函数是严格凹函数, 则存在唯

一的解 f^* 使得该问题有最大值, 对应有 R^* , 但不一定唯一。

最优化问题的 KKT 条件表示为:

$$\mu_i^* \left(\sum_j y_{ij}^{c*} - \sum_j y_{ji}^* - f_c^* l_{i=Src(c)} \right) = 0 \quad (5)$$

$$f_c^* (U_c'(f_c^*) - \mu_{Src(c)}^*) = 0 \quad (6)$$

其中, $y_{ij}^*(t) = \sum_{Z(i,Z) \in \underline{L}, j \in Z} f_{ijZ}^*(t)$ 。从式(5)看出, 只有当流入节点 i 的 c 流量大于流出节点 i 的 c 流量时, μ_i^* 为正。我们可以将 μ_i^* 与流 c 在节点 i 上的队列长度 q_i^c 相关。

从 KKT 条件可以推导出:

$$\mu_i^* - \mu_j^* \geq 0 \quad (7)$$

$$R_{ij}^* = \operatorname{argmax}_{R_i} \sum_i \max_c \max_{(i,j) \in \underline{L}} (\mu_i^* - \mu_j^*) \cdot R_i p_{ij} \quad (8)$$

其中, 式(7)用于背压路由进行流量控制, 式(8)用于优化调度。

2.1 优化算法

假定反馈及时且不会丢失。本节中, 我们将构造一个收敛于最优化问题 $\max_{c \in C} \sum U_c(f_c)$ 的最优值的优化算法。

定义 $q_i^c(t)$ 为节点 i 上流 c 的分组队列, 当一个分组从 i “成功传输”, $q_i^c(t)$ 减 1。这里“成功传输”是指至少存在一个集合 Z , Z 成功接收了这个分组。当节点 i 成功接收一个分组, $q_i^c(t)$ 加 1。目的节点上所有组的队列长度均设置为 0。

节点 i 和节点 j 之间的背压定义为: $z_{ij}^c = q_i^c - q_j^c$; 当 $z_{ij}^c \geq 0$ 时, 节点 i 向 j 发送一个分组。

最优调度即下列最优化问题的解:

$$w_i(t, R, S) = \max_c \max_{(i,j) \in \underline{L}} z_{ij}^c \cdot R_{ij} \quad (9)$$

$$(R, S) = \operatorname{argmax}_{S,R} \sum_i w_i(t, R, S) \quad (10)$$

$$c_i^*(t) = \operatorname{argmax}_c \max_{(i,j) \in \underline{L}} z_{ij}^c \cdot R_{ij} \quad (11)$$

$$A_i^c(t) = l_{c=c_i^*(t)} \quad (12)$$

$$R_{ij} = A_i^c(t) \cdot R_{ij} \quad (13)$$

$$R_{ij}(t) = R_i(R, S) \cdot p_{ij}(R, S) \quad (14)$$

其中, 若 $b = \text{true}$, $l_b = 1$, 否则 $l_b = 0$ 。

式(9)–式(14)是速率和信道状态的联合调度问题。通过求解式(9), 求得最优化调度 $(S(t), R(t))$; 式(10)决定了节点在时隙 t 上所传输的流。

为使每个流能公平使用网络资源, 我们使用一种原始对偶算法, 使每个流的源节点根据自己的本地队列长度信息及与这个流相关的效用函数以迭代的方式更新流速率 $f_c(t)$, 如定义 2 所示。

定义 2(原始对偶拥塞控制^[5]) 在每个时隙 t 的开始, 每个流 c 读取其源节点的队列长度 $q_{Src(c)}^c(t)$, 并按如下方式更新 c 的流速率 $f_c(t)$:

$$f_c(t+1) = \{f_c(t) + \gamma(KU_c'(f_c(t)) - q_{Src(c)}^c(t))\}_m^M \quad (15)$$

记号 $\{y\}_a^b$ 表示将 y 的值投射到区间 $[a, b]$ 上最近的节点。 m 是一个任意小的正数, $M > 2 \frac{\hat{A}}{\eta}$, \hat{A} 是任意链路上的最大速率。取 $\gamma = 1/K^2$ 。

2.2 算法实际应用讨论

式(10)给出的优化问题是一个 NP 难题, 本节从实际角度出发, 给出一个分布式次优化调度过程。

一些无线随机接入过程, 如 802.11, $S(t)$ 满足: 对所有 $(i_1, J_1) \in S(t)$, 不存在 $(i_2, J_2) \in S(t)$ 使得 $p_{i_1, i_2} > 0$ 及 $j \in J_1 \cap J_2$ 使得 $p_{i_1, j} > 0, p_{i_2, j} > 0$ 。即一个节点不能同时发送和接收分组或从同一个节点接收分组。为保证分组的可靠传输,

我们使用一种简单的 ACK 应答方式,发送节点可以根据接收节点的应答来修改自己的队列。

由于链路调度过程不受算法控制,我们可以将式(9)一式(14)的优化调度简化为:

$$(c^*(t), R_i(t)) = \operatorname{argmax}_{c, R} \sum_{(i,j) \in L} z_{ij}^c \cdot R_{ij}$$

当链路层速度相同时,优化调度可以进一步简化为:

$$(c^*(t)) = \operatorname{argmax}_c \sum_{(i,j) \in L} z_{ij}^c \cdot R_{ij} \quad (16)$$

作为优化算法(4)的一种启发式实现,式(16)可由每个节点在本地计算得到。

3 协议基本思想

我们的协议面向任意时刻总是连通的无线 Mesh 网络。针对延迟要求较小的应用,主要考虑大尺寸文件的传输。所研究的机会路由协议主要考虑如何利用无线链路的广播特性和终端的空间多样性来提高无线网络吞吐量和传输可靠性。协议位于 MAC 层之上,提供可靠文件传输。

3.1 术语

流:流由〈源,目的〉节点对唯一确定,我们假定从同一源节点发往同一目的节点的数据均属于一个流。

原始分组(native packet):由源节点产生的、未编码的分组。

组(Generation):源节点和目的节点之间以组为单位进行通信;源节点发送的文件被划分为若干组,每个组有唯一的组号。任何转发节点只对属于同一组的分组进行编码;组中的分组数量 l 可以相同也可以不同,为简化统计,我们假定组中的分组数量相同。

编码分组(coded packet):对属于同一组的分组的线性编码,编码分组 $p' = \sum_i c_i p_i$,其中 p_i 是来自同一个组的原始分组, c_i 是一个随机数,我们将 c_i 的取值范围设置为 $[0, 255]$,即每个系数占两字节空间。线性网络编码执行伽罗华域的运算,即异或运算。注意到,虽然中间节点的编码分组产生于本节点已经收到的来自于同一个组的编码分组的线性组合,但本质上,它们依然是原始分组的线性组合。如, $p_j' = \sum_i c_{ji} p_i$ 是一个编码分组,其中 p_i 是原始分组。从 p_j' 生成新的编码分组 $p' = \sum_j r_j p_j'$,其中 r_j 是新的随机数,则 $p' = \sum_j r_j \sum_i c_{ji} p_i = \sum_i (\sum_j r_j c_{ji}) p_i$,即 p' 依然是原始分组的线性编码。

新分组:如果一个分组与节点上已经接收的所有其他该组的分组线性无关,则称这个分组是一个“新分组”。

队列长度:每个节点 i 为来自同一个流 c 的每个组 g 保存一个队列长度 q_{ig} ,用于记录节点 i 可以为 c 的组 g 发送的最大分组个数。可用于背压路由选择。

发送阈值:只有当中间节点接收的某个组的新分组数量大于一定的阈值时才允许该节点转发该组的数据。

下游节点:一个节点的邻居节点中,到目的节点的最短路径长度小于该节点的节点。

上游节点:一个节点的邻居节点中,到目的节点的最短路径长度大于等于该节点的节点。

3.2 工作流程描述

本节将描述算法的工作流程。不同于 COPE 协议的伪广播,本协议中的数据分组和确认分组均以广播的形式发送,因此需要由算法实现点到点的可靠传输。

图 1 给出了发送端的控制流程。每经过一个发送周期,

即数据分组的发送时间,源节点根据式(15)更新发送速率。每经过一个发送周期,节点都要检查本节点上缓存的组,选出满足条件的组进行发送;如果存在满足条件的组,则发送该组,并将对应队列长度减 1。

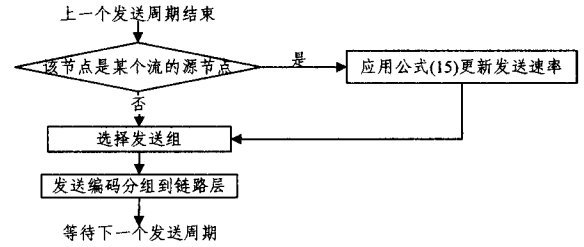


图 1 发送端

图 2 给出了接收端接收数据分组的流程。节点接收到一个数据分组时,首先检查该分组是不是一个有效分组,有效分组满足条件:(a)该分组所在组的组号大于等于节点上缓存的其所在流的最小组号;(b)该分组是一个新分组。如果该分组是一个有效分组,则保留该分组,并考查发送节点是不是本节点的上游节点,如果该节点是本节点的上游节点,则将该组的队列长度加 1。如果接收端是目的节点,检查节点是否已经接收该组的所有分组,如果是,则将自己的接收窗口向前滑动,并立刻沿最短路径向源节点发送更新分组;源节点接收到更新分组后将发送窗口向前移动,其中条件(a)可以作为邻居节点之间的同步信号,即邻居节点的最小组号应该保持一致。

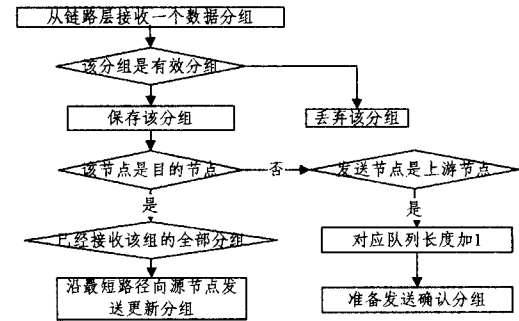


图 2 接收端接收数据分组的流程

发送端的计算量主要集中在组的选择上,即邻居节点的最小组号应该保持一致。

3.3 算法复杂度分析

发送端的计算量主要集中在组的选择上,其选择流程描述如下:(a)依次检查节点上各个流,筛选线性无关分组个数大于等于阈值的组;(b)存在某个下游节点,该下游节点还没有获得该组的所有分组;(c)根据式(16)从满足以上两个条件的组中选出所在流中具有最小组号的组。算法复杂度为 $O(|c| \cdot |n_d| \cdot |w|)$,其中 $|c|$ 表示节点上流的数目, $|n_d|$ 表示下游节点的个数, $|w|$ 表示发送窗口的大小。

接收端的计算量主要集中在新分组的判断上,实现复杂度为 $O(l \cdot (l+k))$,其中 l 是组中分组的个数, k 是每个分组的长度。我们使用渐近高斯约旦消元法,将解码时间分解到分组的接收过程中,从而大大提高了计算效率,具体实现见文献[2]。

4 仿真实验

本节将对本协议与 MORE 协议和文献[4]提出的无控制多路径路由算法进行比较。可以看出,本算法和 MORE 协议

的性能均高于无控制多路径路由算法。

由于文献[4]中的算法和 MORE 协议均没有将流量控制与路由算法相结合,每个转发节点应用先来先服务的规则为节点上的流提供服务。

我们应用效用函数 $U_c(f_c) = \log(f_c)$ 表示比例公平策略, $U_c(f_c) \propto 1/f_c$ 接近 TCP 的效用。这是一个由许多 TCP 版本实现的基于窗口的流控制机制的现实模型,可以依据网络的拥塞反馈使端到端速率呈现加性增、乘性减的特点。每经过一个发送周期,即数据分组的发送时间,源节点根据式(15)更新发送速率 f_c 。

实验采用 NS2 来模拟协议性能。在 $300m \times 300m$ 的面积上随机分布 30 个节点,链路层协议为 802.11 的 DCF(Distributed Coordination Function),信道带宽为 4Mbps,点到点丢失率约为 0.4,每组含有 32 个分组,分组长度为 1000 字节,运行时间为 20s。

基于 ETX 计算每个节点到目的节点的最短路径长度。假定每个源节点都有大量的分组需要发送。

图 3 给出了多路径路由协议和 MORE 协议在邻居节点个数平均为 8、任意选取的 8 个流上,端到端吞吐量的比较。由这个例子可以看出,在稠密网络上,结合流量优化分配算法的多路径路由的吞吐量要远大于 MORE 协议;计算两种算法的效用差值 $\sum_{c \in C} \log(f_c^{Multi}) - \sum_{c \in C} \log(f_c^{More}) = 3.58$,即本文的算法中各个流的速率分配更均匀。

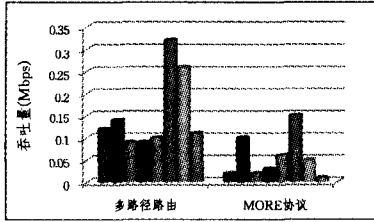


图 3 多路径路由算法与 MORE 协议端到端吞吐量的比较

图 4 给出了多路径路由协议和 MORE 协议及无控制多路径算法在邻居节点个数平均为 8 的网络上,运行 50 次、每次任意选取 8 个流的条件下,重传次数的比率。其中重传次数的计算公式为:网络上所有节点发送分组的总数/接收节点接收的分组数,即目的节点平均每接收一个节点,网络上需要发送的分组数,反映了网络的整体开销。其中本算法与 MORE 协议的比较中,比率大于 3 的运行次数在半数以上;而在与无控制多路径路由的算法比较中,比率大于 3 的运行次数多达 35 次。从而可以看出,本协议在网络整体开销上优于 MORE 协议,且两个算法均优于无控制多路由算法。

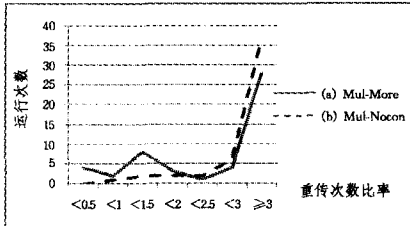


图 4 多路径路由算法与 MORE 协议和无控制多路径算法重传次数的比较

图 5 给出了多路径路由协议和 MORE 协议及无控制多

路由算法在邻居节点个数平均为 8 的网络上,运行 50 次、每次任意选取 8 个流的条件下的效用比较 $\sum_{c \in C} \log(f_c^{Multi}) - \sum_{c \in C} \log(f_c^{More})$ 。其中本算法与 MORE 协议的比较中,差值大于 4 的运行次数在半数以上;而在与无控制多路径路由的算法比较中,差值大于 4 的运行次数多达 40 次。由文献[6]的效用函数证明可以看出,本协议比 MORE 协议能够更好地实现带宽资源在各个流之间的分配,且两个算法均优于无控制多路由算法。

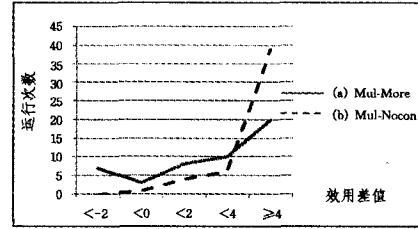


图 5 多路径路由算法与 MORE 协议的效用差值 $\sum_{c \in C} \log(f_c^{Multi}) - \sum_{c \in C} \log(f_c^{More})$ 和无控制多路径路由算法的效用差值 $\sum_{c \in C} \log(f_c^{Multi}) - \sum_{c \in C} \log(f_c^{Nocon})$

结束语 由于无线网络的广播特性,一个节点发出的信号有可能被多个节点接收,同时一个节点也可能从多个节点接收到来自同一信号源的信息。

本文提出了一个适用于无线网络的效用优化模型,旨在最大化网络吞吐量,并在各个流之间均衡地分配资源。基于该优化模型,提出一个分布的、独立于 MAC 层节点调度的启发式算法和一种有效的流抑制策略,以实际发送的分组数指引流量分配,建立了一个实用的单播多路由选择协议。性能分析与仿真实验表明,算法在多个流的条件下,可以取得比同类协议更大的吞吐量、更低的网络整体开销和更合理的带宽资源的分配。

参考文献

- [1] Biswas S, Morris R. Opportunistic routing in multihop wireless networks[J]. ACM SIGCOMM Computer Communication Review, 2004, 34(1): 69-74
- [2] Han Li, Qian Huan-yan. Parallelized Network Coding With SIMD Instruction Sets[C]// International Symposium on Computer Science and Computational Technology, 2008 (ISCSCT'08). 2008
- [3] Chachulski S, Jennings M, Katti S, et al. Trading Structure for Randomness in Wireless Opportunistic Routing[J]. ACM SIGCOMM Computer Communication Review, 2007, 37(4): 169-180
- [4] Sun Bao-lin, Song Ying, Gui Chao, et al. Performance of Network Coding Based Multipath Routing in Wireless Sensor Networks[J]. IJCSI International Journal of Computer Science Issues, 2012, 9(6)
- [5] Eryilmaz A, Srikant R. Joint congestion control, routing and mac for stability and fairness in wireless networks[J]. IEEE Journal on Selected Areas in Communications, 2006, 24(8): 1514-1524
- [6] Srikant R. The Mathematics of Internet Congestion Control [M]. Birkhauser, 2003