

# 一种基于流内与流间网络编码的无线路由算法

田贤忠,周前伟

(浙江工业大学计算机科学与技术学院,浙江杭州 310023)

**摘要:** 网络编码能极大提高网络吞吐量和可靠性. 该文提出了一种基于流内网络编码与流间网络编码的无线路由算法, 首先用改进的流间网络编码策略寻找一条固定路由, 发现尽可能多的编码机会以减少传输次数; 再用流内网络编码与局部机会路由结合的方法实施每跳的数据包传输, 减少数据包的重传次数. 理论分析和仿真实验表明, 此算法比传统的编码感知路由算法具有更高的吞吐量和可靠性.

**关键词:** 无线网络; 流内网络编码; 流间网络编码; 局部机会路由

**中图分类号:** TP393 **文献标识码:** A **文章编号:** 0372-2112 (2013)02-0395-07

**电子学报 URL:** <http://www.ejournal.org.cn> **DOI:** 10.3969/j.issn.0372-2112.2013.02.029

## An Algorithm of Wireless Routing Based on Intra-Flow and Inter-Flow Network Coding

TIAN Xian-zhong, ZHOU Qian-wei

(School of Computer Science and Technology, Zhejiang University of Technology, Hangzhou, Zhejiang 310023, China)

**Abstract:** Network coding can greatly improve network throughput and reliability. This paper presents an algorithm of wireless routing based on intra-flow and inter-flow network coding. First, find a fixed routing using improved inter-flow network coding, which reduce transmission number by finding more coding opportunity. Then, transmit data package at every hop using intra-flow network coding, reduce data retransmission number. It is showed by theory analysis and emulation test that our algorithm obtains higher throughput and more reliability than traditional coding-aware routing.

**Key words:** wireless networks; intra-flow network coding; inter-flow network coding; local opportunistic routing

### 1 引言

无线网络中, 由于无线链路质量差, 容易丢包, 从而影响了网络的吞吐率和可靠性. 所以, 如何同时提高无线路由的吞吐率和可靠性是一个非常重要的研究问题.

网络编码<sup>[1~3]</sup>已被证明是一种能极大提高网络吞吐量和可靠性的新技术. 网络编码可分两类: 流间网络编码和流内网络编码. 流间网络编码(Inter-flow Network Coding)是指网络节点把来自不同流的数据包编码组合在一起, 然后广播发送出去, 从而提高了数据包单次传输的信息量, 减少数据传输次数, 提高网络吞吐量. 流内网络编码(Intra-flow Network Coding)是指网络节点把要传输的同一条流的数据分成大小相同的  $K$  个数据包, 然后编码组合成多个编码包发送, 目的节点只要收到  $K$  个线性无关的编码包(不需要考虑从哪条路径传输的, 也不关心中间是否丢失过编码包)就能解码出原始数据包, 从而避免了对每个收到的数据包的确认, 减少了数

据包的重传次数, 对可靠数据传输具有重要意义.

用流间网络编码提高无线网络吞吐率的研究始于 Katti 等人<sup>[4]</sup>提出的 COPE 协议, COPE 协议用机会网络编码的方法实现了两跳范围的网络编码, 提高了网络的吞吐量. Le 等人<sup>[5]</sup>和 Guo 等人<sup>[6]</sup>讨论了基于流间网络编码的无线路由. 它们通过寻找网络中的编码机会较多的路由来发送数据包, 从而减少数据包的发送次数. Kim<sup>[7]</sup>等人考虑了流间网络编码增益与速率匹配之间的相互作用, 提出了最优速率选择方法, 提高网络编码增益. Eryilmaz 等人<sup>[8]</sup>提出了在有线或无线网络中何时何地怎样实施流间网络编码的方法, 并比较了流间网络编码路由与纯路由的性能. 卢冀等人<sup>[9]</sup>提出了基于机会式网络编码的低时延广播传输算法, 而汪玉等人<sup>[10]</sup>则分析了流间网络编码中的侦听管理策略. 然而上述流间网络编码算法都没有考虑不同流之间的链路丢包情况, 本文希望利用链路的这种特性改进流间网络编码, 以提高网络的吞吐量. 对流内网络编码, Chachulski 等人<sup>[11]</sup>提

出的 MORE 协议利用流内网络编码解决了机会路由中中继节点数据发送的协作问题,它还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,从而提高网络吞吐量.卢文伟等人<sup>[12]</sup>利用流内线性网络编码提出了免确认的节能路由算法,算法利用多路径传输多个编码包,目的节点收到一定数量的编码包解码出所需的数据包,从而避免了复杂的确认重传.Radunovic 等人<sup>[13]</sup>针对无线 Mesh 网络,利用流内网络编码提出了一种基于网络效能最大化的机会路由策略,从而得到了最优的流控制调度与速率分配策略.Soldo 等人<sup>[14]</sup>则考虑了用流内网络编码解决无线 Mesh 网络中多路径机会路由优化和拥塞控制问题.然而,上述这些流内网络编码算法大都与流间网络编码分离的,如何结合流间网络编码与流内网络编码各自的特点,充分发挥它们的优势以提高无线网络的吞吐量与可靠性是一个值得研究的重要问题.

基于流间网络编码的无线路由通过定义一种反映网络编码机会的度量,在网络中寻找一条尽可能多的编码机会的路由,使路由具有更高的网络吞吐量.在实际的传输过程中,由于无线链路存在丢包现象,所以这一条路由发送数据包时每跳都可能丢包,必须数据包重传.如果这时采用流内网络编码进行传输,并利用固定路由外的辅助节点协助重传,就可以减少数据包重传次数.因为下一跳节点只需收到一定数量的编码包就可以解码出所需的数据包,而并不关心数据包是从什么路径传输过来的.本文利用这种思想,提出了一种基于流内与流间网络编码的无线路由算法( $I^2$ NCR: Intra-flow and Inter-flow Network Coding Based Wireless Routing),具体包括:(1)提出了一种改进的流间网络编码策略 INCS(Improved Inter-flow Network Coding Scheme),用 INCS 提出了一种编码感知固定路由的算法;(2)提出了一种应用流内网络编码的局部机会路由方法 LOR(Local Opportunistic Routing),实现路由的每一跳数据传输;(3)理论分析了 INCS 和 LOR 方法的性能;(4)完整提出了一种基于流内与流间网络编码的无线路由算法  $I^2$ NCR.

## 2 算法基本思想

一般的无线路由发现过程是这样的:源节点向邻居节点广播路由请求包 RREQ(Route Request),中间节点向邻居转发 RREQ,目的节点在收到 RREQ 之后向源节点返回 RREP(Route Reply),源节点在收到 RREP 之后,选择一条路径.在编码感知的无线路由中,返回的 RREP 中带有路径中节点是否可编码的信息,所以最终选择的路径充分考虑了网络编码的增益.然而这样选择的固定路由有一个问题,即没有充分考虑无线信道

的广播特性.如图 1 所示,用编码感知路由算法找到了一条最佳路径  $v_1 \rightarrow v_2 \rightarrow v_3 \rightarrow v_4$ .在数据传输过程中,如果  $v_2$  向  $v_3$  发送数据包时发生丢包,则  $v_2$  需要重传.但是,由于无线信道具有广播特性, $v_2$  向  $v_3$  发送数据包时,其距离节点  $v_3$  较近的邻居节点  $v_a$  和  $v_b$  可能也收到了.这时用  $v_a$  或  $v_b$  重传会比用  $v_2$  重传效果更佳.

$I^2$ NCR 的基本思想是在编码感知路由的基础上,充分利用无线信道的广播特性,结合流内网络编码,用局部机会路由的方法传输数据,以提高路由的吞吐量.算法步骤如下:

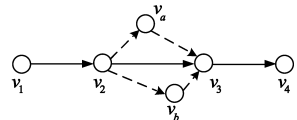


图1  $I^2$ NCR原理图

(1)首先用改进的流间网络编码策略(INCS)建立一条源节点到目的节点的固定路由.

(2)源节点把要发送的数据分成大小相等的数据段,每段又分成  $K$  个数据包, $K$  个数据包用线性网络编码组成多个编码包.然后用局部机会路由(LOR: Local Opportunistic Routing)方法向固定路径中的下一跳节点连续广播发送,直到收到下一跳节点发送过来的已收到  $K$  个线性无关的编码包的确认信息.源节点发送下一段的编码包.

(3)固定路由中的下一跳节点收到编码包后,与之前已收到的编码包再次编码.然后再用 LOR 方法向下一跳节点发送.直到其下一跳节点已收到本段的  $K$  个编码包,再发送下一段编码包.

(4)目的节点收到本段的  $K$  个编码包后,解码出本段的  $K$  个原始数据包.同时向上一跳节点发送一个确认信息.上一跳节点收到后可以发送下一段编码包.

这里定义两个概念:

(1)主节点:是指用 INCS 建立的固定路由中的网络节点.

(2)辅助节点:是指网络中的非主节点,用于 LOR 中协助发送数据.

## 3 算法具体实现

从上面的分析可以看出, $I^2$ NCR 需要解决以下几个问题:(1)如何用 INCS 建立一条固定路由;(2)如何用 LOR 发送每一跳数据;(3)节点收到数据包后,如何发送确认信息.

### 3.1 固定路由的建立

(1)改进的网络编码策略(INCS: Improved Network Coding Scheme)

文献[5]给出了通过节点的两条流可以编码的条件(简称“编码条件”):设  $U(X, f)$  表示流  $f$  的上游节点集合(即流  $f$  上源节点到节点  $X$  之间的所有节点的集合,除节点  $X$  外), $D(X, f)$  表示流  $f$  的下游节点集合

(即流  $f$  上目的节点到节点  $X$  之间的所有节点的集合, 除节点  $X$  外),  $N(X)$  表示节点  $X$  的一跳邻居节点集, 如果两条流  $f_1$  和  $f_2$  交汇于节点  $X$  (即节点  $X$  是流  $f_1$  和  $f_2$  的公共节点), 则流  $f_1$  和  $f_2$  可以在节点  $X$  进行编码, 并且编码包可以被解码的充分必要条件:

(a) 存在  $V_1 \in D(X, f_1)$ , 满足:  $V_1 \in N(V_2)$ ,  $V_2 \in U(X, f_2)$ ; 或者  $V_1 \in U(X, f_2)$ ;

(b) 存在  $V_2 \in D(X, f_2)$ , 满足:  $V_2 \in N(V_1)$ ,  $V_1 \in U(X, f_1)$ ; 或者  $V_2 \in U(X, f_1)$ .

如图 2 所示, 源节点  $S_1$  和  $S_2$  经过节点  $X$  分别向目的节点  $D_1$  和  $D_2$  发送数据包. 同时, 目的节点  $D_1$  和  $D_2$  分别能侦听到源节点  $S_2$  和  $S_1$  发送的数据包. 根据“编码条件”, 中继节点  $X$  能对来自源节点  $S_1$  和  $S_2$  的数据包进行编码, 目的节点  $D_1$  和  $D_2$  收到编码包后, 根据之前已侦听到的数据包解码出各自所需要的数据包.

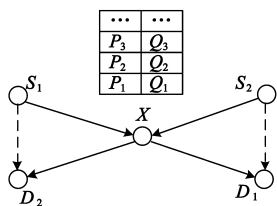


图2 编码条件示意图

然而这样的网络编码策略并不一定带来增益. 假设链路  $XD_1$  和  $XD_2$  的丢包率分别为  $p_1 = 0.1$  和  $p_2 = 0.9$ , 其它链路丢包率为 0. 同时假设中继节点缓冲区中始终有来自源节点  $S_1$  和  $S_2$  的数据包, 以  $P_1, P_2, \dots, P_n$  表示来自源节点  $S_1$  的数据包 (简称“ $P$  包”), 以  $Q_1, Q_2, \dots, Q_m$  表示来自源节点  $S_2$  的数据包 (简称“ $Q$  包”).

(1) 如果采用非网络编码策略 (NNCS: Non-Network Coding Scheme), 中继节点  $X$  发送  $P$  包和  $Q$  包各 10 次, 共 20 次, 则目的节点  $D_1$  平均收到 9 个数据包,  $D_2$  平均收到 1 个数据包, 共 10 个数据包. (2) 如果用网络编码 (这里称原始网络编码策略 ONCS: Original Network Coding Scheme), 中继节点对  $P$  包和  $Q$  包网络编码后再广播发送, 如  $P_1 \oplus Q_1, P_2 \oplus Q_2, \dots, P_n \oplus Q_n$ . 为了与 NNCS 进行比较, 这里也发送 20 次. 根据文献 [15] 可知, 要使目的节点  $D_1$  和  $D_2$  都收到一个编码包, 中继节点  $X$  必须发送的次数为:  $1/(1-p_1) + 1/(1-p_2) - 1/(1-p_1p_2) = 1/(1-0.1) + 1/(1-0.9) - 1/(1-0.1 \times 0.9) \approx 10$ , 故发送 20 次目的节点大约可接收 2 个编码包,  $D_1$  和  $D_2$  共可解码出 4 个原始数据包. 这比 NNCS 共接收到 10 个数据包效率更差.

究其原因, 中继节点  $X$  发送编码包时, 必须使两个目的节点  $D_1$  和  $D_2$  都收到后才发送下一个编码包. 由于链路  $XD_1$  和  $XD_2$  存在丢包, 目的节点  $D_1$  收到编码包

时,  $D_2$  如果没有收到, 中继节点  $X$  必须重传这个编码包. 而这时, 这个编码包对  $D_1$  来说是无用的, 这就造成巨大的浪费. 我们改变其编码策略, 如果  $D_1$  收到编码包而  $D_2$  没有收到编码包时, 中继节点  $X$  用下一个  $P$  包与原来的  $Q$  包编码成新的编码包再广播发送. 这样  $D_1$  再次收到的编码包中包含下一个  $P$  包的信息, 提高了吞吐率. 如表 1 所示, 第 1 次中继节点发送编码包  $P_1 \oplus Q_1$  后, 如果  $D_1$  收到编码包,  $D_2$  没有收到, 则第 2 次中继节点用  $P_2$  与  $Q_1$  编码, 然后发送. 同样, 第 2 次中继节点发送编码包  $P_2 \oplus Q_1$  后, 如果  $D_1$  没有收到编码包, 而  $D_2$  收到了, 则第 3 次中继节点用  $Q_2$  包与  $P_2$  编码, 然后发送编码包  $P_2 \oplus Q_2$ . 以此类推. 我们称这种编码策略为改进的网络编码策略 (INCS: Improved Network Coding Scheme).

表 1 编码优化策略示意图

序号	1	2	3	4	5	6
用于编码的 $P$ 包	$P_1$	$P_2$	$P_2$	$P_3$	$P_3$	$P_3$
用于编码的 $Q$ 包	$Q_1$	$Q_1$	$Q_2$	$Q_2$	$Q_2$	$Q_3$
$D_1$ 是否收到编码包	✓	×	✓	×	×	✓
$D_2$ 是否收到编码包	×	✓	×	×	✓	✓

按 INCS 编码成的数据包, 对目的节点来说都是有用的, 换句话说, 目的节点每次收到的编码包都能解码出新的原始数据包, 而不是重复的数据包. 这样, 上例中如果改用 INCS 策略, 发送 20 次编码包, 则目的节点  $D_1$  平均收到 18 个编码包, 从而能解码出 18 个原始数据包,  $D_2$  平均收到 2 个数据包, 能解码出 2 个不同的原始数据包, 共 20 个数据包. 比 NNCS 和 ONCS 两种方法效果都好.

### (2) 编码感知路由度量与路由建立过程

设路径  $p: v_0 \rightarrow v_1 \rightarrow \dots \rightarrow v_n$ , 其中,  $v_i \rightarrow v_{i+1}$  ( $i = 0, \dots, n-1$ ) 是其中的一条链路, 其丢包率为  $p_i$ , 定义链路  $v_i \rightarrow v_{i+1}$  ( $i = 0, \dots, n-1$ ) 的度量  $M_i$  为其传输一个数据包的期望传输次数 (ETX: Expected Transmission Count), 即:

$$M_i = \begin{cases} 1/(1-p_i), & v_i \text{ 不能编码} \\ 0, & v_i \text{ 能编码} \end{cases} \quad (1)$$

度量  $M_i$  反映了节点  $v_i$  传输一个数据包平均的传输次数, 不能网络编码时, 其传输次数是其成功传输率的倒数, 即  $1/(1-p_i)$ ; 可以网络编码时, 这条链路上传输的数据可以看成被其它链路上的数据编码后稍带走了 (这里我们用 INCS 进行网络编码), 故其值为 0.

整条路径  $p$  的度量可看成其中的各条链路的度量之和, 故定义路径  $p$  的度量为:

$$M(p) = \sum_{i=0}^{n-1} M_i \quad (2)$$

设源节点与目的节点之间有多条路径, 路径的集

合记为 PSet. 我们要在其中选择  $M(p)$  值最小的路径, 即

$$\operatorname{argmin}_{p \in \text{PSet}} \{M(p)\} \quad (3)$$

路由的建立过程与文献[5]中的 DCAR 路由过程类似. 源节点首先广播 RREQ 包, RREQ 包中除了常用的字段外, 还包括路径经过的上游节点集 UpSet、上游节点一跳邻居节点集 UpNbSet 等. 中继节点收到 RREQ 包后, 记下相关信息, 并用本节点的信息更新 RREQ 包, 继续广播. 目的节点收到 RREQ 包后, 记下相关信息, 并逆向单播 RREP 包. RREP 包中除了常用的字段外, 还包括路径经过的下游节点集 DnSet、下游节点一跳邻居节点集 DnNbSet, 以及路径度量  $M(p)$ . 中继节点收到 RREP 包后, 记下相关信息, 根据 UpSet、UpNbSet、DnSet 以及 DnNbSet 信息判断是否可以编码, 并根据式(1)计算出本节点的度量  $M_i$ , 然后更新 RREP 包, 其中  $M(p) = M(p) + M_i$ . 当源节点收到不同的多个 RREP 包后, 选择  $M(p)$  值最小的路径作为最终选择的路径.

### 3.2 局部机会路由 (LOR: Local Opportunistic Routing)

一旦建立了一条编码感知的固定路由后, 数据包可以沿这条选定的路径发送. 在每一跳传输过程中, 分两种情况: (1) 如果发送节点是流间网络编码节点, 则用 INCS 策略编码后直接在一跳范围内的主节点之间广播编码包; (2) 如果发送节点不是编码节点, 我们采用 LOR 方法传输.

如图 3 所示, 主节点  $v_i$  和  $v_j$  是选定的固定路由中的一跳邻居节点,  $v_a$  和  $v_b$  是两个辅助节点, 它们到主节点  $v_j$  的 ETX 值小于  $v_i$  到  $v_j$  的 ETX 值. 节点  $v_i$  不断地向其下一跳节点  $v_j$  广播

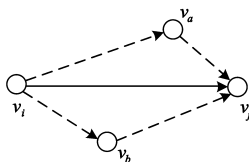


图3 LOR示意图

编码包, 这里的编码包是由同一条流的同一段数据包编码而成的, 换句话说就是流内网络编码. 辅助节点  $v_a$  和  $v_b$  如果收到主节点  $v_i$  发给  $v_j$  的编码包, 与之前已收到的编码包编码, 然后也向主节点  $v_j$  发送编码包. 主节点  $v_j$  收到同一段的  $K$  个线性无关的编码包后, 给主节点  $v_i$  发送一个确认信息. 主节点  $v_i$  收到确认信息后发送下一数据段的编码包.

为了避免主节点  $v_j$  收到过多的线性相关的编码包, 我们对辅助节点发送编码包进行适当控制. 我们希望当主节点  $v_j$  收到  $v_i$  发送的编码包时, 辅助节点就不转发了. 当  $v_j$  没有收到  $v_i$  发送的编码包时, 只有一个辅助节点  $v_k$  转发, 这样能保证主节点  $v_j$  收到的编码包线性无关. 根据文献[10], 设  $p_{ik}$  表示节点  $v_i$  到节点  $v_k$  的丢包率,  $z_i$  表示源节点  $v_i$  发送数据包的次数, 因为源节点

$v_i$  发送的数据包必须至少有一个中继节点或目的节点收到, 故  $z_i$  值为  $1/(1 - \prod_{k < i} p_{ik})$ , 其中  $k < i$  表示节点  $v_k$  到目的节点的 ETX 值比节点  $v_i$  的小, 即  $v_k$  离目的节点更近些. 节点  $v_k$  收到编码包的个数为  $z_i(1 - p_{ik})$ . 我们对节点  $v_i, v_j$  以及辅助节点按其到达  $v_j$  的 ETX 值从小到大排序. 如果  $v_k$  的 ETX 值最小的, 则优先发送 ( $v_j$  的 ETX 值为 0, 是最小的, 表示其收到后, 其它节点不需要再发送). 因为 ETX 值小, 表示离目的节点近, 更容易被目的节点接收到. 如果不是最小的, 则只有较小的 ETX 节点没有收到时才有机会发送, 其发送概率为  $\prod_{a < k} p_{ia}$ . 节点  $v_i$  的 ETX 值最大, 只有其它节点都没有收到时才发送. 这样中继节点  $v_k$  需要传输的数据包的数目为:

$$L_k = z_i(1 - p_{ik}) \prod_{a < k} p_{ia} \quad (4)$$

另外我们希望辅助节点  $v_k$  发送的编码包主节点  $v_j$  能收到, 其概率为  $(1 - p_{kj})$ , 故节点  $v_k$  实际发送的编码包的个数为:

$$z_k = \frac{L_k}{(1 - p_{kj})} \quad (5)$$

考虑辅助节点  $v_k$  每收到一个编码包时需发送的编码包个数.  $z_k$  是源节点发送  $z_i$  个编码包时, 节点  $v_k$  实际需发送的编码包个数, 这时节点  $v_k$  收到的编码包实际只有  $z_i(1 - p_{ik})$  个, 故由公式(4)和(5)可得节点  $v_k$  每收到一个编码包时需发送的编码包个数为:

$$n_k = \frac{z_k}{z_i(1 - p_{ik})} = \frac{\prod_{a < k} p_{ia}}{(1 - p_{kj})} \quad (6)$$

实际操作时, 在每个辅助节点  $v_k$  中设置一个计数器  $\text{Count}_k$ . 首先设  $\text{Count}_k = 0$ ; 每当辅助节点  $v_k$  收到一个编码包, 则  $\text{Count}_k = \text{Count}_k + n_k$ ; 这时, 如果  $\text{Count}_k > 0$ , 节点  $v_k$  发送一个编码包, 同时  $\text{Count}_k = \text{Count}_k - 1$ . 当节点  $v_k$  发送下一数据段时  $\text{Count}_k$  先清零.

### 3.3 确认信息发送

主节点  $v_j$  收到  $K$  个线性无关的编码包后, 必须通知其上一跳主节点  $v_i$ ,  $v_i$  接着发送下一段编码包. 这里我们不是单独发送一个确认信息 ACK, 而是在  $v_j$  向其下一跳主节点发送编码包时在编码包中捎带确认信息. 编码包的格式如图 4 所示, 序号是指节点发送第几个同一段编码包, 段号是指所发送的编码包的数据段编号. 当上一跳主节点  $v_i$  收到节点  $v_j$  发给其下一跳的编码包, 比较编码包中的“上一跳节点 ID”字段是否与自己匹配. 如果匹配, 查看“序号”字段值是否大于等于  $K$ , 如果大于  $K$ , 则节点  $v_i$  发送下一段编码包. 辅助节点收后作同样处理.

序号	段号	上一跳节点ID	下一跳节点ID	编码包
----	----	---------	---------	-----

图4 编码包格式

## 4 性能分析

本节主要对  $I^2NCR$  算法进行性能分析. 由于  $I^2NCR$  算法主要在两个方面进行了改进: (1) 流间编码采用了 INCS 策略; (2) 每跳采用 LOR 发送数据. 整条路由的性能改善 (比如发送次数的减少等) 即是每跳链路的性能改善的累积. 所以我们从 INCS 和 LOR 这两个方面的性能分析即可看出  $I^2NCR$  的性能优劣.

### 4.1 INCS 性能分析

我们这里分析单跳模型, 读者很容易把它推广到多跳情况. 如图 5 所示, 设有  $n$  条流在节点  $v$  中可以编码, 即节点  $v$  可以把来自  $n$  条流的  $n$  个数据包编码在一起, 其每个目的节点  $D_i (i=1, \dots, n)$  收到编码包后能解出各自所需的数据包. 节点  $v$  到目的节点  $D_i$  的链路丢包率设为  $p_i (i=1, \dots, n)$ .

**定理 1** 用改进的网络编码策略 (INCS), 节点  $v$  平均每发送一个编码包, 所有目的节点总共可以解出  $\sum_{i=1}^n (1-p_i)$  个数据包.

**证明** 对任意一个目的节点  $D_i$  来说, 由于其丢包率为  $p_i$ , 故节点  $v$  平均每发送  $1/(1-p_i)$  个编码包,  $D_i$  收到一个编码包. 也就是说, 节点  $v$  平均每发送 1 个编码包, 目的节点  $D_i$  收到  $(1-p_i)$  个编码包. 又因为在 INCS 编码策略中, 解码出的原始数据包都是不重复的, 故  $D_i$  可解码出  $(1-p_i)$  个不同的原始数据包.  $n$  个目的节点共可解码出所需的  $\sum_{i=1}^n (1-p_i)$  个数据包. 得证.

**推论 1** 目的节点接收同样多的数据包, INCS 比 ONCS 需更少的传输次数.

**证明** 从定理 1 证明可以看出, 节点  $v$  平均每发送一个编码包, 所有目的节点总共可以收到  $\sum_{i=1}^n (1-p_i)$  个数据包. 在 ONCS 策略中, 目的节点收到编码包可能是之前已收到过的, 所以解码出的数据包的数量要小于收到的编码包的数量, 故 ONCS 得到的原始数据包数量少于 INCS 得到的数据包, 得证.

**推论 2** 目的节点接收同样多的数据包, INCS 比 NNCS 需更少的传输次数.

**证明** 在 NNCS 策略中, 节点  $v$  需轮流给  $n$  个目的节点发送数据包 (单播), 对任意一个目的节点  $D_i$  来说, 由于其丢包率为  $p_i$ , 故节点  $v$  平均每发送 1 个编码包, 目的节点  $D_i$  收到  $(1-p_i)$  个编码包. 节点  $v$  向  $n$  个

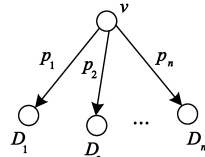


图5 多条流编码情况

目的节点轮发一遍, 需发送  $n$  次, 目的节点共收到  $\sum_{i=1}^n (1-p_i)$  个数据包. 根据定理 1, 目的节点接收相同数量的数据包, 节点  $v$  只需发送 1 次. 得证.

### 4.2 LOR 性能分析

这里我们计算 LOR 中目的节点每收到一个数据包需要传输的次数, 并与传统的固定路由 (FXR: Fixed Routing) 作比较. 如图 3 所示, 节点  $v_i$  是源节点,  $v_j$  是目的节点, 计算节点  $v_j$  平均收到 1 个数据包节点  $v_i$  与辅助节点共需传输多少次.

**定理 2** LOR 中目的节点平均收到一个数据包需要传输的次数为:

$$N_{LOR} = \frac{1}{1 - \prod_{k < i} p_{ik}} \left( 1 + \sum_{j < k < i} \frac{(1 - p_{ik}) \prod_{a < k} p_{ia}}{1 - p_{kj}} \right) \quad (7)$$

其中,  $p_{xy}$  表示链路  $v_x \rightarrow v_y$  的丢包率,  $x < y$  表示节点  $v_x$  到目的节点  $v_j$  的 ETX 值小于节点  $v_y$ .

**证明** 由 LOR 算法可知, 源节点  $v_i$  发送次数为:

$$z_i = \frac{1}{1 - \prod_{k < i} p_{ik}} \quad (8)$$

另根据式 (4) 和 (5), 辅助节点  $v_k$  实际发送的次数为:

$$z_k = \frac{z_i (1 - p_{ik}) \prod_{a < k} p_{ia}}{1 - p_{kj}} \quad (9)$$

辅助节点与源节点共发送次数为:

$$N_{LOR} = z_i + \sum_{j < k < i} z_k \quad (10)$$

由式 (8)、(9) 和 (10) 可得式 (7). 得证.

**推论 3** LOR 比 FXR 具有更少的发送次数.

**证明** 在 FXR 中, 节点  $v_j$  是平均收到 1 个数据包节点  $v_i$  需发送  $1/(1-p_{ij})$  次. 在 LOR 中, 辅助节点的传输概率为源节点发送 1 个数据包时目的节点没有收到的概率减去目的节点与辅助节点都没有收到的概率, 即:

$$\sum_{j < k < i} (1 - p_{ik}) \prod_{a < k} p_{ia} = p_{ij} - \prod_{k < i} p_{ik}$$

又因为辅助节点  $v_k$  的 ETX 值小于源节点  $v_i$ , 故  $p_{ij} > p_{kj}$ . 所以

$$\begin{aligned} N_{LOR} &< \frac{1}{1 - \prod_{k < i} p_{ik}} \left( 1 + \sum_{j < k < i} \frac{(1 - p_{ik}) \prod_{a < k} p_{ia}}{1 - p_{kj}} \right) \\ &= \frac{1}{1 - \prod_{k < i} p_{ik}} \left( 1 + \frac{p_{ij} - \prod_{k < i} p_{ik}}{1 - p_{kj}} \right) = \frac{1}{1 - p_{ij}} \end{aligned}$$

得证.

## 5 仿真实验

我们用 MATLAB 仿真工具进行仿真实验. 首先用

改进的网络编码策略 INCS 与 ONCS、NNCS 策略进行对比实验. 如图 5 所示, 假设中继节点发送编码包给目的节点, 目的节点都能解码. 同时假设中继节点  $v$  每次发送 1 个编码包, 发送次数总共为 100 次. 当设置每条链路的丢包率为 0.3, 图 6 表示不同的目的节点数与目的节点收到的总的数据包数之间的关系. 图 7 表示当目的节点数目为 2 时, 每条链路的丢包率从 0.1 到 0.9 变化与目的节点收到的数据包数目之间的关系. 从这两个图中可以看出: (1) INCS 策略明显比其它两种策略收到的数据包数目要多, 而且收到的数据包数目基本符合定理 1 的结果. 比如, 图 6 中 10 个目的节点时, 因为丢包率为 0.3, 根据定理 1 平均每发送一次收到的数据包数为  $10 \times (1 - 0.3) = 7$ , 故发送 100 次应该收到 700 个数据包, 与图中结果基本一致. (2) INCS 策略收到的数据包数目随目的节点的个数增加而增大, 而 NNCS 则几乎没有变化. 这是因为目的节点个数多表示编码包中包含的原始数据包多, 所以采用 INCS 一次发送包含的数据包就多, 而 NNCS 是不采用网络编码的, 所以几乎没有影响.

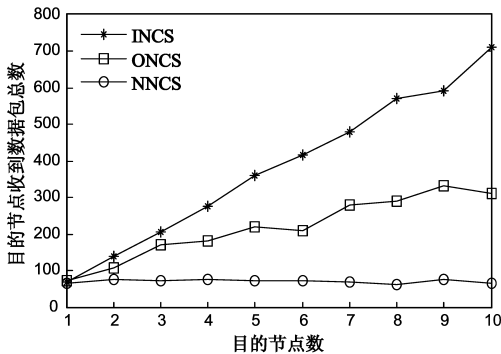


图6 目的节点数与收到数据包数目

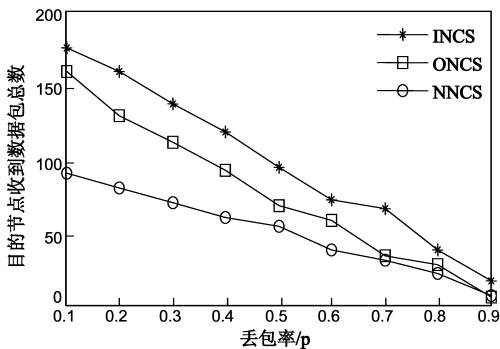


图7 丢包率与收到数据包数目

其次, 我们比较 LOR 与 FXR 性能, 比较目的节点收到一个数据包时, 两种路由策略各需传输多少次. 我们设置有 3 个中继节点, 源节点与目的节点的一跳链路丢包率从 0 到 0.5 之间变化, 中继节点到源节点和目的节

点的丢包率随机产生, 但小于源节点到目的节点的丢包率. 图 8 中结果是运行 1000 次取平均值得到的, 从图中我们可以看出 LOR 算法的传输次数明显小于 FXR 算法. 另外, LOR 算法随源节点到目的节点的丢包率的影响没有 FXR 算法明显, 它更稳定.

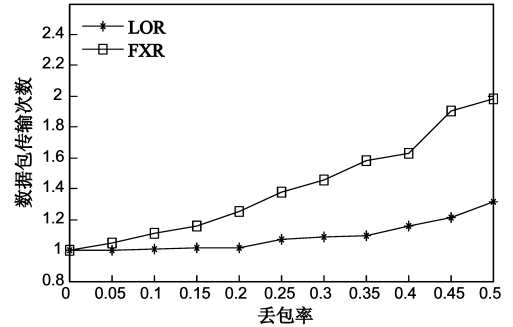


图8 LOR算法与FXR算法性能比较图

## 6 小结

本文提出一种基于流内网络编码与流间网络编码的无线路由算法, 用改进的流间网络编码策略先建立一条固定路由, 用局部机会路由和流内网络编码实现每跳路由的数据传输. 本算法充分利用了流间网络编码能提高网络吞吐量和流内网络编码能减少重传次数的特点, 能保证可靠传输的前提下, 减少数据传输次数. 理论分析和仿真实验都充分证明了这一点.

## 参考文献

- [1] Ahlswede R, Cai N, Li S Y R, et al. Network information flow [J]. IEEE Transactions on Information Theory, 2000, 46(4): 1204 - 1216.
- [2] Li S Y R, Yeung R W, Cai N. Linear network coding [J]. IEEE Transactions on Information Theory, 2003, 49(2): 371 - 381.
- [3] Ho T, Medard M, Koetter R, et al. A random linear network coding approach to multicast [J]. IEEE Transaction on Information Theory, 2002, 48(2): 359 - 383.
- [4] Katti S, Rahul H, Hu W, Katabi D, Médard M. XORs in the air: practical wireless network coding [J]. IEEE/ACM Transactions on Networking, 2008, 16(3): 487 - 510.
- [5] Le J, Lui J C S, Chiu D M. DCAR: Distributed coding-aware routing in wireless networks [J]. IEEE Transaction on Mobile Computing, 2010, 9(4): 596 - 608.
- [6] Guo B, Li H, Zhou C, Cheng Y. Analysis of general network coding conditions and design of a free-ride oriented routing metric [J]. IEEE Transactions on Vehicular Technology, 2011, 60(4): 1714 - 1727.
- [7] Kim Y, De Veciana G. Is rate adaptation beneficial for inter-session network coding? [J]. IEEE Journal on Selected Areas

- in Communications. 2009, 27(5): 635 – 646.
- [8] Eryilmaz A, Lun D, Swapna B. Control of multi-hop communication networks for inter-session network coding [J]. IEEE Transactions on Information Theory, 2011, 57 (2): 1092 – 1110.
- [9] 卢冀,肖嵩,吴成柯,等.基于机会式网络编码的低时延广播传输算法[J].电子学报,2011,39(5):1214 – 1219.  
J Lu, S Xiao, C K Wu. Opportunistic network coding based delay-sensitive broadcast transmission algorithm [J]. Acta Electronica Sinica, 2011, 39(5): 1214 – 1219. (in Chinese)
- [10] 汪玉,卢汉成,洪佩琳,等.网络编码中的侦听管理策略[J].电子学报,2012,40(1):47 – 52.  
Y Wang, H C Lu, P L Hong, et al. Overhearing management policies on network coding [J]. Acta Electronica Sinica, 2012, 40(1): 47 – 52. (in Chinese)
- [11] Chachulski S, Jennings M, Katti S, et al. Trading structure for randomness in wireless opportunistic routing [J]. ACM SIGCOMM Computer Communication Review, 2007, 37(10): 169 – 180.
- [12] 卢文伟,朱艺华,陈贵海.无线传感器网络中基于线性网络编码的节能路由算法[J].电子学报,2010,38(10): 2309 – 2314.  
W W Lu, Y H Zhu, G H Chen. Energy-efficient routing algorithms based on linear network coding in wireless sensor networks [J]. Acta Electronica Sinica, 2010, 38 (10): 2309 – 2314. (in Chinese)
- [13] Radunovic B, Gkantsidis C, Key P, Rodriguez P. Toward practical opportunistic routing with intra-session network coding for mesh networks [J]. IEEE/ACM Transactions on Networking, 2010, 18(2): 420 – 433.
- [14] Soldo F, Markopoulou A, Toledo A. A simple optimization model for wireless opportunistic routing with intra-session network coding [A]. 2010 IEEE International Symposium on Network Coding (NetCod) [C]. Toronto: IEEE Computer Society, 2010, 1 – 6.
- [15] Nguyen D, Tran T, Nguyen T, Bose B. Wireless broadcast using network coding [J]. IEEE Transactions on Vehicular Technology, 2009, 58(2): 914 – 925.

### 作者简介



田贤忠 男,1968年出生于杭州,浙江工业大学副教授,博士研究生,研究方向为网络编码、移动计算、无线网络的协议、算法、性能分析与优化等。

E-mail: txz@zjut.edu.cn

周前伟 男,1985年出生于湖南,浙江工业大学在读硕士研究生,研究方向为无线网络、网络编码。

E-mail: 443424748@qq.com