

# 无线多跳网络上基于网络编码的多源多播算法

韩 莉<sup>1,2</sup> 钱焕延<sup>2</sup> 刘慧婷<sup>1</sup>

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

**摘 要** 考虑到在无线多跳网络中存在多个相互独立的多播流的调度问题,首先提出了一个基于网络编码的动态多源多播模型,并根据该模型设计了一个适用于 802.11DCF MAC 的多播算法 MulSrc。MulSrc 以队列积压作为流调度的主要依据,以网络编码作为数据传输和本地恢复的主要手段,无固定的传输路径,在无线网络中实现以丰富的多路径来克服相关丢失、降低延迟的目标,适用于无线网络中大量存在的随机丢包情况。实验证明,在存在多个多播流的网络上,MulSrc 可以取得比同类协议 CodeCast 和 MMForests 更好的性能。

**关键词** 网络编码,多个流,背压路由,可行流速率,多播

**中图法分类号** TP393 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2015.10.019

## Algorithm for Multisource Multicast with Network Coding over Multi-hop Wireless Networks

HAN Li<sup>1,2</sup> QIAN Huan-yan<sup>2</sup> LIU Hui-ting<sup>1</sup>

(School of Computer Science and Technology, Anhui University, Hefei 230039, China)<sup>1</sup>

(School of Computer Science & Technology, Nanjing University of Science & Technology, Nanjing 210094, China)<sup>2</sup>

**Abstract** We first presented a network coding based model for multisource multicast, in which the theory of back-pressure plays an important role in flow scheduling. Then we proposed a heuristic algorithm MulSrc which is compatible for 802.11 DCF MAC. It is especially well-suited for applications with low-loss, low-latency constraints. The use of network coding transparently implements both localized loss recovery and path diversity with low overhead. Simulation results show that our protocol outperforms the same kind protocol CodeCast and MMForests with multiflows.

**Keywords** Network coding, Multiflow, Back pressure routing, Adaptable flow rate, Multicast

多源多播问题中存在大量的流内干扰和流间干扰,一些针对单流研究设计的组播路由协议并不能很好地适应这种环境。因此,如何在多个多组播流并存的网络环境下准确度量节点的传输能力、合理分配有限的网络资源以提升网络有效承载能力是协议面向实用化的关键问题之一。

构建动态多播树森林在一定程度上解决了这个问题,即多播树森林的结构可以动态地根据网络条件调整<sup>[1]</sup>。背压路由算法<sup>[2]</sup>是实现动态多源多播的另一种选择。这是一个使用队列长度和链路两端队列长度的差值(背压)进行路由和调度的算法。这种算法允许以最大可能的速率向网络发送数据,又无需维持固定的路径,因此往往是最优的。然而如果不使用网络编码,即使在有线网络中,背压路由算法在多播传输中的应用也会变得非常复杂。结合网络编码的背压路由算法,可以综合使用本地补偿和动态路径来实现低延迟的分组丢失控制。在无线网络中,尤其是节点密集型网络,基于网络编码的多播协议可以利用丰富的多路径来克服相关丢失。

D. S. Lun<sup>[3]</sup>等实现了网络编码用于高丢失率网络中的可靠通信,并给出了 MANETs 网络中可靠多播协议的子图选择算法(即何时何地要将编码分组输出到网络)。但 MANETs 中的子图选择是一个 NP 难题,协议 CodeCast<sup>[4]</sup>给出了文献[3]的一个启发式实现。CodeCast 声明自己是一个基于网络

编码的、实现丢失控制和有界延迟的多播协议。但 CodeCast 并没有给出存在多源的解决方案。文献[5]给出了应用网络编码及动态多源多播实现流量调度的模型,但没有给出具体的解决方案。

本文提出了一个基于网络编码的动态多源多播模型,并根据该模型设计了一个适用于 802.11DCF MAC 的多播算法 MulSrc。MulSrc 结合了背压路由的思想,为各个流的每个目的节点维护一个虚拟队列,以队列积压作为流调度的主要依据及寻路和隐含的拥塞控制;以网络编码简化流量调度并用于实现路径的多样化;进行谨慎修复,即控制丢失的同时进行延迟检查,因而其适用于实时应用,如流媒体传输、阶段性传感器信息发布等。

在队列积压的计算上,提出应用信用度机制来实现流量分配。文献[6,7]均提出了用信用度的分配来控制流量分配的方法,但均没有给出具体的实现算法。MORE 协议<sup>[8]</sup>应用阶段性静态统计的方法计算信用度,但不适用于网络动态变化的情况。本文用动态统计方法统计各个流在每个节点上的信用度分配。

实验证明,在存在多个多播流的情况下,文中所提协议能够获得比协议 CodeCast 及应用动态多播树森林实现多源多播的 MMForests 协议<sup>[1]</sup>更好的性能。

到稿日期:2014-09-14 返修日期:2014-12-27 本文受国家自然科学基金(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;刘慧婷(1978—),女,副教授,主要研究方向为云计算。

## 1 模型

无线网络的物理层比较复杂,需要考虑多个传输节点信号有可能被多个节点接收,也可能从多个节点接收到来自同一信号源的信息。定义  $\underline{N}$  为网络上的节点集,  $n = |\underline{N}|$ 。定义  $\underline{L}$  为所有点到点链路  $(i, j)$  及超边  $(i, Z)$  的集合, 其中  $i, j \in \underline{N}$ ,  $Z \subset \underline{N}$ 。由于存在大量可能的无线链路, 仅考虑那些有效链路, 比如,  $Z$  可以选为信干比大于一定阈值的节点。

定义  $\underline{C}$  为网络上多播流的集合, 其中每个流  $c \in \underline{C}$  有一个源  $s_c \in \underline{N}$  和一组目的节点集  $T_c \subset \underline{N}$ 。

定义流调度配置矩阵  $\mathbf{A}$ ,  $A_{ik} = 1$  表示节点  $i$  转发流  $c$  的数据,  $A_{ik} = 0$  表示节点  $i$  不转发流  $c$  的数据。如果对所有  $i \in \underline{N}$ , 仅有一个  $c \in \underline{C}$  使得  $A_{ik} = 1$ , 则称流调度配置  $\mathbf{A}$  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  $\mathbf{S}$  是取值于有限域  $\underline{S}$  上的独立同分布变量; 定义包含所有速率向量  $\mathbf{R} = (R_i : i \in \underline{N})$  的网络图族为  $\underline{R}$ ,  $\underline{R}$  是物理层允许发送速率的集合, 定义  $\hat{\eta}$  是速率  $R_i$  的上界。定义  $p_{ij}(R_i, \mathbf{S})$  为节点  $j$  从节点  $i$  成功接收一个分组的概率。

假定信道状态  $\mathbf{S}$ 、路由  $\mathbf{A}$ 、速率  $\mathbf{R}$  取值于一个有限状态空间, 因此联合配置  $(\mathbf{S}, \mathbf{A}, \mathbf{R})$  具有一个时间平均概率, 定义该概率为  $\alpha_{\mathbf{S}, \mathbf{R}, \mathbf{A}}$ , 其中  $\alpha_{\mathbf{S}, \mathbf{R}, \mathbf{A}} \geq 0$ ,  $\sum_{\mathbf{S}, \mathbf{R}, \mathbf{A}} \alpha_{\mathbf{S}, \mathbf{R}, \mathbf{A}} \leq 1$ 。定义  $Q_i^{c\beta}$  为节点  $i$  上允许发往流  $c$  的目的节点  $\beta$  的分组数, 将其称为虚拟队列。记  $f_{ijZ}^{c\beta}$  为流  $c$  在链路  $(i, Z) \in \underline{L}$  上从  $Q_i^{c\beta}$  到  $Q_j^{c\beta}$  的平均虚拟流速率, 其中  $j \in Z$ 。

## 2 可行速率及流调度

为每个流  $c$  分配一个端到端的速率  $f_c$ , 记速率向量  $\mathbf{f} = (f_c)_{c \in \underline{C}}$ 。令  $\underline{F}$  是端到端速率向量  $\mathbf{f}$  的集合, 满足:

$$f_{\beta Z}^{c\beta} = 0, \forall j, c, \beta, Z \quad (1)$$

$$f_{ijZ}^{c\beta} \geq 0, \forall j, c, \beta, Z \quad (2)$$

$$f_c l_{i=\text{src}(c)} \leq \sum_{j, Z} f_{ijZ}^{c\beta} - \sum_{j, Z} f_{jZ}^{c\beta}, \forall i, c, \beta \neq i, \beta \in T_c \quad (3)$$

$$\sum_{j \in Z} \max_{\beta} f_{ijZ}^{c\beta} \leq R_{iZ}^c, \forall i, j, c, \beta, Z \quad (4)$$

$$R_{iZ}^c \leq \sum_{\mathbf{S}, \mathbf{R}, \mathbf{A}} \alpha_{\mathbf{S}, \mathbf{R}, \mathbf{A}} R_{iZ}, \forall i, Z \quad (5)$$

$\{f_{ijZ}^{c\beta}\}_{i, j, Z}$  给出了从流  $c$  的源节点到目的节点  $\beta$  的流量解决方案。每个在  $(i, Z)$  上传输的流  $c$  的物理分组都是一组分组的线性组合, 该组分组对应一个  $(i, Z)$  上的虚拟传输集合, 其中每个虚拟传输对应于  $T_c$  中一个不同的接收节点; 因此在  $(i, Z)$  上为流  $c$  分配的速率是  $\max_{\beta \in T_c} \sum_{j \in Z} f_{ijZ}^{c\beta}(t)$ 。从节点  $i$  到节点集  $Z$  的一次无线广播中, 尽管  $Z$  中可能有不止一个节点接收到了这个分组, 但它们会按一定的控制算法有选择地修改它们的虚拟队列, 使得每一个虚拟传输都是点到点的。即节点  $j \in Z$  可能会使用不同的策略来修改不同的目的节点  $\beta$  对应的虚拟队列  $Q_i^{c\beta}$ 。

**定理 1** 令  $\underline{F}$  是端到端速率向量  $(f_c)_{c \in \underline{C}}$  的集合, 并满足: 存在向量  $(f_{ijZ}^{c\beta})_{(i, Z) \in \underline{L}, c \in \underline{C}, \beta \in T_c}$ ,  $(R_{iZ}^c)_{(i, Z) \in \underline{L}, c \in \underline{C}}$ ,  $(\alpha_{\mathbf{S}, \mathbf{R}, \mathbf{A}})_{\mathbf{S} \in \underline{S}, \mathbf{R} \in \underline{R}, \mathbf{A} \in \underline{A}}$  使得式(1)~式(5)成立。应用流内网络编码的多个多播流稳定的必要条件是  $(f_c)_{c \in \underline{C}} \in \underline{F}$ 。

证明: 证明类似文献[5]。

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源节点发送分组的数目。

假定链路层协议为 802.11DCF, 其无线随机接入过程  $S(t)$  满足: 对于所有  $(i_1, J_1) \in S(t)$ , 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不同节点接收分组。链路  $(i, Z)$  上的流选择  $c^*$  可以简化为:

$$c^* = \arg \max_c \{ \max_{\beta \in T_c^*} \sum_{j \in Z} \max(Q_i^{c\beta} - Q_j^{c\beta}, 0) \} \quad (6)$$

以上动态策略依赖于队列长度信息进行决策, 不要求输入链路状态信息。直观的说, 该策略致力于在每个时隙中最大化虚拟传输的总量。

## 3 算法描述

本节将对本协议的工作细节进行描述。

### 3.1 术语

流: 一个源节点和它的接收节点构成一个流。

原始分组(Native Packet): 源节点发送的数据被划分为若干个等长的分组, 这些分组称为原始分组。

组(Generation): 源节点将数据划分为若干个长度为  $k$  字节的分组, 并将每  $l$  个分组组织为一个组。源节点和目的节点之间以组为单位进行通信。任何转发节点只对属于同一组的分组进行编码。

编码分组(Coded Packet): 对于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, 将  $c_i$  的取值范围设置为  $[0, 255]$ 。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。

新分组(Innovative Packet): 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”。

编码阈值: 为了降低接收节点收到的编码分组的线性相关性, 为编码设置一个阈值  $thred$ , 只有一个节点接收到的新分组个数大于  $thred$  时才发送编码分组。当  $thred = l - 1$  时, 中间节点需要进行完全的编解码;  $thred = 0$  表示中间节点获得一个组的第一个新分组后, 就可以发送数据。

信用度(Credit): 在中间节点  $i$  上为每个流的目的节点  $\beta$  定义一个变量, 即信用度  $credit_i^{c\beta}$ , 表示  $\beta$  每从源节点获取一个分组时, 该中间节点  $i$  所贡献的份额。信用度可以通过统计获得。对于任意  $\beta \in T_c$ , 源节点  $s_c$  的信用度  $credit_{s_c}^{c\beta}$  为 1, 节点  $\beta$  的信用度  $credit_{\beta}^{c\beta} = 0$ 。

距离: 基于最小 ETX 计算两个节点之间的距离。相邻节点  $(i, j)$  之间的距离等于  $1/p_{ij}$ , 即节点  $j$  每从节点  $i$  接收一个分组时, 节点  $i$  需要发送的分组数。

下游节点: 节点为每个流保留一个  $dist$  变量, 表示距离源节点的最小距离; 节点发送一个流的分组时会将该流的  $dist$  变量拷贝到分组的首部; 节点接收到一个新分组时, 会检查分组首部的  $dist$  变量, 如果该变量值小于本地的  $dist$  值, 则用该值更新本地值; 一个节点的下游节点即  $dist$  值大于自己的邻居节点。

流速率的分配: MulSrc 采用一种简单的方式为各个流  $c$  分配速率, 即仅当链路层允许源节点  $s_c$  发送时,  $s_c$  才会将流  $c$  的虚拟队列  $Q_c^\beta$  的长度加 1。

更新信用度  $credit_i^{\beta}$ : 定义  $credit_{ij}^c = count_{ij}^c / count_j^c$ , 其中  $count_{ij}^c$  表示节点  $j$  从节点  $i$  接收的流  $c$  的新分组数,  $count_j^c$  表示节点  $j$  应该接收的流  $c$  的新分组数。节点  $j$  需要定期将  $credit_{ij}^c$  反馈给节点  $i$ , 并定期向邻居节点通告自己的信用度  $credit_j^{\beta}$ 。 $credit_i^{\beta}$  可以通过如下公式进行更新:

$$credit_i^{\beta} = \sum_{j: (i,j) \in L} credit_{ij}^c \times credit_j^{\beta} \quad (7)$$

何时停止发送一个组: 节点停止发送来自同一个组的分组需满足如下条件: (a) 源节点缓存中的组超时。假定某个源节点的发送窗口大小为 32 个分组长度, 且选定的组中含有 32 个分组, 每个分组的数据长度为 1000 字节; 上层应用以 1Mbps 向协议推送数据, 则每个组在源节点上的保留时间为 256ms。(b) 节点的所有下游节点已经对该组完全解码。实验证明, 这个条件对减少不必要的发送非常重要。(c) 如果新到达分组的组号比当前缓存中的组号新, 则根据该组号对接收缓存进行更新。为简化计算, 假定源节点一次只发送一个组的数据。

### 3.2 工作流程描述

图 1 给出了发送分组的流程, 为提高效率, 假定节点仅在获得链路层允许时才发送数据。(a) 当链路层允许发送时, 节点  $i$  首先检查自己是否是某个流  $c$  的源节点。如果是, 则将虚队列长度  $Q_i^{\beta}$  加 1,  $\beta \in T_c$ ; 并检查当前组是否超时, 如果超时, 则将当前组移动到下一个等待发送的组。(b) 依据式 (6) 在所有不满足停止发送条件的组中选择需要转发的流  $c^*$ 。(c) 选定流  $c^*$  后, 源节点在属于同一组的  $l$  个原始分组上进行网络编码, 中间节点则对同一组中已接收的分组进行再编码。(d) 每转发一个流  $c^*$  的分组, 就将对应的虚队列长度  $Q_i^{c^* \beta}$  减  $credit_i^{c^* \beta}$ ,  $\beta \in T_{c^*}$ 。

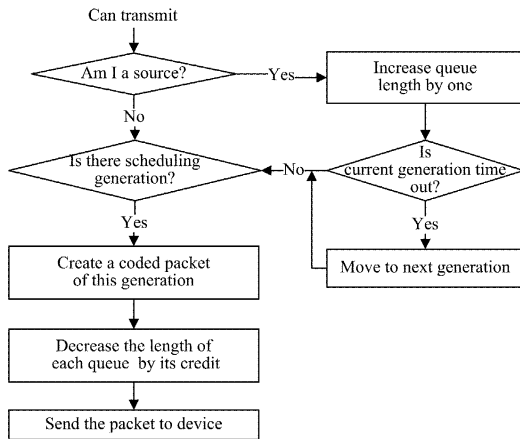


图 1 发送分组

图 2 给出了接收分组的流程。接收节点对每个接收到的分组应用高斯约旦消元法, 边接收边解码<sup>[9]</sup>, 同时检查其线性相关性, 当接收到同一组的  $l$  个分组时, 即获得了该组的  $l$  个原始分组。如果节点  $j$  从节点  $i$  接收的流  $c$  的分组是一个新分组, 则对于所有的  $\beta \in T_c$ , 将虚队列长度  $Q_j^{\beta}$  加上  $credit_{ij}^c$  的值, 并更新变量  $credit_{ij}^c$ 。如果节点  $j$  是流  $c$  的一个目的节点, 则将  $Q_j^{\beta}$  置为 0, 并准备发回 ACK。

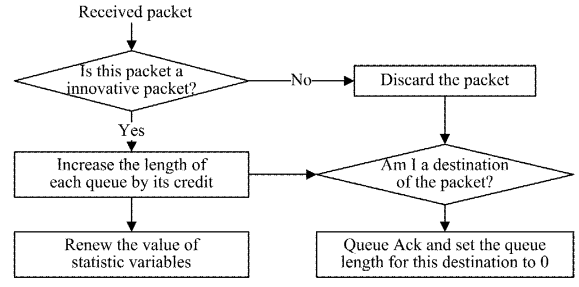


图 2 接收分组

### 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) 依次检查节点上的各个流, 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 (6) 从满足以上两个条件的组中选出在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  $|C|$  表示节点上缓存的流的数目,  $|n_d|$  表示下游节点的个数,  $|w|$  表示发送窗口中组的个数。

接收端的计算量主要集中在判断新分组的判断上, 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。

## 4 模拟结果及性能分析

本节考察 MulSrc 与 CodeCast 和 MMForests 协议之间的性能比较。其中 MMForests 实现完全可靠传输。由于 CodeCast 仅考虑了单个流的多播问题, 因此每个转发节点应用轮循的方式为节点上的流提供服务。

采用 NS2 来模拟协议性能, 实验环境描述如下。

(1) 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), 无线信号的衰减模型为地面反射模型 (TwoRay-Ground); 信道带宽最大为 4Mbps。(2) 每个组含有  $l$  个分组。实验发现, 将  $thred$  设置为  $l-1$  即完全编解码时协议效率最高, 因此实验中将  $thred$  值均设置为  $l-1$ 。(3) 分组长度为  $k$  字节; 实验发现,  $k$  较小时协议性能较高, 但过小的  $k$  值会使有效载荷降低, 因此在模拟实验中取  $k=1000$  Bytes。(4) 假定所有源节点的上层协议总有数据要发送, 且提供负载为  $a$  kbps。(5) 为简化控制, 滑动窗口大小  $|w|$  设置为 1, 即每个节点仅为每个流保留一个组的缓存空间。(6) 为模拟网络丢失, 节点以一定概率丢弃接收到的分组。(7) 运行时间为 20s, 实验结果取 20 次运行结果的平均值。

从以下几个角度对性能进行考评。

(1) 平均端到端吞吐量 (rate): 平均每个接收节点接收新分组的速率。(2) 重传开销 (cost): 网络上所有节点发送分组的总数与接收节点接收的新分组数的比率, 即目的节点平均每接收一个分组网络上需要发送的分组数, 反映了网络的整体开销。(3) 接收比率 (ratio): 接收端接收的有效数据量与上层协议实际推送的数据量的比率。(4) 为量化带宽资源在各个流之间分配的合理性, 引入了整体效用函数  $U(f) = \sum_c \ln(f_c)$ ,  $U(f)$  的值越大, 表明带宽资源分配越均匀<sup>[10]</sup>。为简化计算, 取  $f_c$  为流  $c$  的每个接收节点平均每秒接收新分组的个数。

图 3 给出了在平均邻居节点个数为 5、点到点丢失率为 0.1 的网络上任选的 8 个多播流、每个流的接收节点个数为 5 的情况下, MulSrc 与 CodeCast 在各个流上的接收比率、平均端到端吞吐量上的比较。其中圆柱体表示吞吐量 *rate*, 圆柱体上部的标签表示接收比率 *ratio*。从图 3 可以看到, CodeCast 中大部分流的吞吐量低于 MulSrc 协议, 但高于 MMForests 协议; CodeCast 和 MMForests 流量分布的最大特点是各个流的吞吐量分布不如 MulSrc 均匀, 即出现了极大流和极小流, 甚至是零流; 其中 CodeCast 的效用函数值为 23, MMForests 的效用函数值为 25, 而 MulSrc 效用函数值则为 30。由此可见, 本文协议可以取得比 CodeCast 协议和 MMForests 协议更合理的资源分配; MMForests 由于是一个针对多个流的多播协议, 其在流量分布上优于 CodeCast 协议。MulSrc 在接收比率上也表现出了优异的特性。

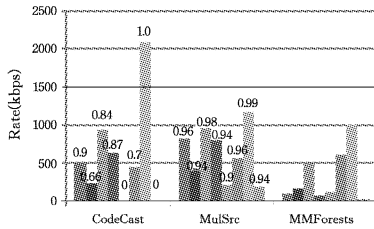
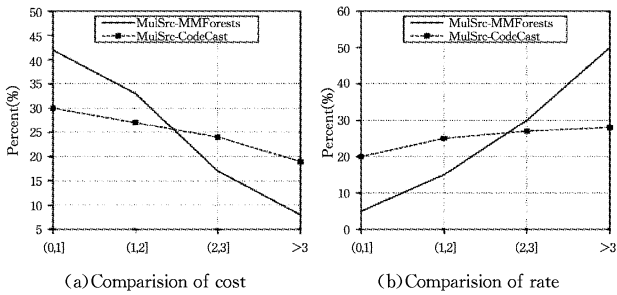
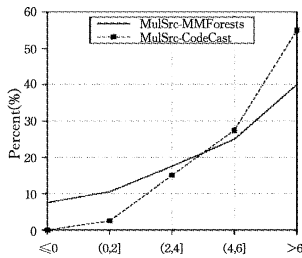


图 3 CodeCast 与 MulSrc 在多个流上的性能比较 ( $\alpha=300\text{kbps}$ ,  $k=1000\text{Bytes}$ ,  $l=4$ )



(a) Comparison of cost

(b) Comparison of rate



(c) Comparison of Utility

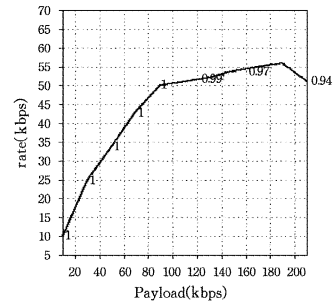
图(a)中的坐标点  $((a, b], \gamma)$  表示  $\text{cost}^{\text{CodeCast}} / \text{cost}^{\text{MulSrc}} \in (a, b]$  或  $\text{cost}^{\text{MMForests}} / \text{cost}^{\text{MulSrc}} \in (a, b]$  上的流占总实验流数的百分比为  $\gamma\%$ ; 图(b)中的坐标点  $((a, b], \gamma)$  表示  $\text{rate}^{\text{MulSrc}} / \text{rate}^{\text{CodeCast}} \in (a, b]$  或  $\text{rate}^{\text{MMForests}} / \text{rate}^{\text{CodeCast}} \in (a, b]$  的流占总实验流数的百分比为  $\gamma\%$ ; 图(c)中的坐标点  $((a, b], \gamma)$  表示  $U(f^{\text{MulSrc}}) - U(f^{\text{CodeCast}}) \in (a, b]$  或  $U(f^{\text{MulSrc}}) - U(f^{\text{MMForests}}) \in (a, b]$  的网络数占总网络数的百分比为  $\gamma\%$ 。

图 4 MulSrc 与 CodeCast 和 MMForests 在网络中存在 8 个多播流时的性能比较 ( $\alpha=300\text{kbps}$ ,  $k=1000\text{Bytes}$ ,  $l=4$ )

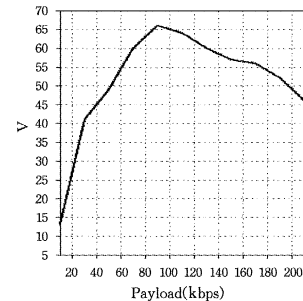
图 4 显示了任选 40 个平均邻居节点个数为 5、点到点丢失率为 0.1 的网络上, MulSrc 分别与 CodeCast 和 MMForests 在接收节点个数为 5 的 8 个多播流上的性能比较。如图 4(a) 所示, MulSrc 的重传开销小于 CodeCast 的流数占实验总流

数的 70%, 而 CodeCast 的重传开销是 MulSrc 两倍以上; 而 MulSrc 的重传开销低于 MMForests 的流仅占总流数的 55% 左右, 并无明显优势。如图 4(b) 所示, MulSrc 的吞吐量较 CodeCast 和 MMForests 均具有明显的优势, 其中吞吐量大于 CodeCast 的流占 80%, 而吞吐量大于 MMForests 的流占总流数的 90% 以上。可以看出, CodeCast 可以获得比 MMForests 更高的吞吐量。图 4(c) 给出了 3 个协议在整体效用上的比较, 其中 MMForests 具有比 CodeCast 更好的效用特性, 从中可以看出, MulSrc 的效用值比 CodeCast 的效用值大 6 的网络数占总实验网络数的 60%, 而比 MMForests 的效用值大 6 的网络数占总实验网络数的 45%。

图 5 给出了在平均邻居节点个数为 5、点到点丢失率为 0.2 的网络上任选 8 个多播流且每个流的接收节点个数为 5 的情况下, 随着提供负载的增加, MulSrc 性能的变化。引入变量  $V, V=10 \sum f_c / \text{cost}$ ,  $V$  值越大表明协议的“性价比”越高。由图 5(a) 可以看出, 随着各流提供负载的增加, *rate* 逐渐增大, 当提供负载约为 110kbps 后, 吞吐量逐渐达到饱和; 曲线上的标签表示该点的接收比率 *ratio*, 随着提供负载的增加, 目的节点的接收比率 *ratio* 虽然有所下降, 但均保持在 90% 以上的理想状态。观察图 5(b) 中变量  $V$  的曲线, 在提供负载约为 100kbps 时其出现了明显的下降, 即传输开销 *cost* 开始明显增大。由图 5 的实验可知, 给定一个网络, 提供负载对协议性能有着较大的影响, 即存在一个最佳的提供负载范围。在本实验中, 最佳提供负载在 100kbps 左右。这个特性可以被网络层用来动态检测网络, 向上层协议提供合适的推送速度, 从而适用于现在应用日益广泛的自适应流媒体传输。



(a) 平均端到端速率 *rate* 随提供负载的变化



(b) 变量  $V$  随提供负载的变化

图 5 随着提供负载的增加, MulSrc 的性能变化 ( $k=1000\text{Bytes}$ ,  $l=4$ )

**结束语** 本文提出了一个在无线多跳网络上的、基于网络编码的多源多播模型, 在此模型的基础上给出了一个适用于 802.11DCF MAC 的、以背压路由为流调度机制的解决方案 (下转第 112 页)

种干扰。FLA-IP 不需要修改协议标准, 仅需增加底层软件模块即可实现干扰预测功能。仿真结果表明, FLA-IP 不仅可以有效预测信道传输干扰因素, 还能在多种拓扑环境下进一步利用网络带宽, 提升帧传输效率。

## 参 考 文 献

- [1] Choi J, Na J, Park K, et al. Adaptive Optimization of Rate Adaptation Algorithms in Multi-Rate WLANs[C]//ICNP. 2007:144-153
- [2] Martorell G, Riera-Palou F, Femenias G. Tuning Fast Link Adaptation Algorithms for CSMA/CA- and CSMA/E2CA-Based WLANs[C]//2012 21st International Conference on Computer Communications and Networks (ICCCN). IEEE, 2012:1-7
- [3] Kamerman A, Monteban L. WaveLAN®-II; a high-performance wireless LAN for the unlicensed band[J]. Bell Labs Technical Journal, 1997, 2(3):118-133
- [4] Kim S, Verma L, Choi S, et al. Collision-aware rate adaptation in multi-rate WLANs: design and implementation[J]. Computer Networks, 2010, 54(17):3011-3030
- [5] Wong S H Y, Yang H, Lu S, et al. Robust rate adaptation for 802.11 wireless networks[C]//Proceedings of the 12th Annual International Conference on Mobile Computing and Networking. ACM, 2006:146-157
- [6] Kulkarni A, Menezes S, Prakash R. FREEZE: Rate adaptation in wireless LANs using channel contention estimates[J]. 2011 Third International Conference on Communication Systems and Networks (COMSNETS), 2011, 4(11):4-8
- [7] Lin T Y, Tsai C Y, Wu K R. EARC: Enhanced Adaptation of Link Rate and Contention Window for IEEE 802.11 Multi-Rate

- Wireless Networks[J]. IEEE Transactions on Communications, 2012, 60(9):2623-2634
- [8] Mardeni R, Solahuddin Y. Path loss model development for indoor signal loss prediction at 2.4GHz 802.11n network[C]//2012 International Conference on Microwave and Millimeter Wave Technology (ICMMT). IEEE, 2012:2:1-4
- [9] Abu-Sharkh O M F, Abdelhadi M J. The impact of multi-rate operation on A-MSDU, A-MPDU and block acknowledgment in greenfield IEEE802.11n wireless LANs[C]//Wireless Advanced (WiAd), 2011. IEEE, 2011:116-121
- [10] Politis A, Mavridis I, Manitsaris A, et al. X-EDCA: A cross-layer MAC-centric mechanism for efficient multimedia transmission in congested IEEE 802.11e infrastructure networks[C]//2011 7th International Wireless Communications and Mobile Computing Conference (IWCMC). IEEE, 2011:1724-1730
- [11] Schmidt F, Hithnawi A, Punal O, et al. A receiver-based 802.11 rate adaptation scheme with On-Demand Feedback[C]//2012 IEEE 23rd International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC). IEEE, 2012:399-405
- [12] Kulkarni P, Motz B, Lewis T, et al. Inferring Loss Causes to Improve Link Rate Adaptation in Wireless Networks[C]//2011 IEEE International Conference on Advanced Information Networking and Applications (AINA). IEEE, 2011:659-666
- [13] Han C, Dianati M, Tafazolli R, et al. Analytical study of the IEEE 802.11p MAC sublayer in vehicular networks[J]. IEEE Transactions on Intelligent Transportation Systems, 2012, 13(2):873-886
- [14] Issariyakul T, Hossain E. Introduction to network simulator NS2[M]. Springer, 2011

(上接第 91 页)

MulSrc. MulSrc 使用信用度在每个转发节点上为各个流的目的节点分配流量, 并将信用度作为队列积压演变的依据。将网络编码应用于背压路由, 可以减少节点协商带来的开销, 降低冗余数据的传输, 实现路径的多样化, 能充分利用无线网络中存在的丰富的多路径, 提高协议性能。在实验部分, 假定上层协议以不会引起拥塞的速率向协议推送数据, 将 MulSrc 与基于网络编码的多播协议 CodeCast 及多源多播协议 MMForests 在端到端吞吐量、整体开销、总体效用等方面进行了比较。实验证明, 在存在多个流的情况下, MulSrc 具有更好的性能。如何根据网络条件寻找最佳的提供负载、降低整体开销, 是 MulSrc 的下一步目标。

## 参 考 文 献

- [1] Chen Y R, Radhakrishnan S, Dhall S, et al. On multi-stream multi-source multicast routing[J]. Computer Networks, 2013, 57(15):2916-2930
- [2] Tassiulas L, Ephremides A. Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks[C]//Proceedings of the 29th IEEE Conference on Decision and Control, 1990. IEEE, 1990:2130-2132
- [3] Lun D S, M M M, Koetter R. Efficient Operation of Wireless Packet Networks Using Network Coding[C]//Proc. Int'l.

- Workshop. Convergent Technologies (IWCT) 2005. June 2005:1-5
- [4] Park J, Gerla M, Lun D, et al. Codecast: a network-coding-based ad hoc multicast protocol[J]. Wireless Communications, IEEE, 2008, 13(5):76-81
- [5] Ho T, Viswanathan H. Dynamic algorithms for multicast with intra-session network coding[J]. IEEE Transactions on Information Theory, 2009, 55(2):797-815
- [6] 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
- [7] Radunovic B. An Optimization Framework for Opportunistic Multipath Routing in Wireless Mesh Networks[C]//The 27th Conference on Computer Communications (INFOCOM 2008). 2008
- [8] Chachulski S, Jennings M, Katti S, et al. MORE: A network coding approach to opportunistic routing[R]. 2006
- [9] Li H, Huan Y Q. Parallelized Network Coding with SIMD instruction sets[C]//International Symposium on Computer Science and Computational Technology, 2008(ISCST'08). IEEE, 2008, 1:364-369
- [10] Kelly F P, Maulloo A, Tan D. Rate control in communication networks: shadow prices, proportional fairness and stability[J]. Journal of the Operational Research Society, 1998, 49(3):237-252