

# 一种基于网络编码的组播路由算法

李陶深 曾明霏 葛志辉

(广西大学计算机与电子信息学院 南宁 530004)

**摘 要** 网络编码是 2000 年提出的一种新算法,其主要优点是使组播传输速率能达到理论上限值。介绍了传统组播路由算法的局限性,分析了现有网络编码算法的优点和不足,在某个改进的网络编码数学模型上,提出了一种静态分布式分层网络编码 SDLNC 算法(Static Distributed Layered Network Coding)。模拟实验表明,该算法可以显著提高组播路由的数据传输速率。

**关键词** 网络编码,组播路由,分层编码

**中图分类号** TP391 **文献标识码** A

## New Multicast Routing Algorithm Based on Network Coding

LI Tao-shen ZENG Ming-fei GE Zhi-hui

(School of Computer and Electronics and Information, Guangxi University, Nanning 530004, China)

**Abstract** Network coding is a new algorithm first presented in 2000. Its main advantage is allowing the multicast transmission rate reach theoretically limits. This paper introduced the traditional multicast routing algorithms' limitation, analyzed the existing network coding algorithms' benefits and disadvantages, and based on an existence improved mathematical model of network coding, implemented a static distributed layered network coding (SDLNC) algorithm. Simulation results show that this algorithm can significantly improve multicast routing data transmission rates.

**Keywords** Network coding, Multicast routing, Layered coding

## 1 引言

提高组播传输协议的吞吐量,是实现组播协议在 Internet 上广泛应用的一个重要因素,也是组播协议的优势之一。现有的组播路由协议其组播成员的速率易受最低瓶颈带宽的影响,难以达到无向图最小割、最大流定理的理论上限值。常见的组播路由算法的理论基础主要是基于最小生成树的 Steiner 树算法<sup>[1]</sup>和基于中心点选择的 CBT(Core Based Tree)算法<sup>[2]</sup>。这两类算法都是基于最优路径思想的算法,使用某个标准寻找一条从源点到汇点的最优路径,舍弃非最优路径,只使用最优路径进行数据传输。此类算法由于必然舍弃网络中的一部分路径,网络资源的利用率不会很高,并使得网络中的负载集中在少部分最优路径之上,无法从本质上提高吞吐量。因此,必须使用最优路径之外的方法才能克服这些缺点。

2000 年新出现的网络编码的核心思想是让网络中的节点将接收到的数据进行某种数学计算,之后进行路由、转发等操作。它突破数据传输的固定模式,使得数据传输时可以使用网络中的所有路径,并充分地利用网络中的资源,让网络流量均匀地分布在整个网络之中,为进一步提高目前的组播网络传输的速率奠定了基础<sup>[3]</sup>。

目前,比较有代表性的网络编码算法有 LIF(Linear Information Flow)算法<sup>[4]</sup>和 HRRE(Heterogeneous Receiver

Rate Estimate)算法<sup>[5]</sup>。

LIF 算法是最早出现的网络编码算法之一,它通过对网络中的链路进行编码,然后根据链路的编码实现路由计算,最后汇点通过消去法还原原始数据。LIF 算法给出一个适用范围较广的网络编码算法,指出了网络编码算法的基本步骤,并证明了该算法能在多项式时间内完成。此外,它给出了一个简化了的消去法,用于判断线性相关性,使得算法可以快速还原原始数据。但是 LIF 算法是建立在有向无环网络基础上的,并且没有考虑异质性问题,当汇点间的差异较大时,网络的利用率受到很大的限制,不能充分利用带宽。

HRRE 算法是在 LIF 算法的基础上提出的,它通过将异质性节点进行分组,反复应用 LIF 算法的核心部分,利用分层编码来适应异质性节点的不同需求,解决了异构性问题,充分利用了带宽。但是 HRRE 算法和 LIF 算法一样,是一个为有向链路设计的算法,处理无向链路时只是将无向链路作为一对有向链路看待,并没有为无向链路进行优化。此外它还是一个在源点执行的集中式算法。

文献[6]中提出一个用于组播网络的网络编码的数学模型,证明模型中必然存在可行的组播路由,而且给出两个定义,分别是寻找路径和确定编码规则两方面的理论基础。

综合上述 3 篇文献的研究内容,本文提出一种静态分布式分层网络编码算法(SDLNC)。该算法的主要设计思想是:

到稿日期:2009-08-10 返修日期:2009-12-01 本文受广西自然科学基金项目(桂科自 0640026),国家自然科学基金(60963022)资助。

李陶深(1957—),男,博士,教授,CCF 高级会员,主要研究方向为计算机网络、路由算法等,E-mail: tshli@gxu.edu.cn;曾明霏(1984—),男,硕士生,主要研究方向为网络编码;葛志辉(1978—),男,博士,副教授,主要研究方向为无线网络等。

在网络模型的假设上保留静态网络、无延迟、单源和异构性这4个LIF算法和HRRE算法的共同假设,使用文献[6]提出的数学模型,目的是使得每个汇点的吞吐量接近或达到最小割最大流定理的理论上限值,尽可能提高组播网络数据传输速率。

## 2 算法结构

根据数学模型,一些拓扑结构比较特殊的中间节点可能无法起到编码的作用。例如节点度为2的中间节点,只能在相邻两节点间传递信息,不能起到编码作用。为了减少算法的复杂度,可以将这类没有编码作用的节点去除。

**定义1(不具备编码功能的节点)**  $V_i \notin R \cup s$  且  $V_i$  的度小于3,则称  $V_i$  为不具备编码功能的节点。

根据对数学模型的分析,计算节点编码是路由算法的第一步,之后是计算分层和具体路由,最后由汇点进行解码。因此,算法的总流程如下:

- Step1 计算每个节点的节点编码;
- Step2 源点根据节点编码计算分层;
- Step3 各个节点编码、组合、转发数据;
- Step4 汇点用消去法解码。

下面分别介绍这4个步骤的具体流程。

### 2.1 链路编码计算算法

计算节点编码前,首先要了解模型中关于不损失信息原则的定义。

**定义2(不损失信息原则)** 令  $vector(e)$  表示链路  $e$  上的向量组,  $band(e)$  表示链路  $e$  的容量。如果  $\forall e \in \Gamma^+(v)$ , 都有  $rank(vector(e)) = \min(rank(\{vector(e') \mid e' \in \Gamma^-(v)\}), bank(e))$

则称节点  $v$  的编码不损失信息。

根据数学模型,各个节点的节点编码可以通过源点  $s$  根据自身最大流发送一组线性无关的向量组,经过各个节点按定义2进行线性组合并转发之后,检测每个节点最终接收到的向量组即可。

为了保证节点的编码满足定义2,可以采用复制所接受到的向量到所有输出链路的方法。相对地,接受向量的时候就需要用消去法去除重复的向量,避免重复的向量在计算中多次出现。

此外,数据包从源点  $s$  到汇点  $R$  的过程可以视为是数据包不断远离  $s$  并趋近于  $R$  的过程。如果给出节点距离的定义,要求数据包不能向趋近  $s$  又远离  $R$  的节点发送,可以加快节点编码计算的速度。

**定义3(源点距离)** 如果  $V_i \in V - R$ , 数据包从  $s$  到达  $V_i$  的最小跳数称为  $V_i$  的源点距离;如果  $V_i \in R$ , 则  $V_i$  的源点距离为  $\infty$ 。

**定义4(汇点距离)** 如果  $V_i \in V - s$ , 数据包从  $\forall R_i \in R$  到达  $V_i$  的最小跳数称为  $V_i$  的汇点距离;  $s$  的汇点距离为  $\infty$ 。

节点距离采用扩散法计算,具体算法不再详述。计算完节点距离后,即可根据表1得到数据流动的方向。

表1 最大流计算数据传输方向表

| $V_1$ 是否向 $V_2$ 传输数据 | 源点距离 $V_1 > V_2$ | 源点距离 $V_1 = V_2$ | 源点距离 $V_1 < V_2$ |
|----------------------|------------------|------------------|------------------|
|                      |                  |                  |                  |

|             |   |   |   |
|-------------|---|---|---|
| $V_1 > V_2$ | Y | N | N |
| $V_1 = V_2$ | Y | Y | N |
| $V_1 < V_2$ | Y | Y | Y |

以下是节点编码计算伪代码描述:

```

Start
扩散法计算节点距离;
根据节点距离计算可转发链路;
while (等待新向量时间尚未超过阈值)
while (接收到新向量) {
if (新向量未通过消去法检查) then break;
for each 可转发链路 do {
if (可以直接转发) then 转发向量; else
if (待组合向量已存在) then 组合后转发;
else 保存向量等待组合;
}
}
发送缓存的未发送向量;
将计算结果用扩散法发回源点;
End

```

### 2.2 分层计算算法

分层计算主要是通过数学模型中编码相容的定义进行的。

**定义5(编码相容)**  $\alpha$  和  $\beta$  是两个链路编码,如果存在另一个链路编码  $\gamma$ , 满足  $\alpha \perp \gamma$  且  $\alpha + \gamma = \beta$ , 则称  $\alpha$  和  $\beta$  编码相容。特别地,当  $\gamma = 0$  时,即  $\alpha = \beta$  时,  $\alpha$  和  $\beta$  编码相容。

对于任意两条链路,只要链路编码相容,那么这两条链路属于同一条路径,即已知起点链路的编码和终点链路的编码就可以识别一条完整的路径。因此可以根据链路编码,建立流出源点  $s$  到某个汇点  $R_i$  之间的联系。将某个分层  $L_j$  分配给某条相邻链路,那么和  $R_i$  相邻链路中符合定义5的链路就可以收到  $L_j$ , 也就是  $R_i$  可以收到  $L_j$ 。同样地,可以用类似的办法把分层编码  $L$  中的其他分层分配给其他链路。

图1是分层计算的流程图。

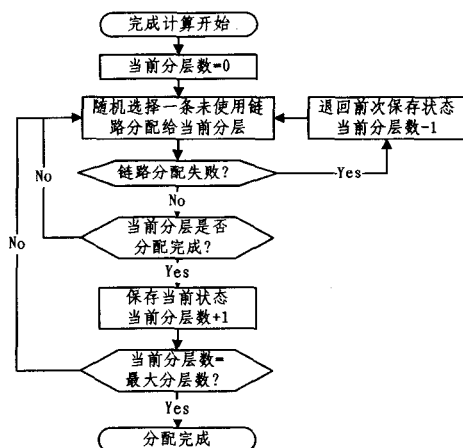


图1 分层计算流程图

### 2.3 路由计算算法

与分层计算算法相同,路由计算也是依靠数学模型中编码相容的定义进行的。

以下是路由计算算法伪代码描述:

```

Start
根据链路编码计算可转发链路;
while (等待新信息时间尚未超过阈值)
while (接收到新信息)

```

```

for each 可转发链路 do {
    if (编码不相容) then break;
    if (可直接转发) then 转发信息; else
        if (待组合信息已存在) then 组合后转发;
        else 保存信息等待组合;
}
将未发送信息发送;
End

```

其中,可直接转发条件和组合的条件均由链路编码判定。如果输入链路的编码和输出链路的编码相同,就可直接转发。编码相容的话,需等待另外一个相容的编码。

2.4 快速消去法算法

消去法用于节点编码计算和汇点解码数据流。该算法一次检验一个向量,并将计算结果保留在缓存  $\beta = (\beta_1, \beta_2, \dots, \beta_n)$  中,供下次计算使用。

- Step1 将  $\alpha$  复制到缓存  $\beta_{n+1}$  中;
- Step2 如果缓存中除了  $\alpha$  没有其他向量,缓存下标  $n = n + 1$ ,则返回检测成功;
- Step3 循环访问  $\beta_{n+1}$  中的每一个元素  $\beta_{n+1,i}$ ;
- Step4 如果  $\beta_{n+1,i} = 1$ ,则在  $\beta$  寻找第一个符合  $\beta_{j,i} = 1$  的  $j$ ,如果存在这样的  $j$ ,则  $\beta_{n+1} = \beta_{n+1} + \beta_j$ ;
- Step5 循环结束后,如果  $\beta_{n+1} = 0$ ,则返回检测失败,否则转到 Step6。
- Step6 按照出现第一个 1 的位置对缓存  $\beta$  中的元素进行插入排序,将  $\beta_{n+1}$  插入适合位置;
- Step7 缓存下标  $n = n + 1$ ,返回检测成功。

3 仿真实验与结果分析

在用 C# 语言编写的一个模拟实验平台下,通过两个模拟实验来验证所提算法的有效性。分别从以下两个方面来分析实验结果:

- 1)在随机生成的不同结构的网络拓扑中执行 SDLNC 算法,比较每个汇点的实际最大流,计算最大流和实际吞吐量的差异,并且和 IP 组播的最大可能值相比较,验证算法的普适性和有效性,突出算法的优势。
- 2)在同一个网络拓扑中,选择不同的汇集集  $R$ ,多次执行 SDLNC 算法。定义  $\eta = \sum_{R_i \in R} r_i / \sum_{R_i \in R} m_i$  为算法的效率,计算每一组节点的算法效率。比较各组节点的算法效率有无较大起伏,并且和 IP 组播的最大可能值相比较,说明 SDLNC 算法的普适性。

模拟实验 1 中的网络拓扑使用 gt-itm 工具生成,生成算法使用 Waxman 和 Transit-Stub 两种算法,分别用于生成网状网络和层次网络。由于 gt-itm 工具不生成链路带宽,因此带宽采用随机的方式生成。在 Waxman 模型中, $s$  选择度比较高的节点, $R$  随机选择;在 T-S 模型中, $s$  选择 Transit Domain 的节点, $R$  选择 Stub Domain 的节点。最后将符合定义 1 的无法编码的节点剔除。

模拟实验 1 采用的网络拓扑的参数如表 2 所列。

| 表 2 拓扑图生成参数表 |     |     |              |
|--------------|-----|-----|--------------|
|              | 节点数 | 边数  | 生成算法         |
| 拓扑图 1        | 30  | 132 | Waxman       |
| 拓扑图 2        | 20  | 76  | Waxman       |
| 拓扑图 3        | 60  | 228 | Transit-Stub |

算法执行的结果如图 2—图 4 所示。

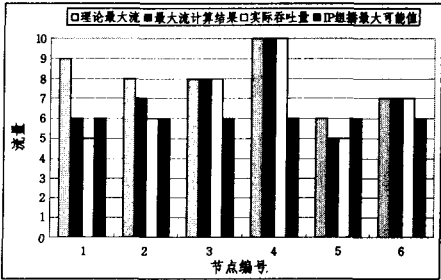


图 2 网络拓扑图 1 运行结果

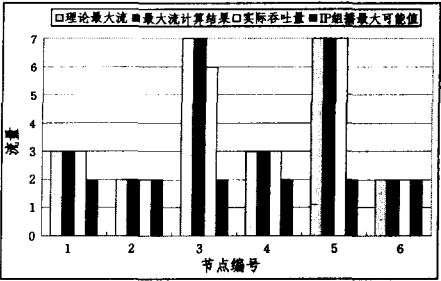


图 3 网络拓扑图 2 运行结果

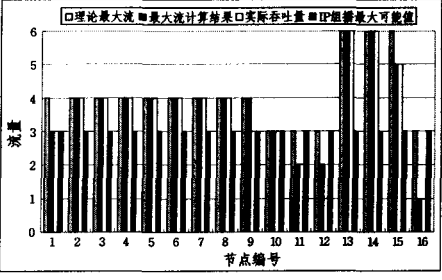


图 4 网络拓扑图 3 运行结果

从上面的 3 幅结果图中可以看出,SDLNC 算法在各种网络拓扑下计算出的最大流和实际吞吐量与理论上限值较接近,部分节点可以达到理论上限值,达到了算法设计的目标。与 IP 组播的最大可能值相比,SDLNC 算法的结果仍有一定的优势。特别在节点异构性较强的网络拓扑中,SDLNC 算法的优势更大。实验结果证明了 SDLNC 算法的正确性和有效性。

模拟实验 2 的网络拓扑采用 Waxman 算法生成,去除了度为 1 的节点。以节点 0 作为  $s$ ,随机选择了 8 组  $R$ ,每组  $R$  中含有 4 至 6 个节点。通过  $\sum_{R_i \in R} r_i / \sum_{R_i \in R} m_i$  计算算法的效率。图 5 为实验的网络拓扑,链路旁的数字为链路容量。

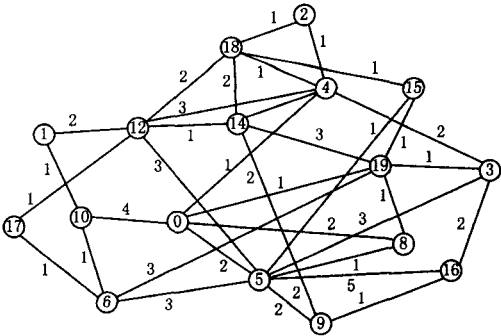


图 5 模拟实验 2 的网络拓扑图

on rough set theory: A review[J]. Applied Soft Computing, 2009,9(1):1568-4946

- [6] Leung Yee, Fung Tung, Mi Ju-Sheng, et al. A rough set approach to the discovery of classification rules in spatial data[J]. International Journal of Geographical Information Science, 2007,21(9):1033-1058
- [7] Yang Xibei, Qu Fang, Yang Jingyu, et al. A Novel Extension of Rough Set Model in Incomplete Information System[C] //Proceedings of the 2008 3rd International Conference on Innovative Computing Information and Control, June 2008;306-312
- [8] Sabu M K, Raju G. Rough Set Approaches for Mining Incomplete Information Systems[C]//Proceedings of the 4th International Conference on Intelligent Computing, Sep. 2008;914-921
- [9] 张文明,薛青. 粗糙集方法在作战仿真数据挖掘中的应用[J]. 系统仿真学报,2006,18(2):179-181
- [10] 于艾清,顾幸生. 基于广义粗糙集的不确定条件下的 Flow Shop 调度[J]. 系统仿真学报,2006,18(12):3369-3376
- [11] 王国胤. Rough 集理论在不完备信息系统中的扩充[J]. 计算机研究与发展,2002,39(10):1238-1243
- [12] 赵克勤. 集对分析及其初步应用[M]. 杭州:浙江科学出版社,

2000

- [13] 黄兵,周献中. 不完备信息系统中基于联系度的粗糙集模型拓展[J]. 系统工程理论与实践,2004(1):88-92
- [14] 徐怡,李龙涛,李学俊. 基于集对势的扩充粗糙集模型[J]. 系统仿真学报,2008,20(6):97-103
- [15] 黄兵,周献中. 基于集对分析的不完备信息系统粗糙集模型[J]. 计算机科学,2002,29(9 专刊):1-3
- [16] 王丽娟,吴陈,严熙. 基于限制容差关系和集对分析的数据依赖在 IIS 中的应用[J]. 系统工程理论与实践,2007,11:97-103
- [17] 张海东,舒兰. 限制容差关系下的集对变精度粗糙集模型[J]. 模糊系统与数学,2007,21(5):125-130
- [18] 杨习贝,杨静宇,於东军,等. 不完备信息系统中的可变精度分类粗糙集模型[J]. 系统工程理论与实践,2008,28(5):116-121
- [19] 赵克勤. SPA 的同异反系统理论在人工智能研究中的应用[J]. 智能系统学报,2007,2 (5):20-35
- [20] 赵明清,胡美燕,郭世伟. 量化容差关系与量化非对称相似关系的比较研究[J]. 计算机科学,2004,30(10):98-100
- [21] 张文修. 基于粗糙集的不确定决策[M]. 北京:清华大学出版社,2005

(上接第 124 页)

网络中各个节点的实际最大流如表 3 所列。

表 3 网络中各个节点的实际最大流

| 节点  | 1  | 2  | 3  | 4  | 5  | 6  | 8  | 9  |
|-----|----|----|----|----|----|----|----|----|
| 最大流 | 3  | 2  | 7  | 6  | 7  | 7  | 4  | 5  |
| 节点  | 10 | 12 | 14 | 15 | 16 | 17 | 18 | 19 |
| 最大流 | 5  | 7  | 7  | 3  | 7  | 2  | 6  | 6  |

模拟实验 2 的结果如图 6 所示。

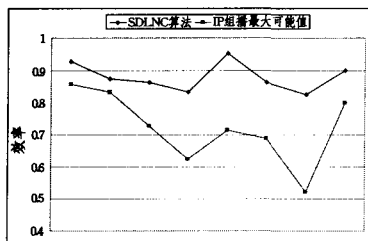


图 6 模拟实验 2 的运行结果

算法的效率表示 SDLNC 算法的组播容量和网络拓扑理论上最大的容量的比值。从图 4—图 6 的普适性实验结果可以看出,无论  $R$  如何选择,效率均在 0.8 以上,比同样条件下的 IP 组播要高。汇点的差异越大,SDLNC 算法的优势就越明显。这充分说明了 SDLNC 算法的高效率并不依赖于  $R$  的选择,而是网络编码算法的优越性带来的提高。

**结束语** 本文基于现有的网络编码算法,实现了一种组播网络的路由算法——SDLNC 算法。SDLNC 算法利用分层编码理论,充分满足了异质性节点对带宽的不同需求,并初步解决了同类算法中存在的一些问题。它的特点是网络中节点

不需要知道周围节点的情况,能在分布式计算的过程中逐步获取编码及路由所需要的信息。仿真试验的结果表明,本算法在不同的网络拓扑下都有不错的性能,均接近或达到了理论上限值。

但是 SDLNC 算法也存在着一些不足之处,并没有完全实现数学模型中的目标,还有很大的优化和改进的空间。

## 参考文献

- [1] Sarrafzadeh M, Wong C K. Bottleneck Steiner Trees in the Plane [J]. IEEE Transactions on Computers, 1992,41(3):370-374
- [2] RFC2189. Core Based Trees (CBT version 2) Multicast Routing [S]
- [3] Ahlswede R, Cai Ning, Li S-Y R, et al, Network Information Flow [J]. IEEE Transactions on Information Theory, 2000,46 (4):1204-1216
- [4] Sanders P, Egner S, Tolhuizen L. Polynomial Time Algorithms for Network Information Flow [C]//Proceedings of the 15th annual ACM Symposium on Parallel Algorithms and Architectures, San Diego California USA, 2003;286-294
- [5] Sundaram N, Ramanathan P, Banerjee S. Multirate Media Streaming Using Network Coding [C]//Proc. 43rd Allerton Conference on Communication Control and Computing, Monticello IL, Sep. 2005
- [6] Koetter R, Medard M. An Algebraic Approach to Network Coding [J]. IEEE/ACM Transactions on Networking, 2003,11 (5):782-795