一种高效的冗余编码 Mesh 流媒体覆盖网

王锐[1]，钱德沛[1,2]，朱青林[1]，刘涛[2]，栾钟治[1]
(1. 北京航空航天大学计算机学院，100083，北京；2. 西安交通大学电子与信息工程学院，710049，西安)

摘要：为了减小以 Mesh 方式传输流媒体的覆盖网络节点动态加入和退出对数据流传输稳定性的影响，建立了一种基于冗余编码的流媒体多播机制. 数据源以冗余方式将数据流编码为多个等长的分支，以数据流分支作为路由和传输单位，而网络中的每个节点通过与其他节点建立邻居关系来获取所需的数据分支. 在新节点加入的过程中，提出了一种高效的邻居节点选择算法，它充分利用了冗余编码特点，以此降低自身的时间复杂度. 在节点失效处理的过程中，推出了一种数据分支替换机制，通过已有父节点的轮转替换可快速恢复数据传输，从而避免了传输中断问题. 模拟试验表明：所提机制可以在网络抖动度小于 10 的范围内保证 90%以上节点的传输质量，使得高冗余度的编码更好地适应高度变动的网络；在网络抖动度大于 8 时，冗余度在 0%～33.3%范围内的编码不会显著降低网络的传输效率.

关键词：覆盖网；流媒体；冗余编码
中图分类号：TP393 **文献标志码**：A **文章编号**：0253-987X(2009)10-0056-05

An Effective Redundant Coded Mesh Overlay Streaming Network

WANG Rui[1], QIAN Depei[1,2], ZHU Qinglin[1], LIU Tao[2], Luan Zhongzhi[1]
(1. School of Computer Science, Beihang University, Beijing 100083, China; 2. School of Electronics and Information Engineering, Xi′an Jiaotong University, Xi′an 710049, China)

Abstract: The network dynamics is one of the most important factors that impact the stability of the mesh streaming networks. A streaming multicast mechanism using redundant coding is proposed to alleviate the influence to the transmission performance from dynamic nodes. The media source encodes the stream into several sub-streams with equal length redundantly, and distributes them to different successors. Each node in the network establishes the cooperative relationships with others to get the required sub-streams. A parent selection algorithm with relatively lower complexity is proposed to take full advantage of redundant coding. In case of node's failure, a repairing algorithm of swap-in-turn between different sources of sub-streams is proposed. The algorithm makes the best use of the current parents to ensure high continuity of steaming transmission. Simulation results show that the multicast mechanism can ensure high quality for more than 90% users when the churn rate is below 10; and the codes with higher redundant degree can adapt to more dynamic scenario; and that the codes with redundancy less than 33.3% does not significantly decrease the effective transmission rate when the churn rate is beyond 8.

Keywords: overlay network; streaming; redundant coding

互联网范围的流媒体服务得到了越来越多的关注. 随着用户规模的增加，传统的客户端/服务器模

收稿日期：2009-02-05. 作者简介：王锐(1978—)，男，博士生；钱德沛(联系人)，男，教授，博士生导师. 基金项目：国家自然科学基金资助项目(60673180，90612004)；国家高技术研究发展计划资助项目(2006AA01A106，2006AA01A118)；科技部国际合作项目(2006DFA11080).

式已无法承受动辄数以百万计的用户负载，而 IP 层多播协议很长时间以来均未受到广泛的支持. 覆盖网是解决网络基础设施发展缓慢与网络应用日新月异矛盾的一种有效途径[1]，构建覆盖网络，以对等方式协作传输数据，已成为大规模流媒体系统的有效解决方案. 这其中，以随机 Mesh 方式组织多播网络在带宽利用率、鲁棒性等方面占据了主导地位. PPLive[2]、CoolStreaming[3] 等均采用了这种方式，并取得了很好的效果.

随机 Mesh 流媒体网络的基本特征是，由数据源将流媒体数据编码后分成多个支流，再分别传给不同的后续节点，并由这些节点转发给更多的节点，而节点之间互相补充数据支流. 当某些节点失效时，其后续节点必须在一定的时限内找到替代的数据来源才能不影响接收质量. 如何保证节点数据来源的稳定性、降低节点失效对后续节点的影响是一个关键问题.

PRIME[4] 采用了一种将带宽与节点度相关联的方式，根据带宽与节点度的比值发现瓶颈并做出调整. Reed-Solomon(RS)编码[5] 把原有数据加入一定的冗余度重新编码，使后续节点只需要接收部分数据就可以使用. Kumar[6] 采用了节点缓存来避免传输中断.

本文提出了低复杂度的邻居选择算法，采用 RS 编码，以冗余方式传输流数据，并优化了网络节点在父节点失效时的处理策略，同时利用其他父节点和备份节点来缩短寻找替代数据来源的时间，提高网络对节点动态性的应对能力.

1　Mesh 流系统框架

系统中的每个视频流可划分成多个等长的支流，经编码后分别传输给不同的子节点. 一个新节点会从系统获得一个节点列表，并从列表中选择一部分节点来建立邻居关系，之后再向它的多个邻居节点分别请求不同的支流. 一旦获得了一个数据支流，新节点将在它的上传带宽范围之内为其他节点提供上传服务，且优先提供时延较小的分支. 假设一个邻居只提供一个支流，并且节点只从单个邻居下载任意一个支流，支流与邻居是一对一的关系.

用 s' 表示原数据拆分后的支流数，s 表示经 RS 编码后产生的支流数. 节点只要收到任意个 s'，就可以完成数据恢复.

假设每个节点有 B 个覆盖网邻居，节点至少需要选择 S 个覆盖网邻居提供流数据，并选择 D 个可以提供数据的节点作为数据来源的备份.

整个网络可以抽象为一个图，图的节点代表网络节点，边代表网络节点的邻居关系. 对于一个处于某一时刻的系统，节点的关系定义如下.

定义 1　如果节点 i 向节点 j 提供支流 k，那么称节点 i 是节点 j 关于支流 k 的活动父节点.

定义 2　如果 2 个节点之间仅存在连接，并不相互提供数据，那么这 2 个节点互为邻居.

用一个 $n\times n$ 的邻接矩阵 $\boldsymbol{C}$ 表示图中节点的关系，$\boldsymbol{C}$ 中的每一个元素 c_{ij} 表示节点 i 和节点 j 的关系，则 c_{ij} 的值定义为

$$c_{ij}=\begin{cases}0 & i=j\\ 0 & \text{节点 } i,j \text{ 之间没有连接}\\ \eta & \text{节点 } i,j \text{ 是邻居}\\ \alpha_k & \text{节点 } i \text{ 是节点 } j \text{ 关于支流 } k \text{ 的活动父节点}\\ \beta_k & \text{节点 } i \text{ 是节点 } j \text{ 关于支流 } k \text{ 的备份父节点}\end{cases}$$

其中的父节点关系是有向的. 由于每个节点的上行带宽有限，对于一个给定的支流速率，一个节点可以具有的活动子节点的最大数目是有限的.

$\boldsymbol{C}$ 是一个表示全局信息的邻接矩阵，但其每个节点 i 维护的信息是该矩阵的第 i 行 $\boldsymbol{L}_i=[c_{i0},c_{i1},\cdots,c_{in}]$(节点 i 的所有子节点)及第 i 列 $\boldsymbol{V}_i=(c_{0i},c_{1i}\cdots,c_{mi})^{\mathrm{T}}$(节点 i 的所有父节点)，并且去掉向量中取值为 0 的元素(对不相邻的节点不保存任何信息). 这种表示方式使得节点只需要保存局部信息，是一种支持分布式算法的模型.

2　节点变动处理算法

从图的角度看，网络的状态变迁是由节点的加入和退出来驱动的. 下面利用邻接矩阵来描述节点的加入和退出.

2.1　节点的加入

在任意时刻，网络可以表示为一个确定的邻接矩阵. 当一个新节点到达时，它会从网络中随机挑选一个节点，然后在这个节点的帮助下，建立自己的邻居列表. 相应地，在邻接矩阵 $\boldsymbol{C}$ 中会增加一个新行和一个新列，变成 $(n+1)\times(n+1)$ 矩阵. 当新节点找到一定数量的节点作为邻居后，还需要将相应的邻居关系值修改为 η. 节点挑选 B 个邻居作为活动父节点，并发出对支流的请求，另外选择 D 个节点作为备份父节点，备份父节点不需要为子节点预留带宽.

对任意节点 b，它从每个邻居节点 i 接收一个向

量 $\boldsymbol{V}_{i,b}$，向量中包含了邻居可以提供支流的标识. 每个向量具有 R 个元素，每个元素分别代表一个支流. 如果一个邻居可以提供支流 k，那么第 k 个元素为 a_k，否则为 0. 在接收到的向量 $\{\boldsymbol{V}_{0,b},\boldsymbol{V}_{1,b},\cdots,\boldsymbol{V}_{x,b}\}$ 中，节点 b 为每个支流选择数据来源，即在向量组成的矩阵中为每个支流挑选一个对应向量的支流元素为 α_k 的节点，组成一个完整的流，并且为每个支流均匀地预留备份节点. 用邻接矩阵可以描述为，完善节点 b 对应的列向量 $\boldsymbol{V}_b=(c_{0b},c_{1b},\cdots,c_{nb})^{\mathrm{T}}$，使该向量中至少包含全部 $\{\alpha_1,\alpha_2,\cdots,\alpha_n\}$ 的每个元素，并尽可能均匀地包含 $\{\beta_1,\beta_2,\cdots,\beta_n\}$. 父节点选择算法的主要思想是：①保证每个支流都有提供者；②尽可能选择支流资源较多的节点作为活动父节点，因为该节点本身就是一种潜在的备份节点；③在备份节点中，每个支流的后备数量应尽可能均匀. 该算法可以先转换为二分图匹配问题，然后用复杂度为 $O(n^3)$ 的匈牙利算法来解决. 由于编码有一定的冗余度，所以无需采用高复杂度的匈牙利算法. 下面给出算法描述.

(1)设邻居集合为 N，支流集合为 S，每个邻居提供的支流集合所组成的集合为 $\Gamma=\{S_0,\cdots,S_n\}$.

(2)将邻居按照提供支流的个数从大到小排序，得到向量 $\boldsymbol{N}_S$.

(3)按顺序统计结果提供每个支流的邻居，当找到了全部支流的源，并且遍历的邻居数大于或等于支流数，停止遍历，记录当前位置，得到 $\boldsymbol{N}_S$ 的子向量 $\boldsymbol{N}'$.

(4)每个支流按照源的个数递增排序，得到向量 $\boldsymbol{S}$.

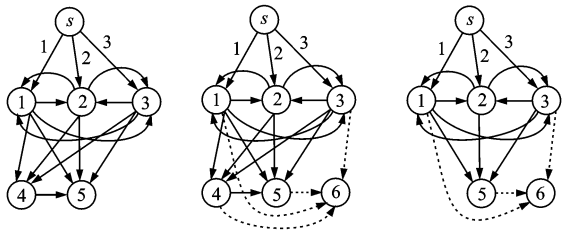
(5)在可选择的源中，每个支流应选择可以提供支流最少的节点作为活动父节点. 一旦活动父节点被选中，就将其从邻居集合中删除. 全部选择完成后，剩余邻居将作为备份节点.

(6)在备份节点集合中，计算 $\boldsymbol{S}$ 的每个分支的数据来源个数 b_m，并求出平均数 avg(b). 为 $\boldsymbol{S}$ 中满足 $b_m<\mathrm{avg}(b)$ 的数据分支 m 查询更多的备份节点，直到 $b_m\geqslant\mathrm{avg}(b)$，算法结束.

下面是一个节点加入、退出的过程示例(见图1)，网络连接如图 1a 所示. 假设网络中的支流数 $s=3$，$s'=2$，α_k 的取值为 $\{\alpha_1,\alpha_2,\alpha_3\}$，除源点外，网络有 5 个节点.

假设此时网络的邻接矩阵为

$$\boldsymbol{C}=\begin{bmatrix}0&\alpha_1&\alpha_1&\alpha_1&\alpha_1\\\alpha_2&0&\alpha_2&\alpha_2&\beta_2\\\alpha_3&\alpha_3&0&\alpha_3&\alpha_3\\\eta&\eta&\eta&0&\alpha_2\\\eta&\eta&\eta&\eta&0\end{bmatrix}$$



(a)网络连接 (b)节点 6 加入 (c)节点 4 退出

图 1 节点加入与退出过程

需要指出的是，给定网络连接，其邻接矩阵并不唯一. 当节点加入后，网络连接关系如图 1b 所示，则有

$$\boldsymbol{C}=\begin{bmatrix}0&\alpha_1&\alpha_1&\alpha_1&\alpha_1&\eta\\\alpha_2&0&\alpha_2&\alpha_2&\beta_2&0\\\alpha_3&\alpha_2&0&\alpha_3&\alpha_3&\eta\\\eta&\eta&\eta&0&\alpha_2&\eta\\\eta&\eta&\eta&\eta&0&\eta\\\eta&0&\eta&\eta&\eta&0\end{bmatrix}$$

其中第 6 行表示节点 6 的所有子节点，第 6 列表示节点 6 的所有父节点. 节点 6 可以向它的邻居提供资源，假设节点 6 接收到如下的向量

$$\boldsymbol{V}_{1,6}=[\alpha_1,\alpha_2,0]$$
$$\boldsymbol{V}_{3,6}=[0,\alpha_2,\alpha_3]$$
$$\boldsymbol{V}_{4,6}=[0,0,\alpha_3]$$
$$\boldsymbol{V}_{5,6}=[\alpha_1,0,0]$$

根据选择算法得到

$$\boldsymbol{C}=\begin{bmatrix}0&\alpha_1&\alpha_1&\alpha_1&\alpha_1&\alpha_1\\\alpha_2&0&\alpha_2&\alpha_2&\beta_2&0\\\alpha_3&\alpha_2&0&\alpha_3&\alpha_3&\alpha_2\\\eta&\eta&\eta&0&\alpha_2&\alpha_3\\\eta&\eta&\eta&\eta&0&\beta_1\\\eta&0&\eta&\eta&\eta&0\end{bmatrix}$$

可以看出，邻接矩阵 $\boldsymbol{C}$ 的最后一行无 α_k、β_k，表明新节点对系统中的原有节点没有做出带宽贡献，所以需要将新节点可以提供的数据分支信息发给原有节点，使新加入的节点为原有节点提供备份服务. 这个过程与新节点加入时获取资源向量的过程类似. 最后，更新的邻接矩阵为

$$\boldsymbol{C}=\begin{bmatrix}0&\alpha_1&\alpha_1&\alpha_1&\alpha_1&\alpha_1\\\alpha_2&0&\alpha_2&\alpha_2&\beta_2&0\\\alpha_3&\alpha_2&0&\alpha_3&\alpha_3&\alpha_2\\\beta_2&\eta&\beta_2&0&\alpha_2&\alpha_3\\\beta_3&\beta_1&\eta&\eta&0&\beta_1\\\eta&0&\eta&\beta_2&\beta_3&0\end{bmatrix}$$

2.2　节点退出/失效

当节点 b 退出网络时，所有与之相关的连接被取消，即邻接矩阵的第 b 行和 b 列均为 0，其所有后继节点需要立刻找到替代者. 如果这个节点是某个节点的备份父节点，那么该节点不需要立刻做出反应，只需要在下一次的更新过程中增加一个备份父节点. 如果退出的节点是某节点的活动父节点，那么该节点需要将它的某个备份父节点的状态切换为活动父节点.

目前，大多数节点失效处理算法只为缺失的支流寻找替代节点，成功率很低. 本文采用一种轮转式替换方式，即在所有邻居提供的可用支流向量中重新选择一个匹配的替代节点. 节点失效后的处理算法如图 2 所示.

在图 1 的示例中，假设节点 4 退出，网络连接如图 1c 所示，此时节点 6 依据资源向量集合{$\boldsymbol{V}_{1,6}$，$\boldsymbol{V}_{3,6}$，$\boldsymbol{V}_{4,6}$，$\boldsymbol{V}_{5,6}$}、节点失效处理算法调整数据来源，邻接矩阵的第 6 列变化如图 3 所示.

```
input: active parents collection V_i, the failed stripe id : x;
substitute parents collection Sub.
output: final stripe collection V_i.
foreach p in Sub do
  subnode= find_stripe(a_k, x, p);
  if subnode not null
      add subnode to V_i; return;
  endif
end foreach
foreach n_i in V_i do
    add n_i to V;
end foreach
V_i =Algorithm (V);//调用节点加入的邻居选择算法
```

图 2　节点失效处理算法

$$[\alpha_1\ 0\ \ \alpha_2\ \alpha_3\ \beta_1\ 0\ ]^{\mathrm{T}}$$

$$[\alpha_2\ 0\ \ \alpha_3\ 0\ \ \alpha_1\ 0\ ]^{\mathrm{T}}$$

图 3　节点失效处理后邻接矩阵的第 6 列变化结果

在第 6 列中，节点 6 因为节点 4 的退出失去了分支 3 的来源，此时无备份节点提供分支 3，但是提供分支 2 的节点 3 可以提供分支 3，并由节点 1 替代节点 3 提供分支 2，由备用节点 5 提供分支 1，这样就避免了缺失分支 3 之后暂时无法找到数据来源而造成质量损失. 这样，整个邻接矩阵变为

$$\boldsymbol{C}=\begin{bmatrix} 0 & \alpha_1 & \alpha_1 & 0 & \alpha_1 & \alpha_2 \\ \alpha_2 & 0 & \alpha_2 & 0 & \alpha_2 & 0 \\ \alpha_3 & \alpha_2 & 0 & 0 & \alpha_3 & \alpha_3 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ \beta_3 & \beta_1 & \eta & 0 & 0 & \alpha_1 \\ \eta & 0 & \eta & 0 & \beta_3 & 0 \end{bmatrix}$$

其中除了节点 6 调整数据来源以外，节点 5 也因为节点 4 的离开需使用备份节点 2 提供分支 2.

3　模拟试验

利用 Peersim[7] 模拟平台进行了若干组试验，试验的硬件环境为一台 4 核双 CPU、主频 2.0 GHz、内存 4 GB 的 IBM HS21 刀片服务器. 在试验中创建了节点连接度符合幂律分布的物理网络拓扑，在其中建立了 1 000 个覆盖网节点，采用莱斯大学的 DS^2 生成节点间的时延矩阵[8]. 参考 Coolstreaming[3] 设置：流速率为 400 kb/s，每个数据段的大小为 400 kb，播放时间为 1 s. 每个数据段划分为 400 个长度为 1 kb 的数据片，然后将每个数据段中的数据片以冗余方式编码，如 RS[9，8]编码先将原数据拆分成 8 个支流，然后编码为 9 个支流，每个支流的流速为 50 kb/s.

节点随机加入和离开系统，间隔时间均服从数学期望为 λ 的泊松过程，即节点的加入平均速率和离开平均速率均为 $1/\lambda$. 通过变化 λ 值可以得到不同的平均速率.

对试验中用到的参数和评价指标做如下定义.

网络抖动度：在一个给定的时间间隔 t（设为 100 s）内，离开系统的节点总数与系统中节点数平均值的比值. 这个值越大，说明节点变化越频繁. 若一个抖动度为 2 的系统，λ 的值为 0.05，节点加入和退出的平均速率为 20 s^{-1}，则在 100 s 内，退出的节点数约为 2 000 个，它等于系统平均节点数的 2 倍.

数据中断：在节点离开后考察它的每个子节点. 如果该子节点在现有的活动父节点和备份父节点中无法找到完整的匹配流，则定义为一次数据中断.

传输质量：在父节点失效后，没有发生数据中断的子节点占全部子节点的百分比.

传输效率：所有节点收到的数据解码为原始数据后的总数据量与网络中数据总流量的比值.

考察节点失效算法与冗余编码对传输质量的影响，其中的冗余编码采用 RS[12，8]编码的结果如图 4 所示.

由图 4 可以看出，采用冗余编码和节点失效算法可以提高传输质量，将二者结合后，在抖动度小于 10 的范围内能保证 90%以上节点的传输质量.

传输质量对冗余度的敏感度如图 5 所示. 可以看出，随着冗余度的增加，传输质量逐渐提高.

将本文算法与非冗余编码方式的传输效率进行比较，结果如图 6 所示. 可以看出：在节点轻微变动

的情况下，编码冗余度较大，传输效率较低；在网络抖动度较大(大于8)的情况下，即使RS[12,8]的编码冗余度为33.3%，本文算法的网络传输效率也没有显著下降.

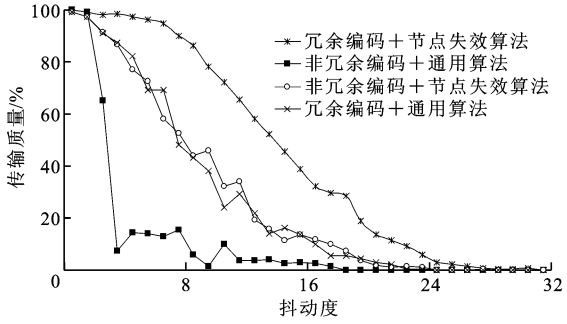


图4 冗余编码与节点失效算法对传输质量的影响

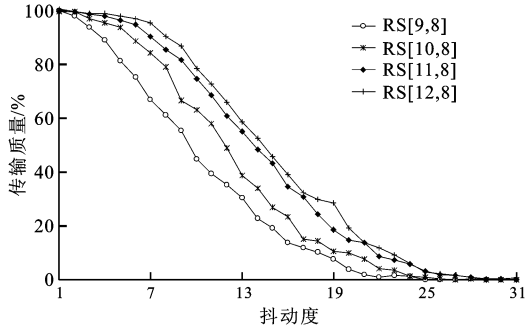


图5 不同冗余度编码的传输质量对比

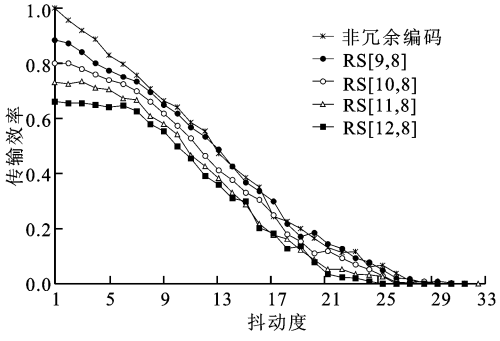


图6 2种编码方式的传输效率对比

4 结 论

本文提出了一种能够快速响应节点变化的Mesh流媒体传输网络，建立了系统模型，并提出了节点加入和失效的处理算法. 分析与试验表明：在节点频繁变动的场景下，采用冗余编码和节点失效处理机制均可有效减少数据中断；高冗余度的编码更适应高度变动的网络；冗余编码不会显著降低网络的传输效率，网络传输效率对编码的冗余度不敏感.

参考文献：

[1] 刘涛，钱德沛，王锐，等. 一种可演化网络的研究与实践[J]. 西安交通大学学报，2008，42(10)：1193-1203.
LIU Tao，QIAN Depei，WANG Rui，et al. Research and practice on evolutionary networks[J]. Journal of Xi′an Jiaotong University，2008，42(10)：1193-1203.

[2] HUANG Y，FU T Z J，CHIU D M，et al. Challenges，design and analysis of a large-scale P2P-VoD system[C]// ACM SIGCOMM 2008 Conference on Data Communication. New York，USA：ACM，2008：375-388.

[3] LI Bo，XIE Susu，QU Yang，et al. Inside the new coolstreaming：principles，measurements and performance implications[C]// The 27th IEEE Conference on Computer Communications. Piscataway，NJ，USA：IEEE，2008：1031-1039.

[4] MAGHAREI N，REJAIE R. PRIME：peer-to-peer receiver-driven mesh-based streaming[C]// The 26th IEEE Conference on Computer Communications. Piscataway，NJ，USA：IEEE，2007：1415-1423.

[5] KOETTER R，VARDY A. Algebraic soft-decision decoding of Reed-Solomon codes[J]. IEEE Trans on Information Theory，2003，49(11)：2809-2825.

[6] KUMAR R，LIU Y，ROSS K. Stochastic fluid theory for P2P streaming systems[C]// The 26th IEEE Conference on Computer Communications. Piscataway，NJ，USA：IEEE，2007：919-927.

[7] JELASITY M. PeerSim[EB/OL]. [2008-11-21]. http://peersim.sourceforge.net/.

[8] EUGENENG T S. DS2[EB/OL]. [2008-11-21]. http://www.cs.rice.edu/～bozhang/ds2/.

(编辑 苗凌)