

文章编号:1001-9081(2010)02-0285-04

## 基于网络编码的 P2P 流媒体推拉结合数据调度方法 customR2

王 晖,陈伟涛,刘亚杰

(国防科学技术大学 信息系统与管理学院,长沙 410073)

(huiwang@nudt.edu.cn)

**摘 要:**相比传统的基于 Pull 模式的 P2P 流媒体数据调度方法,基于随机网络编码的随机 Push 方法 R2 具有提高视频服务质量、降低服务器负载和增强对节点动态的适应能力等优势,但是容易产生大量的冗余编码数据块,消耗一定的网络带宽的问题。因此提出了一个基于网络编码的 P2P 流媒体推拉结合数据调度方法 customR2。当下游节点某个数据段所需的编码数据块将要接收完成时,通知它的种子节点停止向它 Push 该数据段编码数据块,转为向它的种子节点定制所需的编码数据块,从而有效减少了冗余编码数据块的产生。通过仿真实验,从播放延迟、流速率、网络规模等方面对 customR2 和 R2 进行了性能对比,实验结果表明 customR2 相比 R2 具有较小的编码数据块冗余度和较高的节点播放质量。

**关键词:**对等流媒体;网络编码;数据调度;性能优化

**中图分类号:**TP393.7 **文献标志码:**A

### customR2: A hybrid push/pull scheduling method with network coding in P2P live streaming systems

WANG Hui, CHEN Wei-tao, LIU Ya-jie

(College of Information Systems and Management, National University of Defense Technology, Changsha Hunan 410073, China)

**Abstract:** Compared with the traditional pull-based scheduling methods in P2P live streaming systems, the random push scheme with random network coding, named R2, has a lot of advantages such as improving video quality of service, reducing overload of streaming servers and enhancing the resilience to nodes dynamics. However, the random push strategy in R2 is prone to producing a large amount of redundant coded blocks, thus wasting some precious network bandwidth. In this paper, a novel hybrid push/pull scheme with network coding, customR2, was proposed. When one down-streaming node received the most of coded blocks of one segment actively pushed from its up-streaming seeds, it would notify its seeds to stop push operations and start pull operations for the remaining coded blocks, to reduce the quantity of redundant blocks. By the simulations, the comparison between customR2 and R2 was conducted with respect to playback delay, streaming rate and network size. The results show that customR2 outperforms R2 in terms of redundancy ratio of coded blocks and playback quality.

**Key words:** Peer-to-Peer (P2P) stream media; network coding; data scheduling; performance optimization

## 0 引言

近年来,随着互联网对等网络技术(Peer-to-Peer, P2P)和个人计算机处理能力的飞速发展,P2P 流媒体直播系统(如 PPLive、PPStream、UUSee 等)得到了广泛应用,已成为互联网 P2P 应用领域仅次于 P2P 文件下载的第二大应用。P2P 流媒体直播系统的成功得益于其简单的基于 Gossip 协议的成员管理机制和基于 Pull 模式(即拉模式)的随机数据调度策略,使得系统具有较好的扩展性、服务质量以及对 Peer 节点动态的适应能力,但在系统整体吞吐量、播放延迟等方面仍存在不足<sup>[1]</sup>。

网络编码理论<sup>[2]</sup>是进入 21 世纪后网络通信研究领域中的一重要突破,自首次提出以来就得到国内外众多著名大学和研究机构的高度关注和重视,并已经在理论和应用等方面取得了良好的进展。相关研究成果表明,在 P2P 流媒体直播系统<sup>[3-5]</sup>中引入网络编码特别是线性网络编码<sup>[6]</sup>,可在较大程度上提高用户节点的播放质量水平,减少 Peer 节点动态随机性对用户节点播放质量的影响,并可在一定程度上简化数据分发

协议的设计,降低应用协议的通信开销。例如:针对网络编码特别是线性网络编码特有的数据编码/解码过程,同时考虑 P2P 流媒体边下载边播放的要求,相关研究人员采用基于 Pull 模式的数据调度策略,提出了对视频数据先分段后分块,并把网络编码操作局限在数据段内进行的设计方案<sup>[1,3-4]</sup>。在此基础上,H-C Chi 等人<sup>[7]</sup>提出了一种考虑数据播放时限的混合网络编码方法;Wang 等人<sup>[5]</sup>基于网络编码可简化数据调度协议的特点,考虑 P2P 视频直播系统的同步性,提出了一种基于随机 Push 策略(即推模式)的数据调度方法 R2。尤其是 Feng 等人<sup>[8]</sup>采用概率论和连续时间马尔可夫过程分析方法,对 R2 的相关性能(包括播放启动延迟、播放连续性、服务器负载、流速率以及节点收到线性相关数据报文的概率)进行了建模分析,指出采用网络编码的 R2 是迄今为止在系统综合性能指标上表现最佳的 P2P 视频直播方案。

通过对 R2 基于随机网络编码的随机 Push 数据调度方法的深入分析,发现基于随机 Push 的数据调度容易造成 Peer 节点收到过多冗余的编码数据块,消耗过多的节点上传带宽。

收稿日期:2009-08-28。 基金项目:国家自然科学基金资助项目(60872053)。

作者简介:王晖(1968-),男,湖南衡阳人,教授,博士生导师,主要研究方向:信息系统工程、多媒体信息网络; 陈伟涛(1984-),男,河南濮阳人,硕士研究生,主要研究方向:多媒体信息网络; 刘亚杰(1975-),男,湖南常德人,讲师,博士,主要研究方向:多媒体信息网络。

因此,本文结合 Pull 模式的优点,提出了一种基于网络编码的 P2P 流媒体推拉结合数据调度方法 customR2。通过仿真实验验证,customR2 使用推拉结合策略可以改进随机 Push 策略的不足,有效减少冗余数据块的产生,进一步提高 Peer 节点的播放质量。

## 1 R2 的基本原理

图1给出了R2的基本原理<sup>[5]</sup>,节点 $p$ 的上游节点随机选择节点 $p$ 缺少数据块的数据段,完成随机网络编码操作,并将编码数据块主动 Push 给节点 $p$ 。

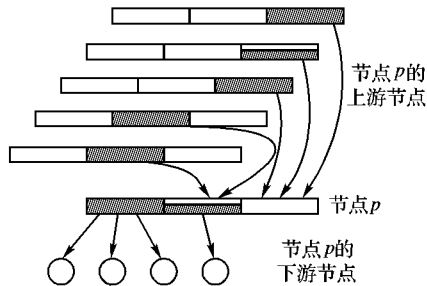


图1 R2 的基本原理

### 1.1 随机网络编码

所谓“随机网络编码”,即将数据段划分成 $n$ 个数据块 $[b_1, b_2, \dots, b_n]$ ,每个数据块 $b_i$ 大小为 $k$ 字节。种子节点(提供数据服务节点)为它的一个下游节点 $p$ 生成编码数据块时,在一个有限域 $GF(t)$ 中独立地随机选择一组编码系数 $[c_1^p, c_2^p, \dots, c_m^p]$ ,其中 $m \leq n$ 。然后从其缓存的所有数据块中(源节点则从所有的源数据块中)随机选择 $m$ 个数据块 $[b_1^p, b_2^p, \dots, b_m^p]$ ,从而生成一个 $k$ 字节的编码块 $x: x = \sum_{i=1}^m c_i^p \cdot b_i^p$ ,其中 $m/n$ 为编码密度。选择较小的编码密度,采用稀疏编码策略可以降低编码复杂性,提高编码速度,同时也能保证较高的解码成功概率。

下游节点 $p$ 收到编码数据块后就存储到自己的缓存区中,同时也为它的下游节点生成新的编码数据块,提供数据服务。为了减少下游节点接收新编码数据块的等待时间,种子节点在其缓存区中拥有不少于 $\alpha \cdot n$  ( $0 < \alpha \leq 1$ )个编码数据块时便能生成新编码数据块。参数 $\alpha$ 称为侵占比, $\alpha$ 越小意味着下游节点能越快得到数据服务,或者说种子节点越是侵占的。

当节点收到完整的 $n$ 个编码数据块 $X = [x_1, x_2, \dots, x_n]$ 后,就可以开始解码过程。首先它把 $n$ 个编码数据块的 $n$ 维系数向量组成一个 $n \times n$ 矩阵 $A$ ,矩阵 $A$ 中的每一行代表一个编码数据块的 $n$ 维系数向量。然后就可以解码出源数据块 $b = [b_1, b_2, \dots, b_n]: b = A^{-1}X^T$ 。使用高斯消元法计算矩阵 $A$ 的逆,而矩阵 $A$ 的逆存在的充要条件是矩阵 $A$ 满秩,即 $n$ 个编码数据块对应的 $n$ 维系数向量线性无关,因此节点收到 $n$ 个线性无关的编码数据块时才能成功解码出源数据块。

### 1.2 随机 Push 数据调度方法

种子节点在执行随机 Push 数据调度前,像 Pull 策略一样,节点之间不定期地交换各自的缓存数据指示信息(Buffer Map, BM),这样种子节点就可以获得它的下游节点所缺少的数据段信息。下游节点只需在它的缓存状态改变时向种子节点发送 BM,即每下载完一个数据段发送一次 BM。

一个种子节点为它的一个下游节点 $p$ 生成编码块时,一

个关键的问题是种子节点选择哪个数据段进行编码。显然,节点 $p$ 已完全接收完的数据段应该排除,因此应该在节点 $p$ 没有完全接收的数据段中选择。在传统的 Pull 策略中,这个选择操作由节点 $p$ 执行,节点 $p$ 向种子节点发送数据段请求;种子节点收到数据请求后发送相应的数据块。随机 Push 策略采用截然不同的调度方式,种子节点在节点 $p$ 没有完全接收的数据段中随机选择一个数据段进行编码,生成一个编码数据块发送到节点 $p$ 上。可见 Push 策略不需要节点 $p$ 发送任何数据请求消息,同时由于随机网络编码的特性,每个编码数据块都是平等的,节点 $p$ 的所有种子节点不需要任何消息协议交换就可以共同协作为节点 $p$ 提供数据服务。

那么种子节点是如何随机选择一个数据段进行编码呢? 一个重要的原则是下游节点 $p$ 中距离播放时刻近的数据段应获得较高的优先级。因此 Push 策略把节点的缓存区分成两部分,如图2所示,其中灰色部分代表缓存区中接收的数据块。在下游节点缓存区的优先区域中,如果存在没有接收完的数据段,那么种子节点等概率地选择一个数据段。从下游节点的角度来看,如果一些缺少的数据段落到优先区域中,那么它的所有种子节点都将选择这些数据段提供数据服务。因此只要下游节点下行带宽充足,一般都会在播放时刻到来前接收完缺少的数据段。

随机 Push 策略能充分发挥随机网络编码的优势,最大化提高系统吞吐量,显著减少 P2P 流媒体的启动播放延迟。

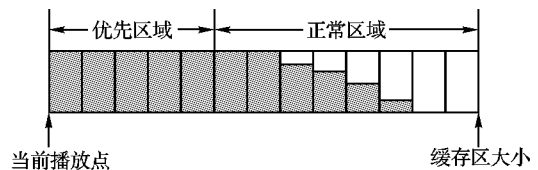


图2 随机 Push 策略缓存区

## 2 R2 冗余数据块的产生及原因分析

通过分析,可以发现随机 Push 策略容易导致下游节点收到一定量的冗余编码数据块,这主要有两方面的原因。

原因1:由于网络延迟等因素,BM 的交换会有一定的延迟。当下游节点已经接收到能够解码某个数据段所需的编码数据块,而其种子节点还没有接收到其最新的 BM,继续选择该数据段发送编码数据块时,下游节点就会收到冗余编码数据块。如图3所示,节点1、2是节点3的种子节点,数据段划分成8个数据块。节点3收到8个线性无关的编码数据块后向节点1、2发送 BM,由于 BM 传输有一定的延迟,节点3又收到2个编码数据块,这2个编码数据块对节点3而言是冗余编码数据块,是要丢弃的,浪费了20%的带宽。

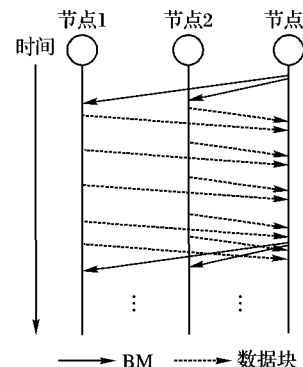


图3 冗余编码数据块产生场景示例1

原因2:多个种子节点对同一个数据段的供应超过了该

数据段的需求,使得下游节点在更新 BM 前已经收到冗余编码数据块。如图4所示,下游节点的某个数据段只需再接收2个编码数据块,但却有4个种子节点同时选择该数据段发送编码数据块,导致下游节点收到2个冗余编码数据块。

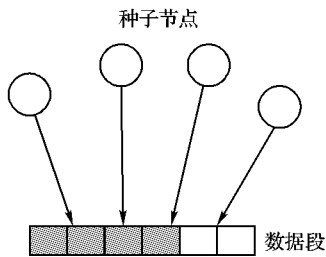


图4 冗余编码数据块产生场景示例2

### 3 基于网络编码的推拉结合数据调度方法

为了解决 R2 随机 Push 策略产生冗余编码数据块的问题,本文提出一种推拉结合的数据调度方法 customR2。

#### 3.1 定制阈值的引入

所谓“定制阈值”,是指下游节点所需的某个数据段缺少的编码数据块数。推拉结合策略中包括推和拉两种工作模式:推模式与 R2 的随机 Push 策略相同;而拉模式则截然不同,在拉模式下,节点向它的种子节点定制编码数据块,从而有效减少了冗余编码数据块的产生。

#### 3.2 customR2 的基本原理

customR2 需要解决的问题是:1)种子节点如何判断它的下游节点的数据段是否满足定制阈值;2)当数据段满足定制阈值时,种子节点是否为其提供服务。

为了解决第一个问题,customR2 需要改变节点发送的 BM 格式。随机 Push 策略的 BM 仅仅用 0、1 代表缓存区中数据段的接收状态就可以了。customR2 使用推拉结合的策略,需要种子节点判断它的下游节点的数据段是否满足定制阈值,因此用各个数据段已经接收的编码数据块数作为 BM。节点发送 BM 的时机也相应地做出改变:下载完一个数据段或者数据段满足定制阈值时,发送一次 BM。

每个 Peer 节点主要有两种操作:发送数据和接收数据。发送数据时节点为其他节点的种子节点,接收数据时节点为其他节点的下游节点。下面通过介绍节点的这两种操作来解决第二个问题。

节点作为其他节点的下游节点接收数据时,节点在下载完一个数据段或者某个数据段满足定制阈值时,向它的种子节点发送 BM。数据段满足定制阈值时,节点把该数据段缺少的数据块数进行划分。为简单起见,本文采用等分策略,引入数据划分单元(Data Division Unit, DDU)的概念,每份含有 DDU 个数据块,但最后一份可以不足 DDU 个数据块。划分完毕后随机选择若干种子节点(种子节点数等于数据块划分份数),分别向它们分配一份定制任务,为了减少延迟,定制请求和 BM 一起发送。如图5所示,节点3的种子节点为节点1和节点2,数据段划分为10块,定制阈值为2,即数据段缺少的数据块数为2时,节点转为定制数据。初始时节点1、2采用 Push 策略向节点3提供数据。当节点3接收了8个数据块后,还有2个数据块未接收,满足定制阈值,因此节点3向节点1、2发送 BM,通知它们停止 Push 数据,同时节点3对未接收的数据块进行划分,这里设定 DDU = 1,因此划分成两份(1,1),向节点1、2分别定制1个数据块。节点1、2收到节

点3的定制请求后,根据定制情况,向节点3发送数据块。

节点作为种子节点向某个下游节点发送数据时,具体过程如下:

步骤1 根据该下游节点发送的 BM,检查是否有数据段满足定制阈值,若没有,转步骤2,否则,转步骤3;

步骤2 采用 Push 策略,从该下游节点缺少的数据段中随机选择一个数据段,生成一个编码数据块进行发送;

步骤3 对于满足定制阈值的数据段,检查是否收到相应的定制请求,若没有,转步骤4,否则转步骤5;

步骤4 在不满足定制阈值并且不完整的数据段中,采用 Push 策略,随机选择一个数据段,生成一个编码数据块进行发送;

步骤5 根据该数据段的定制请求发送数据块,发送的数据块数由定制请求决定。

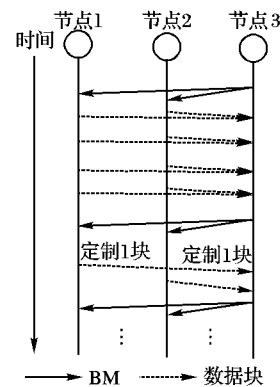


图5 下游节点接收定制数据块示例

### 4 仿真实验与结果分析

为验证 customR2 的有效性,本文设计了一个 P2P 流媒体离散仿真器,实现了类 R2 的随机 Push 策略和 customR2 数据调度策略。设置比较仿真实验如下:生成一个 100 个节点组成的随机拓扑,每个节点从中选择 6 个节点作为自己的邻居节点,服务器节点选择其中的 16 个节点直接发送数据。仿真中设置服务器带宽为 1 Mbps,节点上行带宽为 80 ~ 100 Kbps,下行带宽没有限制,流速率为 64 Kbps。仿真实验以秒为单位向前推进,每次推进过程中,节点运行相应的数据调度算法。

本文引入编码数据块冗余度的评价指标,它定义为节点在一段时间内收到的多余编码数据块个数与所有数据块个数的比率。同时,引入播放连续度性能指标,它定义为节点在一段时间内得到播放(即在播放期限之前实现完全解码)的数据段个数与应该播放的数据段个数的比率。

通过实验结果表明<sup>[5]</sup> R2,当每个数据段播放时长为 4 s,并分为 132 块时,节点的播放质量最好。因此,本文参照 R2 的实验结论,设置每个数据段播放时长为 4 s,其数据分为 128 块。另外设置节点的播放启动延迟为 16 s,编码密度为 8%。每次仿真实验运行 600 s 终止,重复 10 次,结果取平均值。实验结果与 R2 进行比较。

customR2 采用推拉结合策略是通过设定定制阈值来实现的,那么设定什么样的定制阈值和 DDU,能最大限度地减少节点接收的编码数据块冗余度呢?下面通过仿真实验来确定,图6给出了实验结果,其中定制阈值为0时表示采用 Push 策略,代表 R2。

从图6(a)可以看出,定制阈值越大,节点冗余度越低,这

是因为定制阈值越大,意味着节点可以越早转为拉模式,向种子节点定制数据,从而有效减少了冗余数据块的产生;从图中还可以看出,设定较大的 DDU 可以减少冗余数据块的产生。当定制阈值达到 32,设定  $DDU = 5$  时,节点的冗余度可以控制在 0.3%,因此在后续的实验中,设置定制阈值为 32(数据段缺少的数据块数为 32),DDU 为 5。

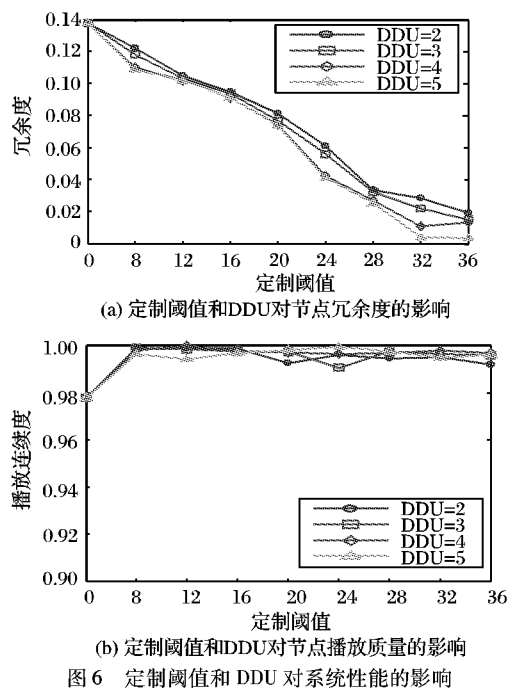


图6 定制阈值和 DDU 对系统性能的影响

图 6(a) 的实验结果也表明推拉策略仍然会产生一定量的冗余数据块。这主要有两方面的原因:1) 定制请求的传输有一定的延迟,在这个延迟时间内,节点有可能收到种子节点发送的数据块;2) 数据段接近定制阈值时,多个种子节点同时选择了该数据段发送数据块,并且总的发送数据块数超过了该数据段的需求,即节点在发送定制请求前已经收到一定量的冗余数据块,这也是本文在后续实验中设定较大定制阈值的原因。

图 6(b) 给出了不同定制阈值和 DDU 下节点的播放质量,可以看出,采用推拉结合策略比纯 Push 策略进一步提高了节点的播放质量。这是因为播放时刻早的数据段获取数据的优先级高,可以较早满足定制阈值,而对于满足定制阈值的数据段,种子节点会优先发送,从而进一步提高了这类数据段的优先级,因此相比纯 Push 策略进一步提高了节点的播放质量。

图 7 给出了播放延迟对系统性能的影响。从图 7(a) 可以看出,相比于 R2, customR2 在不同的播放延迟下都具有较小的编码数据块冗余度。图 7(b) 给出了播放延迟对节点播放质量的影响,直观上,较大的播放延迟可以使使得节点有充足的时间缓存数据,从而获得较高的播放质量,但是从图 7(b) 可以看出,customR2 和 R2 受播放延迟的影响不大,都能保持较高的播放质量,这充分说明了 Push 策略的优越性,同时由于 customR2 改进了 R2 的 Push 策略的不足,从而进一步提高了节点的播放质量。可见 customR2 可以适应不同的播放延迟,在较小的播放延迟下也能达到很高的系统性能。

图 8 给出了不同流速率对系统性能的影响。从图 8(a) 可以看出,不同流速率下 customR2 的编码数据块冗余度都比 R2 小;流速率较小时(流速率 = 40 Kbps),由于节点间有充足

的网络带宽资源,两者的编码数据块冗余度都较大。从图 8(b) 可以看出,不同流速率下 customR2 的节点播放质量都高于 R2;在流速率小于 70 Kbps 时,节点间有足够的带宽资源传输流速率,两者都能保持较好的播放质量;当流速率较大时(流速率 = 80 Kbps),两者的播放质量明显下降,这是节点间的带宽资源紧张所致。

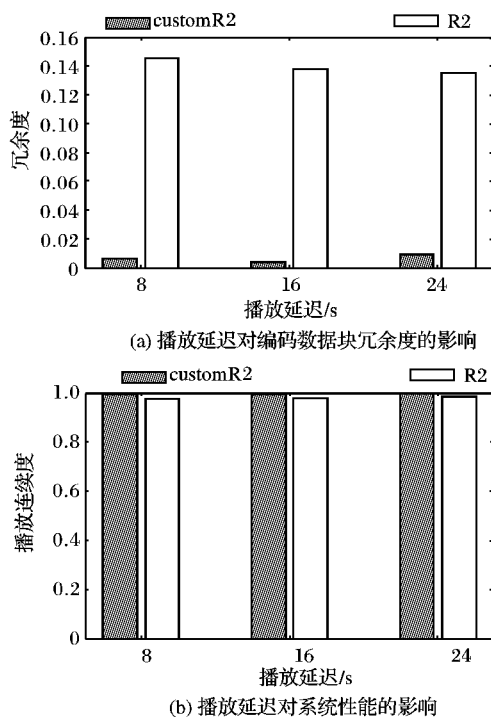


图7 播放延迟对系统性能的影响

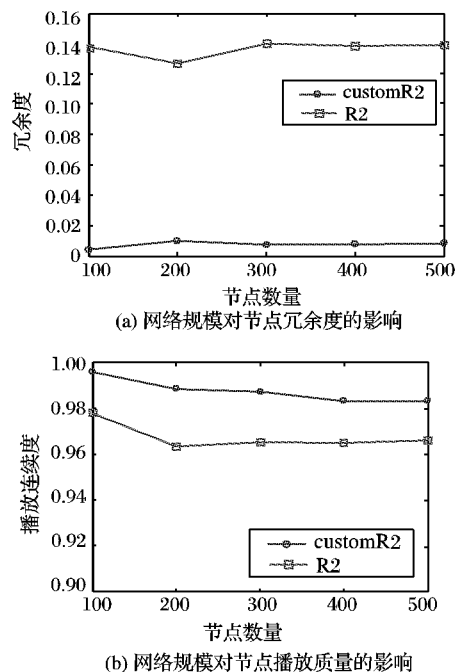


图8 流速率对系统性能的影响

图 9 给出了网络规模对系统性能的影响,为了评估 customR2 的可扩展性,节点数量从 100 变化到 500。从图 9(a) 可以看出,在不同节点数量下,customR2 节点冗余度都保持在 1% 以内,相反 R2 的冗余度都在 13% 以上;从图 9(b) 中可以看出,customR2 在不同网络规模下进一步提高了节点的播放质量,因此 customR2 具有良好的可扩展性。

(下转第 302 页)

“能量热点”问题。结合数据路由进一步实现全网能耗均衡是下一步的工作。

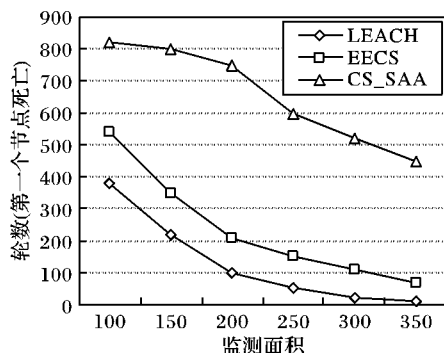


图5 监测面积与轮数(第一个节点死亡)

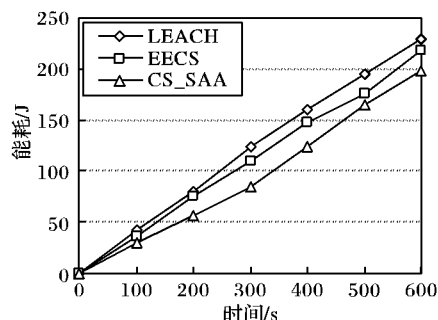
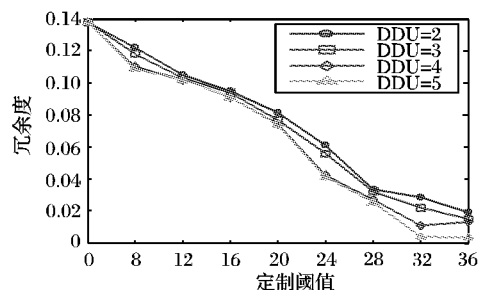


图6 时间与能量消耗

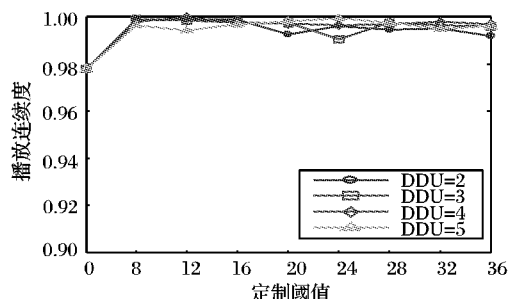
#### 参考文献:

- [1] ROMER K, MATTERN F. The design space of wireless sensor networks [J]. IEEE Wireless Communications, 2004, 11(6): 54-61.
- [2] YOUNIS O, FAHMY S. HEED: A hybrid, energy-efficient, distributed clustering approach for Ad Hoc sensor networks [J]. IEEE Transactions on Mobile Computing, 2004, 3(4): 366-379.
- [3] HEINZELMAN W R, CHANDRAKASAN A, BALAKRISHNAN H. Energy-efficient communication protocol for wireless micro sensor networks [C]// Proceedings of the 33rd Annual Hawaii International Conference on System Sciences. Maui: IEEE Computer Society, 2000: 3005-3014.
- [4] MAO YE, LI CHENG-FA, CHEN GUI-HAI, *et al.* EECS: An energy efficient clustering scheme in wireless sensor networks [C]// IPCCC 2005: Proceedings of the IEEE International Performance, Computing, and Communications Conference. New York: IEEE Press, 2005: 535-540.
- [5] 申帅, 林亚平, 胡玉鹏, 等. 无线传感器网络中一种全局节能的聚合树构建算法[J]. 2009, 29(4): 1068-1071.
- [6] 马云刚, 周群彪. 基于 LEACH 的无线传感器网络节能算法[J]. 计算机应用, 2009, 29(6): 1514-1516.
- [7] NICULESCU D, NATH B. Ad-hoc Positioning System (APS) using AOA [C]// INFOCOM 2003: 22nd Annual Joint Conference of the IEEE Computer and Communications Societies. San Francisco: IEEE Press, 2003, 3: 1734-1743.
- [8] 康立山, 谢云, 尤矢勇, 等. 非数值并行算法(第一册)——模拟退火算法[M]. 北京: 科学出版社, 1994: 22-124.
- [9] XIAN XIAO-DONG, SHI WEI-REN, HE HUANG. Comparison of OMNET++ and other simulator for WSN simulation [C]// ICIEA 2008: 3rd IEEE Conference on Industrial Electronics and Applications. Washington, DC: IEEE Press, 2008: 1439-1443.

(上接第288页)



(a) 定制國值和DDU对节点冗余度的影响



(b) 定制國值和DDU对节点播放质量的影响

图9 网络规模对系统性能的影响

## 5 结语

基于网络编码的随机 Push 策略会产生一定量的冗余数据块,本文分析了该问题产生的原因,针对该问题,对随机 Push 策略进行了改进,提出了一种推拉结合的网络编码数据块调度方法 customR2,其主要思想是当下游节点某个数据段所需的编码数据块将要接收完成时,通知它的种子节点停止向它 Push

该数据段编码数据块,转为向它的种子节点定制所需的编码数据块,从而有效减少了冗余编码数据块的产生。通过仿真实验,从播放延迟、流速率、网络规模等方面对 customR2 和 R2 进行了性能对比,实验结果表明 customR2 相比于 R2 具有较小的编码数据块冗余度和较高的节点播放质量。

#### 参考文献:

- [1] WANG M, LI B C. Lava: A reality check of network coding in peer-to-peer live streaming[EB/OL]. [2009-02-10]. <http://www.eecg.toronto.edu/~bli/papers/mwang-infocom07.pdf>.
- [2] AHLSEDE R, CAI N, Li S-Y R, *et al.* Network information flow[J]. IEEE Transactions on Information Theory, 2000, 46(4): 1204-1216.
- [3] 刘亚杰. P2P 流媒体内容分发关键技术研究[D]. 长沙: 国防科学技术大学, 2005.
- [4] LIU Y, PENG Y, DOU W, *et al.* Network coding for P2P live media streaming[C]// Proceedings of 5th International Conference on grid and Cooperative Computing. Washington, DC: IEEE Computer Society, 2006: 392-398.
- [5] WANG M, LI B C. R2: Random push with random network coding in live peer-to-peer streaming[J]. IEEE Journal on Selected Areas in Communications, 2007, 25(9): 1655-1666.
- [6] LI S-Y R, YEUNG R W, CAI N. Linear network coding[J]. IEEE Transactions on Information Theory, 2003, 49(2): 371-381.
- [7] CHI H-C, ZHONG Q. Deadline-aware network coding for video-on-demand service over P2P networks[J]. Journal of Zhejiang University Science, 2006, 7(5): 775-763.
- [8] FENG C, LI B. On large-scale peer-to-peer streaming systems with network coding[C]// Proceedings of ACM Multimedia. New York: ACM, 2008: 26-31.