

# 无线多播中基于网络编码的 QoS 研究<sup>①</sup>

郑新建

(浙江工业大学 计算机科学与技术系, 杭州 310023)

**摘要:** 在无线网络中使用网络编码可以提高吞吐量和降低能耗。近来, 有许多工作在研究如何利用网络编码来提高吞吐量, 但是很少有考虑到 QoS。本文重点研究在实时无线网络中数据包在延时约束条件下, 网络编码的广播调度问题, 目标是在数据包的误时率(deadline miss ratio)容许范围内, 减少重传包的数量。我们将综合两种编码调度方案, 阐释各自的优劣, 动态地选取编码方案。仿真结果显示, 我们的算法可以有效减少重传包的数量。

**关键词:** 无线网络; 网络编码; QoS; 时延; 重传包

## Network Coding for QoS in Wireless Multicast

ZHENG Xin-Jian

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

**Abstract:** Using network coding in wireless networks can increase throughput and reduce energy consumption. Recently there are many works focusing on how to utilize network coding to increase throughput while there are few works considering the quality of service. This paper focuses on network coding based broadcast scheduling in wireless networks with the packet delay constraint and aims at minimizing the number of retransmissions. We will combine two encoding and scheduling schemes, analyze their respective advantages and disadvantages, and dynamically select the scheme. Simulation results show that our algorithm can reduce the number of retransmissions.

**Key words:** wireless networks; network coding; QoS; delay; retransmissions

## 1 引言

随着近年来 WiFi, 3G 通信以及移动设备的采用, 无线网络变得越来越普遍。有许多工作研究如何在无线网络中利用网络编码来提高吞吐量和降低能耗。近来的一些商用无线服务的发展产生大量有关信息流传输的要求, 比如多媒体、音频以及视频都要求很高的 QoS 保证。数据包的时延保证是非常重要的 QoS 指标, 但是很少有工作在研究中考虑到。在实时应用中, 每个数据包都有一个时延时限, 如果接收节点接收这个包的时间超过这一时限, 那么这个包对于接收节点将没有用处。举例来说, 在一些无线金融服务中, 一些用户需要收集最近几分钟(甚至最近几秒钟)内的股票报价, 以便对动态迅速变化的市场做出相应的反应。因此, 在采用网络编码思想的网络中, 时延问题无疑是一个关键性问题。可以发现, 有些应用(如视频)对时

延具有一定的容忍度, 即在一定范围内的误时率 (deadline miss ratio) 对该应用的效果影响不大。我们研究的便是在数据包的误时率容许范围内, 减少重传包的数量, 从而在保证时延这一重要指标的同时, 提高网络吞吐量。

我们提出一种编码调度方案——基于缓存的多播重传编码, 然后对比另一种编码方案——动态多播重传编码, 通过动态地选取编码方案, 在数据包的误时率容许范围内, 减少重传包的数量。我们将这种方法称为择优重传编码。下面先简单说明两种编码方案各自的特点。动态多播重传编码下的接收节点在接收重传编码包后会对其进行解码操作, 如果不能解码出想要的数据包, 会将之丢弃。基于缓存的多播重传编码下的接收节点在接收重传包后也会对其进行解码, 但无论能否解码出想要的数据包, 接收节点都会将其存

<sup>①</sup> 收稿时间:2013-05-13;收到修改稿时间:2013-06-24

储; 当后面的重传包到达时, 可结合之前存储的重传包进行解码. 下面通过一个例子简要说明.

考虑一个单跳多播无线网络, 如图 1 所示, 该网络有一个发送节点  $s$  及 6 个接收节点  $r_1, r_2, r_3, r_4, r_5, r_6$ . 假设  $s$  已经发送了 6 个数据包  $p_1, p_2, p_3, p_4, p_5, p_6$  至接收节点. 由于无线链路损耗, 接收节点可能只成功接收部分数据包, 如下图所示, 根据接收节点发回的 ACK, 发送节点  $s$  可知  $r_1$  丢失了  $\{p_1\}$ ,  $r_2$  丢失了  $\{p_2, p_3, p_4\}$ ,  $r_3$  丢失了  $\{p_4, p_5, p_6\}$ ,  $r_4$  丢失了  $\{p_2, p_3, p_5\}$ ,  $r_5$  丢失了  $\{p_2, p_4, p_6\}$ ,  $r_6$  丢失了  $\{p_3, p_5, p_6\}$ .

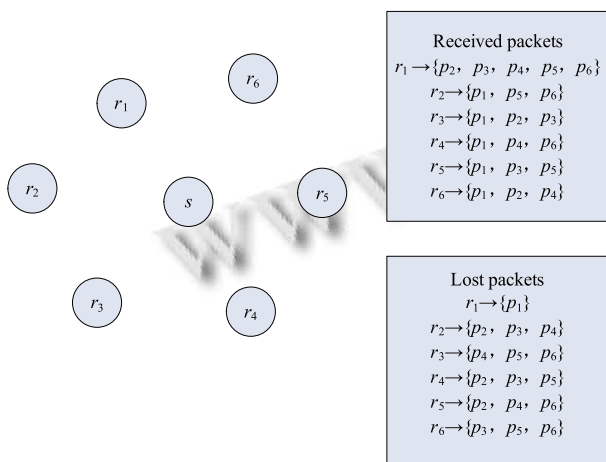


图 1 单跳多播无线网络

假设传输一个数据包需花费一个单位时间. 如果  $p_1, p_2, p_4$  的时延时限(deadline)为两个单位时间,  $p_3, p_5$  为三个单位时间,  $p_6$  为四个单位时间. 在没有编码情况下, 将有 6 个数据包错过它们的时延时限. 在动态多播重传编码情况下,  $s$  分别发送  $p_1 \oplus p_6, p_2 \oplus p_4, p_3 \oplus p_5, p_2 \oplus p_5$ . 有 1 个数据包错过它们的时延时限, 为发往  $r_1$  的  $p_1$ . 在基于缓存的多播重传编码情况下,  $s$  发送  $p_1 \oplus p_2 \oplus p_6, p_3 \oplus p_5 \oplus p_6, p_4 \oplus p_5$ , 有 3 个数据包错过它们的 deadline, 为分别发往  $r_2, r_3, r_5$  的  $p_4$ . 如果要求错过自身时延时限的数据包个数不超过 2 个, 那么我们会选择动态多播重传编码. 如果错过自身时延时限的数据包个数可以超过 2 个, 那么我们会选择基于缓存的多播重传编码, 因为它只需发送 3 个重传编码包.

本文将综合上述两种编码方案的优缺点, 在编码包发送前, 对比两种编码法案的效果, 动态地选择, 目的是在保证时延要求的前提下, 尽可能减少重传包的数量.

## 2 相关工作

无线广播因其在无线网络中的固有属性, 使其通过网络编码来提高网络吞吐量和能量效率具备可行性. 文献[1-2]研究了通过网络编码来获取网络吞吐增益. 文献[3]研究了在非可靠无线网络中, 在使用或者未使用网络编码技术的条件下, 一个多播集所能够达到的最大吞吐量. 文献[4]提出一个基于次阶梯的分布式算法来实现最优的多播吞吐量. 文献[5]研究了在高效节能的无线 ad hoc 网络中使用网络编码技术. 文献[6]结合线性规划与网络编码研究了在无线 ad hoc 网络中的最低能耗多播.

最近, 有一些工作描绘了在有损网络中网络编码带来的可靠性效益. 文献[7]显示在数据包消除或无消除的无线网络中, 随机分布网络编码是渐进最优的. Lun 等人在文献[8]中提出了一个针对有损网络的单播和多播的容量接近编码方案, 其节点都使用随机线性编码. M. Ghaderi 等人在文献[9]中给出了基于 ARQ 和网络编码可靠性机制性能的严格渐近界.

上述的大部分工作都使用随机线性编码. 鉴于无线网络中节点的计算能力是有限的, 我们将使用 XOR 操作进行编码, 从而使得发送节点的编码以及接收节点的解码都更加容易. 文献[10]研究了在无线网状网络中使用 XOR 操作的网络编码来提高网络吞吐量. 文献[11]研究证明了最优广播编码选择问题是一个 NP-难问题. 文献[12]研究了在可靠性广播中使用 XOR 操作编码.

文献[13]研究了如何最小化解码延时. 文献[14]提出了在重传过程中使用网络编码可以达到的理论吞吐上限. 该研究假设发送节点在发送一个数据包的同时, 该数据包的确认可以在同一时隙内发回至发送节点. 文献[15]研究利用网络编码来降低重传包的数量.

只有极少数工作考虑了数据包的时延保证, 这个对于实时应用特别重要的指标. 文献[16]考虑了数据包接收的时延约束, 并提出一个编码方案, 用以最小化超过时延时限数据包的数量.

与上述工作不同, 我们的研究着眼于在保证时延限制的前提下, 尽可能减少重传包的数量, 并研究基于接收节点反馈的编码策略.

本文安排了以下章节对上述内容进行详细说明, 具体的文章结构安排如下. 第 3 节提出基于缓存的多播重传编码方案, 计算分析该方案及文献[16]提到

weighted clique coding(本文称为动态多播重传编码)在时延与吞吐量方面的代价,并择优作为最终的编码方案.第4节对仿真结果进行说明.第5节进行全文总结.

### 3 择优重传编码

在这部分,我对要处理的问题进行简单陈述,然后提出我们的编码方案并计算其产生的误时率,最后与动态多播重传编码进行比较,择优选取.

#### 3.1 问题陈述

一个单跳无线网络中,有一个发送节点  $s$  以及  $n$  个接收节点  $R=\{r_1, r_2, \dots, r_n\}$ .  $s$  已经发送了  $m$  个数据包  $P=\{p_1, p_2, \dots, p_m\}$  至接收节点.用  $H(r_i)$  表示  $r_i$  成功接收的数据包集合,  $L(r_i)=P-H(r_i)$  表示  $r_i$  丢失的数据包.对于每一个  $p_j \in L(r_i)$ ,  $T_{ij}$  表示其对应的时延时限(deadline).假设时间是分时隙的,在每一个时隙发送节点可以发送一个数据包.如果  $p_j$  到达  $r_i$  的时间超过了  $T_{ij}$ ,那么  $p_j$  对于  $r_i$  而言就是无用的,也就是  $r_i$  所需要的  $p_j$  超过了它的时延时限.定义误时率(deadline miss ratio)表示数据包超过时延时限的百分比:

$$\frac{\text{超过时延时限的数据包的总数}}{\text{接收节点丢失的数据包总数}} \times 100\%$$

其中,接收节点丢失的数据包总数即为重传过程中接收节点需要恢复的数据包总数.

我们的问题是,给定了时延约束条件即不可超过的误时率  $DMR$ ,接收节点成功接收的数据包集合  $H(r_i)$ ,  $1 \leq i \leq n$ ,丢失的数据包集合  $L(r_i)$  以及相应的时延时限  $T_{ij}$ ,  $1 \leq i \leq n$ ,  $1 \leq j \leq m$ ,决定在每一个传输机会如何进行数据包的编码及传输,从而在满足给定时延约束的条件下,发送的重传编码包最少.

#### 3.2 基于缓存的多播重传编码

动态多播重传编码下的接收节点,当接收到的编码包  $p'$  无法立即解码时,会将之丢弃.而基于缓存的多播重传编码下的接收节点会存储那些暂时无法解码的编码包,因此编码包的总数会减少.

##### 3.2.1 数据包编码

基于缓存的多播重传编码是在利用最少的重传包恢复所有丢失的数据包的同时,尽可能让数据包在时延时限之前被恢复.现将编码算法进行简要说明.

$|L(r_i)|$  表示  $L(r_i)$  的个数,即  $r_i$  所丢失的数据包的个数.令  $l_{\max} = \max_{r_i \in R} |L(r_i)|$ ,即  $l_{\max}$  表示所有接收节点中丢失数据包最多的那个节点所丢失的数据包的个数.

由于接受节点最多可从一个 XOR 编码包中解码出一个原始包,所以任何算法都至少需要  $l_{\max}$  个重传包,才能使所有接收节点成功解码出丢失的数据包.算法分为两步.第一步,构造  $l_{\max}$  个编码包,每个编码包都由若干原始包通过 XOR 操作生成,最多只有一个原始包从一个接收节点丢失的数据包集中挑选.利用这些编码包,有些接收节点可能无法恢复出所有丢失的数据包.因此,在第二步,需要再发送一些原始包以确保所有的接收节点都能恢复所丢失的数据包.

第一步,构造  $l_{\max}$  个重传编码包.根据前面关于图  $G(V, E)$  的定义,对于任意  $i, 1 \leq i \leq n$ , 如果  $j \neq k, (v_{ij}, v_{ik}) \in E(G)$ . 可以将  $V$  分成  $n$  个子集合  $\{V_1, V_2, \dots, V_n\}$ , 其中  $V_i = \{v_{ij} | p_j \in L(r_i), 1 \leq j \leq m\}$  表示接收节点  $r_i$  所丢失的数据包的集合.构造算法的大致思路是,从每一个  $V_i$  中挑选一个顶点对应于一个丢失的数据包,组成一个重传编码包.对于每一个  $V_i$ , 挑选权值最大的那个顶点,并在图  $G$  中将其删除.在算法的最后,会得到  $l_{\max}$  个重传编码包.

依据文献[15]可知,第一步得到了重传编码包集  $P'$ ,为了使所有的接收节点恢复出丢失的数据包,还需要再增加一些明文包,即  $Q'$ .

结合第一步得到的重传编码包  $P'$  和第二步得到的重传明文包  $Q'$ ,便有了重传包  $P' \cup Q'$ .而重传包的发送顺序,先是  $P'$  按构造的先后顺序发送,再是  $Q'$  依时延时限由小到大发送.

##### 3.2.2 计算误时率

下面计算以此重传包发送的误时率.用  $y$  计算超过时延时限的数据包的数量,那么其误时率为  $\frac{y}{|V(G)|}$ .

对于接收节点  $r_i$ ,当接收一个重传包  $p_k'$ ,用  $P_k'$  表示编码  $p_k'$  的数据包集合.定义一个接收矢量  $v_{ik}=(a_1, a_2, \dots, a_m)$ ,当  $p_j \in L(r_i)$  且  $p_j \in P_k'$  时,  $a_j=1$ ; 否则,  $a_j=0, 1 \leq i \leq n, 1 \leq j \leq m$ .

对于任意顶点  $v_{ij}$ , 其时延时限为  $T_{ij}$ , 假设  $r_i$  丢失了  $k$  个数据包,  $r_i$  可基于接收矢量构造出一个矩阵

$M_{T_{ij} \times m}^i$ , 如果  $p_j \in L(r_i)$ , 在矩阵  $M_{T_{ij} \times m}^i$  中保留第  $j$  列, 否则, 删除第  $j$  列. 经此变换, 得到  $T_{ij} \times k$  的子矩阵  $M_{T_{ij} \times k}^i$ .

对矩阵  $M_{T_{ij} \times k}^i$  应用高斯消元, 将其化为行最简形矩阵.

考虑是否存在某一行只有第  $j$  列元素不为 0, 其他元素

全为 0, 由线性代数方法易知如果存在则  $r_i$  可以根据前  $T_{ij}$  个重传包恢复出  $p_j$ , 反之无法恢复. 算法如下:

```

1:  $y \leftarrow 0$ ;
2: for ( $I \leftarrow 1$  to  $n$ ) do
3: for ( $j \leftarrow 1$  to  $m$ ) do
4: If ( $v_{ij} \in V(G)$ ) then
5:  $k \leftarrow IL(r_i)$ ; ( $L(r_i) = \{P_{i_1}, P_{i_2}, \dots, P_{i_k}\}$ )
6: Construct matrix  $M_{T_{ij} \times m}^i$  and  $M_{T_{ij} \times k}^i$  based on  $T_{ij}$ 
   sending vectors of  $s$ ;
7: Do Gaussian elimination on  $M_{T_{ij} \times k}^i$  to be The
   minimalist form matrix;
8: if (there is no row where the  $j$ -th Column is the only
   column which is non-zero) then
9:  $y \leftarrow y + 1$ ;
10: break;
11: end if
12: end if
13: end for
14: end for

```

### 3.3 择优重传编码

我们在引言中用一个例子介绍了动态多播重传编码和基于缓存的多播重传编码. 从例子中可以看出通过网络编码可以显著降低数据包的误时率, 而两种编码方案之间各有优劣, 下面对此进行分析.

动态多播重传编码的优点是编码包发送至接收节点时, 接收节点可以马上解码出其所要的编码包, 有利于数据包在其时延时限(deadline)之前被接收; 缺点是发送的编码包总数相比基于缓存的多播重传编码较多.

基于缓存的多播重传编码的优点是所发送的编码包总数较动态多播重传编码少; 缺点是有些编码包在到达接收节点时, 不能马上解码, 需等待足够多的编码包到达之后才能解码出想要的数据包, 使得误时率相比动态多播重传编码较高.

我们将结合两种方案的优点, 在不同的时延约束条件下选择相应的编码方案, 从而使得发送的重传编码包最少.

通过上述计算可以得到基于缓存的多播重传编码产生的误时率与所需发送的重传包数量, 以及利用文献[16]的方法可以得到动态多播重传编码的误时率与重传包数量. 结合给定的误时率  $DMR$ , 如果只有一种方案的误时率小于  $DMR$ , 则选择该方案做为

编码方案; 如果两种方案的误时率都小于  $DMR$ , 则选择重传包数量较少的方案作为编码方案. 如果两种方案的误时率都大于  $DMR$ , 则选择误时率较小的方案作为编码方案.

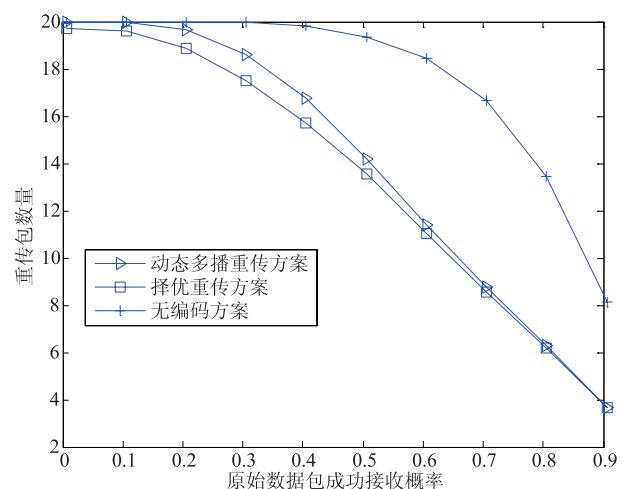
## 4 仿真

我们建立了一个仿真环境, 并做了几个实验, 以评估我们提出的算法的性能.

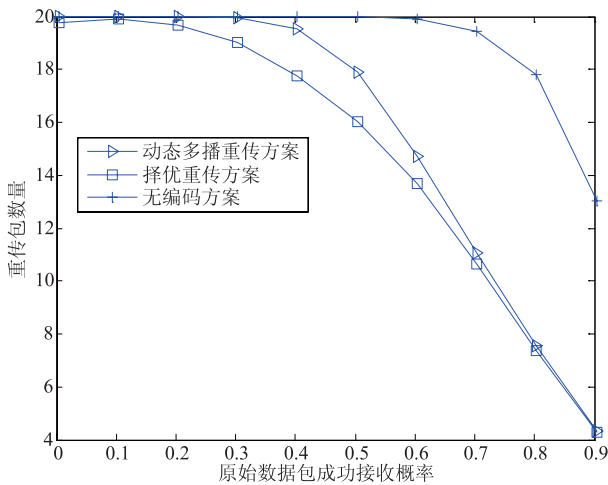
仿真环境如图 1 所示, 由一个发送节点和若干个接收节点组成. 发送节点发送  $m$  个数据包至  $n$  个接收节点, 为了模拟数据包在接收端的丢失, 用数据包成功接收率(packet delivery ratio)随机产生接收节点丢失的数据包集合  $L(r_i)$ . 每个数据包  $p_j$  被成功接收的概率为  $g$ , 丢失的概率为  $1-g$ . 在仿真中, 设定所有的接收节点的数据包成功接收率相同, 都为  $g$ . 数据包的时延时限在区间[1,20]均匀分布. 编码方案的误时率不可超过  $DMR$ .

将我们提出择优重传算法与<sup>[16]</sup>提出的动态多播重传算法以及无编码条件下的传统算法进行比较. 在每个仿真场景, 我们仿真 1000 个实例并报告它们的平均性能.

图 2 显示了当  $m=20$ ,  $DMR=0.6$  时, 使用网络编码带来的网络增益, 我们在图 2(a)中设定  $n=5$ , 在图 2(b)中设定  $n=10$ . 在图 2 中, 可以看出, 我们的算法优于动态多播重传算法, 动态多播重传算法优于无编码算法. 如图 2 所示, 随着  $g$  增大, 重传数据包随之减少. 因为当  $g$  增大时, 丢失的数据包减少了, 同时解码的机会增加了.



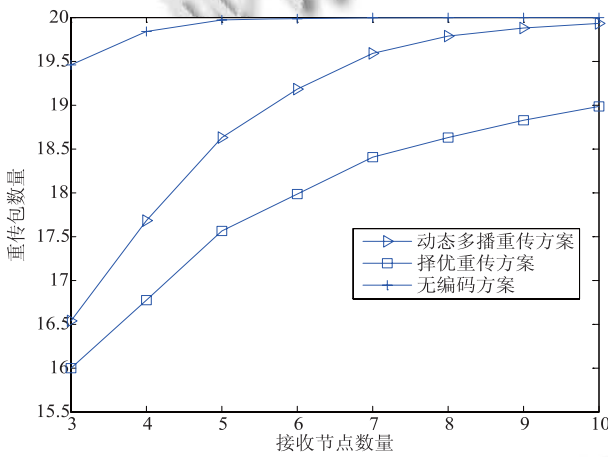
(a)  $n=5$



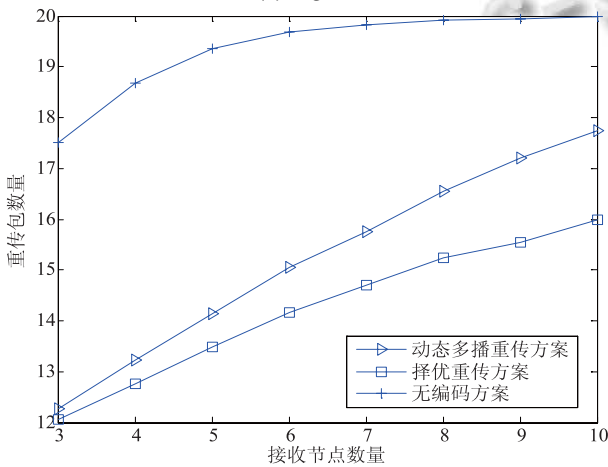
(b)  $n=10$

图 2 重传包数量随数据包成功接收率变化曲线

图 3 显示了当  $m=20, DMR=0.6$  时, 重传包数量随  $n$



(a)  $g=0.3$



(b)  $g=0.5$

图 3 重传包数量随接收节点数量变化曲线

增长的变化情况. 我们在图 3(a)中设定  $g=0.3$ , 在图 3(b)中设定  $g=0.5$ . 如图 3 所示, 当接收节点的数量增加时, 重传包数量也随之增加. 因为有更多的数据包丢失了.

图 4 显示了当  $m=20, n=5$  时, 误时率随  $g$  增长的变化情况. 在图 4 中, 可以看出, 随着  $g$  增大, 误时率随之减小. 因为当  $g$  增大时, 丢失的数据包减少了, 同时解码的机会增加了. 但是在基于缓存的多播重传方案中,  $g=0$  的误时率较小. 因为当  $g=0$ , 即所有数据包全部丢失时, 为保证接受节点解码所有数据包, 发送节点需要发送的重传包较少, 且在接收过程中解码数据包的几率较大.

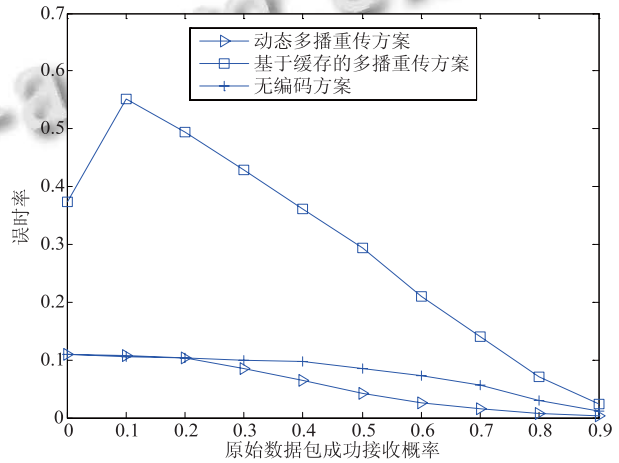


图 4 误时率随数据包成功接收率变化曲线

图 5 显示了当  $m=20, g=0.5, n=5$  时, 重传包数量随 DMR 增长的变化情况. 当 DMR 较小时, 择优编码方案较多地选择动态多播重传编码作为编码方案, 因此重传包数量与动态多播重传编码相近. 当 DMR 增大后, 择优编码方案较多地选择基于缓存的多播重传编码, 因此重传包数量少于动态多播重传编码.

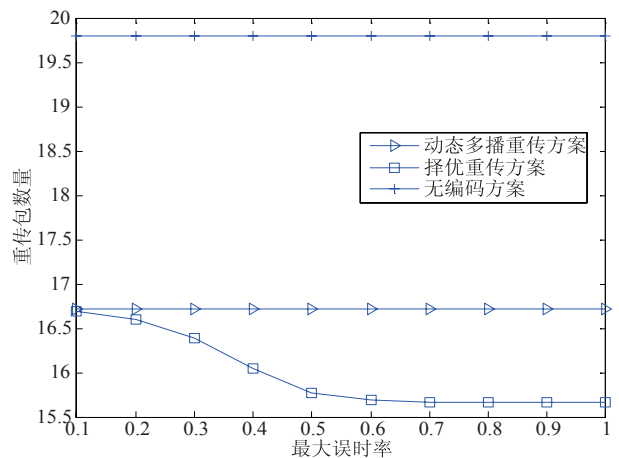


图 5 重传包数量随最大误时率变化曲线

## 5 结论

在无线网络中使用网络编码可以进一步提高网络吞吐量. 本文关注在满足时延约束条件下, 通过编码的选择使得发送的重传编码包最少. 提出了基于缓存的多播重传编码, 并分析了它与动态多播重传编码各自的优劣. 通过动态的选取编码方案, 在满足时延约束条件下, 减少了重传包的数量. 仿真结果显示, 该算法可以有效减少重传包的数量.

### 参考文献

- 1 Sagduyu YE, Ephremides A. Joint scheduling and wireless network coding. Proc. WINMEE, RAWNET and NETCOD 2005 Workshops. Apr. 2005.
- 2 Wu Y, Chou PA, Zhang Q, Jian K, Zhu W, Kung SY. Network planning in wireless ad hoc networks: a cross-layer approach. IEEE J. Selected Areas in Comm. Jan. 2005.
- 3 Park JS, Lun DS, Soldo F, Gerla M, Medard M. Performance of network coding in ad hoc networks. Proc. IEEE Milcom 2006. Oct. 2006.
- 4 Lun D, Ratnakar N, Koetter R, Medard M, Ahmed E, Lee H. Achieving minimum-cost multicast: a decentralized approach based on network coding. Proc. IEEE INFOCOM. Miami, Florida. Mar. 2005.
- 5 Fragouli C, Widmer J, LeBoudec JY. A network coding approach to energy efficient broadcasting: from theory to practice. Infocom 2006. March 2006.
- 6 Wu Y, Chou PA, Kung SY. Minimum-energy multicast in mobile ad hoc networks using network coding. IEEE Information Theory Workshop. San Antonio. Oct. 2004.
- 7 Deb S, Effros M, Ho T, Karger DR, Koetter R, Lun DS, Medard M, Ratnakar N. Network coding for wireless applications: A brief tutorial. IWWAN. 2005.
- 8 Lun DS, Medard M, Effros M. On coding for reliable communication over packet networks. Proc. 42nd Annual Allerton Conference on Communication, Control, and Computing. 2004.
- 9 Ghaderi M, Towsley D, Kurose J. Reliability gain of network coding in lossy wireless networks. Proc. IEEE INFOCOM, miniconference. Phoenix, USA. April 2008.
- 10 Katti S, Rahul H, Hu W, Katabi D, Medard M, Crowcroft J. XORs in the Air: Practical Wireless Network Coding. Proc. of ACM SIGCOMM'06. 2006.
- 11 Rouayheb SE, Chaudhry MAR, Sprintson A. On the minimum number of transmissions in single-hop wireless coding networks. IEEE Information Theory Workshop (Lake Tahoe). 2007.
- 12 Nguyen D, Nguyen T, Bose B. Wireless Broadcast Using Network Coding. IEEE NetCod Workshop. 2007.
- 13 Costa RA, Munaretto D, Widmer J, Barros J. Informed network coding for minimum decoding delay. IEEE MASS. Atlanta, Georgia, US. Sep. 2008.
- 14 Sundararajan J, Shah D, Medard M. ARQ for Network Coding. Proceedings of the IEEE International Symposium on Information Theory (ISIT 2008). Toronto, Canada. Jul. 2008.
- 15 Zhan C, Xu Y, Wang J, Lee V. Reliable multicast in wireless networks using network coding in Mobile Ad hoc and Sensor Systems. MASS'09. IEEE 6th International Conference on. Oct. 2009.
- 16 Dong Z, Zhan C, Xu Y. Delay aware broadcast scheduling in wireless networks using network coding. Networks Security Wireless Communications and Trusted Computing (NSWCTC). Second International Conference. 2010.