

多播网络中基于网络编码的高效丢失恢复机制

周志恒* 周亮

(电子科技大学通信抗干扰技术国家级重点实验室 成都 611731)

摘要: 网络编码为无线网络中可靠多播通信提供了有效解决途径。该文分析了网络中编码机会的变化规律, 研究了解码失败的编码数据包对网络编码性能的影响, 提出了新的基于网络编码的丢失恢复算法(NCLR)。NCLR要求节点缓存解码失败的编码数据包, 并反馈信息给发送节点。根据各个节点的丢包情况, NCLR通过优先传输对编码性能影响较大的数据包, 并在需要重传的已编码数据包和原始数据包中选择编码组合, 来充分挖掘网络中的编码机会。仿真结果表明相对于已有算法, NCLR算法可以在重传次数和丢失恢复时延方面有显著性能改善。

关键词: 多播网络; 网络编码; 丢失恢复; 优化策略; 数据调度

中图分类号: TN92

文献标识码: A

文章编号: 1009-5896(2012)08-1962-06

DOI: 10.3724/SP.J.1146.2011.01233

Efficient Loss Recovery Based on Network Coding in Multicast Networks

Zhou Zhi-heng Zhou Liang

(National Key Laboratory of Communications, University of Electronic Science and Technology of China, Chengdu 611731, China)

Abstract: Network coding is a promising technique for reliable multicast in wireless networks. In this paper, the characteristics of network coding opportunities are analyzed in multicast networks. The effect of the network coded packets, which can not be decoded, on the performance of network coding is studied. Based on these analyses, a Network Coding Loss Recovery (NCLR) scheme is proposed. NCLR allows receivers to store the packets, which they can not decode, and report the reception status to the sender. According to the feedback, NCLR prioritizes the packets that can contribute more to the overall transmission performance. Moreover in NCLR, not only original packets, but also encoded packets are scheduled and coded together to fully exploit coding opportunities. Simulation results show that NCLR improves significantly the multicast performance in terms of the number of retransmissions and recovery latency compared with existing schemes.

Key words: Multicast networks; Network coding; Loss recovery; Optimal scheme; Data scheduling

1 引言

多播是无线网络中最常用的数据传输方式之一。在许多无线网络应用中, 例如视频传输、文件下载和军事通信中, 都要求多播传输具有可靠性。然而, 由于无线链路间的干扰和多径衰落等因素, 导致无线通信具有传输带宽受限和高丢包率等特性, 从而影响了多播传输的可靠性。

针对上述问题, 文献[1]提出了将机会网络编码^[2](Opportunistic Network Coding, ONC)与自动重传请求^[3](Automatic Repeat Request, ARQ)相结合的重传技术——ONC-ARQ。但是文献[1]中并没

有研究网络编码优化问题。Rouayheb 等人^[4]证明了在单跳网络中使用 ONC 进行最小次数传输是 NP 问题。Rozner 等人^[5]建立了图模型, 并将最优网络编码重传策略转化为图中的最小团划分问题, 提出了 Efficient Retransmission(ER)算法。文献[6-8]指出合理安排丢包的重传顺序, 可以减少传输时延和重传次数。虽然这些算法^[3,5,7]的复杂度较低, 但是由于其要求生成的编码包必须能被所有节点成功解码, 因此会浪费大量编码机会, 导致其性能不佳。Zhan 等人^[9]证明了允许编码包在部分节点解码失败, 可以有效地减少重传次数。此外, 文献[9]提出的 CMRE 算法还允许节点缓存解码失败的编码包来方便随后的解码运算。虽然基于缓存的编码算法^[9,10]可以减少重传次数, 但是这些算法并没有考虑当编码数据包在目标节点丢失时, 应该如何通过有效地重传这些编码包来充分挖掘网络中的编码机

2011-11-25 收到, 2012-05-24 改回

国家科技重大专项(2010ZX03003-003)和中央高校基本科研业务费(E022050205)资助课题

*通信作者: 周志恒 miharry0606@gmail.com

会，提高网络编码的效率。

基于以上考虑，本文针对无线多播网络的特点和已有算法的优缺点，提出了新的基于机会网络编码的丢失恢复算法(NCLR)。NCLR 允许编码包在部分节点解码失败，同时还要求接收节点缓存解码失败的编码包，并反馈该信息给发送节点。根据节点的反馈信息，NCLR 算法在需要重传的已编码数据包和原始数据包中选择最优的编码组合，以保证每次重传能尽可能多地恢复丢失的数据。NCLR 分析了重传过程中编码机会变化的规律，设计了数据调度机制。通过优先传输对编码性能影响较大的数据包，NCLR 算法可以进一步减少重传次数和丢失恢复时延。

2 基于网络编码的丢失恢复(NCLR)

2.1 系统模型和定义

多播网络包括一个发送节点(BS)和 $N(N>2)$ 个接收节点 U_i , $1 \leq i \leq N$ 。整个通信过程分为两个阶段：原始数据传输阶段和丢失恢复阶段。在原始数据传输阶段，发送节点向接收节点广播 K 个原始数据包。在丢失恢复阶段，发送节点不断重传丢失的数据，直到所有数据都被正确接收。接收节点采用自动重传请求(ACKs)同步反馈其丢包信息，并假定反馈信道是可靠的^[3,7,10,11]。

定义 1 接收向量 $\mathbf{V}(P)$ 表示数据包 P 的接收情况。如果节点 U_i 接收到(或解码出) P ，则 $\mathbf{V}(P)$ 中的第 i 个元素赋值为 1；如果 U_i 丢失 P ，则该元素赋值为 0。

定义 2 $Z(P)$ 表示 $\mathbf{V}(P)$ 中零元素的个数。

图 1 给出了接收向量的示例。其中， $Z(P_1) = 1$, $Z(P_2) = Z(P_3) = 2$ 。

定义 3 $C(P)$ 是数据包 P 中包含的需要重传的原始数据包的集合。特别地，如果 P 为原始数据包，则 $C(P) = \{P\}$ 。

定义 4 如果节点 U_i 能通过接收(或解码)数据包 P 来恢复自己的丢包，则称 U_i 为 P 的目标节点。 $D(P)$ 是 P 目标节点的集合。

如图 1 所示， $C(P_1 \oplus P_2) = \{P_1, P_2\}$, $D(P_1 \oplus P_2) = \{U_1, U_2, U_4\}$ 和 $C(P_3) = \{P_3\}$, $D(P_3) = \{U_1, U_3\}$ 。虽

	U_1	U_2	U_3	U_4
$\mathbf{V}(P_1) =$	[0	1	1	1]
$\mathbf{V}(P_2) =$	[1	0	1	0]
$\mathbf{V}(P_3) =$	[0	1	0	1]

图 1 接收向量示例

然 U_3 也能解码 $P_1 \oplus P_2$ ，但是由于该编码包中没有 U_3 的丢包，所以 U_3 不是 $P_1 \oplus P_2$ 的目标节点，即 $U_3 \notin D(P_1 \oplus P_2)$ 。

2.2 NCLR 概述

NCLR 方法采用同步反馈方式，实时更新数据包丢失情况，以此来动态选择数据包的组合方式。由于在发送节点恢复数据的过程中，随着接收节点正确接收到自己的丢包，网络中的编码机会也在发生变化。因此，NCLR 方法充分考虑网络中编码机会的变化性，设计动态编码方法来提高网络编码效率。

NCLR 方法优先恢复较多节点需要的数据包。由于无线链路的不可靠性，一个数据包在被重传后，网络中的编码机会会出现下面 3 种变化情况：(1) 如果目标节点都没有接收到该数据包，则网络中的编码机会不变；(2) 如果目标节点都接收到，则编码机会减少；(3) 如果只有部分目标节点接收到，则编码机会可能增加。这是因为，一个数据包被越多节点接收到，那么能与之同时恢复的数据包个数就越多。在图 1 所示例子中，如果存在数据包 P_4 , P_5 和 P_6 ，且 $\mathbf{V}(P_4) = [1\ 0\ 1\ 1]$ ， $\mathbf{V}(P_5) = [1\ 1\ 0\ 1]$ 和 $\mathbf{V}(P_6) = [1\ 1\ 1\ 0]$ ，则发送节点可以通过传输编码包 $P_1 \oplus P_4 \oplus P_5 \oplus P_6$ 来同时恢复这 4 个数据。而在传输 P_2 (或 P_3) 时，却最多只能同时恢复 P_4 , P_5 和 P_6 中的任意 2 个。NCLR 充分利用上述编码机会变化的规律，设计了数据调度机制。通过优先传输被较多节点需要的数据包，以达到增加网络中的编码机会和减少重传次数的目的。

NCLR 方法允许编码包在部分节点解码失败。在 ONC 中，当编码包中包含了多个某接收节点丢失的原始数据包时，该节点将无法求解这个编码包。文献[4,5,7]为了能让所有接收节点成功解码接收到的每个编码数据包，采用了如下编码原则：对每个接收节点 U_i ($i=1, \dots, N$)，发送节点必须保证其生成的编码数据包 P_{en} 中最多只含有一个 U_i 丢失的数据包。虽然上述编码方案的复杂度较低，但是会浪费编码机会，降低网络编码算法效率。在图 1 中，如果数据包 P_1 , P_2 和 P_3 所对应的接收向量是 $\mathbf{V}(P_1) = [1\ 0\ 0\ 0]$ ， $\mathbf{V}(P_2) = [0\ 0\ 0\ 1]$ 和 $\mathbf{V}(P_3) = [1\ 0\ 0\ 1]$ ，则采用上述编码方案，发送节点 BS 只能分别传输 P_1 , P_2 和 P_3 。

为了能充分发挥网络编码的优势，NCLR 方法采用了新的编码原则，即允许编码包在部分节点解码失败。例如，将 P_1 和 P_2 编码在一起来同时恢复 U_1 和 U_4 中丢失的数据。值得注意的是，虽然 U_2 和 U_3 无法解码编码包 $P_1 \oplus P_2$ ，但是 U_2 和 U_3 能通过缓

存该编码包来协助以后的解码运算。假定当 BS 发送 $P_1 \oplus P_2$ 后, 只有 U_2 和 U_3 接收到 $P_1 \oplus P_2$ 。如果 U_2 和 U_3 缓存了 $P_1 \oplus P_2$, 并反馈 ACK 给 BS, 则 $P_1 \oplus P_2$ 对应的接收向量为 $\mathbf{V}(P_1 \oplus P_2) = [0 \ 1 \ 1 \ 0]$ 。在下次重传时, BS 就可以通过传输 $P_1 \oplus P_2 \oplus P_3$ 来同时恢复 U_1, U_2, U_3 和 U_4 的丢包。

总而言之, NCLR 方法通过(1)使用动态编码算法优先重传被较少节点接收到的数据包, (2)允许部分节点解码失败和(3)允许节点缓存解码失败的编码数据包并反馈接收信息, 就能够充分挖掘网络中的编码机会, 提高网络编码效率。

2.3 更新算法

要挖掘网络中的编码机会, 发送节点首先需要收集各个接收节点反馈的信息, 并以此更新数据包的接收情况, 最后判断哪些原始/编码数据包需要重传。本节给出相应的更新算法。

在 NCLR 方法中, 当数据包 P 被发送以后, 发送节点首先将 P 放入队列 TransQueue 中, 并将集合 $C(P) - \{P\}$ 中的数据放入队列 PendQueue 中。然后, 发送节点等待接收节点反馈信息。当接收到节点 $U_i (i=1, \dots, N)$ 对数据包 P 的确认信息后, 发送节点根据该信息更新集合 $C(P) \cup \{P\}$ 中的数据。本文提出的更新算法如表 1 所示。

算法设计思路如下: 当发送节点接收到 U_i 对 P 的确认信息后, 发送节点首先将 $\mathbf{V}(P)$ 中第 i 个元素

赋值为 1, 表示 U_i 正确接收到 P 。然后判断 U_i 是否为 P 的目标节点, 如果是, 则再判断 P 是否为编码包。如果 P 不是编码包, 则判断 P 是否已经被所有节点接收到, 如果是, 则丢弃 P (步骤 4-步骤 6)。如果 P 是编码包, 则判断集合 $C(P)$ 中的数据包 $P_j (j=1, \dots, |C(P)|)$ 是否被集合 $D(P_j) \cap D(P)$ 中的所有节点接收到。如果是, 则表明 P_j 不再需要被编码在 P 中, 于是执行步骤 11。接着再判断 P_j 是否已经被所有节点接收到, 如果是, 则丢弃 P_j , 否则, 将 P_j 放入队列 TransQueue 中, 以便安排下次重传 (步骤 12-步骤 16)。注意, 集合 $D(P_j) \cap D(P)$ 中的元素是能够通过接收 P 来恢复自己丢包 P_j 的接收节点。

2.4 NCLR 编码算法

发送节点通过两个方面来确定数据包的重要性: 首先, 一个数据包被越少的接收节点接收到, 则该数据对编码性能的影响就越大; 其次, 如果两个数据包的接收情况相同, 则被传输较多次的数据包对编码性能的影响就比另一个数据包所带来的影响要大。

NCLR 将重传队列划分为 N 个优先级队列 $Q(k), k=1, \dots, N$ (N 为网络中接收节点个数)。其中, 队列 $Q(k)$ 中存放着被 $N-k$ 个节点接收到的数据包。为了减少时延, 队列 $Q(k)$ 中的数据包还按照其被传输的次数降序排列。当接收到所有节点的反馈消息后, BS 将队列 TransQueue 中的所有数据包 $P_i (i=1, \dots, |\text{TransQueue}|)$ 插入到队列 $Q[Z(P_i)]$ 的相应位置中。

另一方面, 为了保证每次重传能尽可能多的恢复丢失地数据, NCLR 采用如下编码原则: 数据包 P_1, \dots, P_n 能被编码在一起, 仅当编码数据包 $P_{en} = P_1 \oplus \dots \oplus P_n$ 同时满足

$$n \leq \min_{P_i \in \{P_1, \dots, P_n\}} \{N - Z(P_i)\} \quad (1)$$

$$D(P_i) \cap D(P_{en}) \neq \phi, \quad \forall P_i \in C(P_{en}) \quad (2)$$

其中 $P_i (1 \leq i \leq n)$ 可以是原始数据包, 也可以是编码数据包。

结合上述优先级队列和编码原则可以给出 NCLR 编码算法。基于优先级队列的网络编码算法如表 2 所示。首先, 初始化各项参数, 其中设置数据包 P 为零向量, $\mathbf{V}(P)$ 为全 1 向量 (步骤 1)。然后, 从最高优先级队列开始, 判断该队列是否为空队列, 如果是, 则进入下一优先级队列。如果不是, 则从该队列中依次取出数据包 P_i , 判断 P_i 是否能与 P 编码在一起 (步骤 5)。如果是, 则执行步骤 6 和步骤 7, 同时在步骤 8 中判断 P_i 是否为编码包。如果 P_i 是编码包, 则丢弃它。最后, 当查找完所有优先

表 1 更新算法

```

//发送节点接收到节点  $U_i$  对数据包  $P$  的确认信息
(1)  $\mathbf{V}(P)$  中的第  $i$  个元素赋值为 1
(2) if  $U_i \in D(P)$ 
(3)   if  $C(P) = \{P\}$ 
(4)     if  $\mathbf{V}(P)$  为全 1 向量
(5)       丢弃数据包  $P$ 
(6)     end if
(7)   else
(8)     for  $j \leftarrow 1$  to  $|C(P)|$ 
(9)        $\mathbf{V}(P_j)$  中的第  $i$  个元素赋值为 1
(10)      if  $\mathbf{V}(P_j)$  中的第  $k$  个元素为 1,  $\forall U_k \in D(P_j) \cap D(P)$ 
(11)         $P \leftarrow P \oplus P_j$ 
(12)      if  $\mathbf{V}(P_j)$  是全 1 向量
(13)        丢弃数据包  $P_j$ 
(14)      else
(15)        将  $P_j$  放入队列 TransQueue 中
(16)      end if
(17)    end for
(18)  end if
(19) end if
(20) end if

```

表 2 编码算法

(1)	初始化: $P \leftarrow 0, \mathbf{V}(P) \leftarrow 1, D(P) \leftarrow \{U_1, \dots, U_N\}, C(P) \leftarrow \{\}$
(2)	for $k \leftarrow N$ to 1
(3)	if $ Q(k) \neq 0$
(4)	for $i \leftarrow 1$ to $ Q(k) $
(5)	if 编码包 $P \oplus P_i$ 满足条件式(1)和式(2)
(6)	$P \leftarrow P \oplus P_i$
(7)	将 $C(P_i)$ 中的数据包放入 PendQueue 中
(8)	if $C(P_i) \neq \{P_i\}$
(9)	丢弃数据包 P_i
(10)	end if
(11)	end if
(12)	end for
(13)	end if
(14)	end for
(15)	将 P 放入 TransQueue 中
(16)	return P

级队列后, 发送节点将 P 放入 TransQueue 中。需要注意的是, 如果 P 是编码包, 则为了使目标节点能够解码, 发送节点还需要在 P 中加入所有被编码在 P 中的数据包的 ID。

当节点 U_i 接收到数据包 P 后, 首先判断 $C(P)$ 中是否有自己的丢包, 如果没有, 则丢弃 P ; 否则, U_i 尝试解码 P 。在解码过程中, U_i 首先在缓存中寻找数据包 P_j , 其满足条件 $C(P_j) \cap C(P) \neq \phi$; 然后, 节点将所有满足该条件的数据包 P_j 以及 P 组成一个方程组, 最后, 通过高斯消元法来尝试求解这个方程组。如果 U_i 能通过该方程组解码出自己的丢包, 则 U_i 缓存该丢包。如果解码出的数据包包含在集合 $C(P)$ 中, 则 U_i 丢弃 P , 否则 U_i 缓存 P 。

3 性能分析

3.1 每个数据包的平均重传次数

假定 N 个接收节点的丢包率互不相关且服从伯努利分布^[3,11,12]。令 γ_i 表示接收节点 U_i ($1 \leq i \leq N$) 的丢包率, $\gamma_m = \max\{\gamma_1, \dots, \gamma_N\}$, 其中, m 是 $1, 2, \dots, N$ 中的一个数。

定义 5 每个数据包的平均重传次数 η : 在丢失恢复阶段, 一个数据包被重传的平均次数。

定理 1 在 N 个接收节点的多播网络中, 当数据包数 K 足够大时, NCLR 方法每个数据包的平均重传次数 η 为

$$\eta = \frac{\gamma_m}{1 - \gamma_m} \quad (3)$$

证明 令 X_i 表示接收节点 U_i 的丢包数, 则当发送节点传输 K 个数据包时, 各接收节点的平均丢包数为

$$E[X_i] = K\gamma_i, \quad 1 \leq i \leq N \quad (4)$$

因为 $\gamma_m \geq \gamma_i$, 所以 $E[X_m] \geq E[X_i]$ ($i = 1, \dots, N$), 即当 K 足够大时, 节点 U_m 为丢包最多的节点。则在 NCLR 方法中, 每个重传包都可以看成是 U_m 的丢包与其余 $N-1$ 个节点丢包的编码组合。因此在丢失恢复阶段, 重传次数由丢包率最大节点 U_m 决定。由于节点从一个重传包中最多只能恢复一个丢包, 所以 U_m 需要接收 $K\gamma_m$ 个重传包才能恢复出所有丢包。令 Y_m 表示 U_m 成功接收到一个数据包前, 发送节点需要传输的次数。则当 K 足够大时, NCLR 方法 K 个数据包的平均重传次数 R 为

$$R = E[X_m]E[Y_m] = \frac{K\gamma_m}{1 - \gamma_m} \quad (5)$$

每个数据包的平均重传次数 $\eta = R/K = \gamma_m/(1 - \gamma_m)$ 。证毕

3.2 丢失恢复时延

定义 6 数据平均恢复时延 D_p : 在丢失恢复阶段, 恢复一个数据包的平均传输延迟。

定理 2 在 N 个接收节点的多播网络中, 当 K 足够大时, NCLR 方法的数据平均恢复时延 D_p 为

$$\frac{(1+K)\gamma_m\Delta t}{2(1-\gamma_m)} \leq D_p \leq \frac{K\gamma_m\Delta t}{1-\gamma_m} \quad (6)$$

其中 Δt 表示发送节点从传输数据包到获得所有节点反馈信息的平均时隙长度。

证明 由定理 1 可知, 所有接收节点成功获得 P_j 所需传输 P_j 的次数是由 U_m 决定的。假定数据包在 U_m 的接收顺序为: P_1, \dots, P_K 。假定发送节点重传 R_n ($1 \leq n \leq K$) 次后, 所有节点成功接收到原始数据包 P_n 。因为发送节点平均需重传一个数据包 η 次才能使 U_m 获得该数据, 所以有

$$n\eta \leq E[R_{n-1} + \Delta R_n] = E[R_n] \leq K\eta, \quad 1 \leq n \leq K, R_0 = 0 \quad (7)$$

其中 ΔR_n 表示在 U_m 接收到 P_{n-1} 后, 发送节点为了恢复 P_n 还需要传输的次数。当 $E[\Delta R_n] = \eta$, $n = 1, \dots, K$ (发送节点连续恢复同一数据包) 时, 式(7)左边等号成立; 当 $E[\Delta R_1] = K\eta$, $\Delta R_n = 0$, $n = 2, \dots, K$ (使用随机线性网络编码^[13]) 时, 式(7)右边等号成立。对 $E[R_n]$ ($1 \leq n \leq K$) 取平均, 就得到了数据平均恢复时延 D_p 。

$$\frac{(1+K)\eta\Delta t}{2} \leq D_p = \frac{\Delta t}{K} \sum_{n=1}^K E[R_n] \leq K\eta\Delta t \quad (8)$$

最后, 将式(3)代入式(8)中得出式(6)。证毕

3.3 算法复杂度

NCLR 方法主要由调度、更新和编码 3 部分组成。在调度过程中, 根据数据包接收情况查找相应优先级队列的复杂度是 $O(N)$, 而根据被传输次数将数据包插入队列相应位置的复杂度是 $O(K)$, 从而总的复杂度是 $O(KN)$; 在更新算法中, 由于一个

编码包中最多包含 N 个丢包, 所以该算法的时间复杂度是 $O(N)$; 在编码算法中, 判断一个数据包是否满足编码条件的平均复杂度是 $O(N)$, 因此从重传队列中选择不同丢包生成重传包的复杂度是 $O(KN)$ 。

4 仿真结果和分析

本文比较了下述基于 ONC 的重传算法在数据平均重传次数和平均恢复时延方面的性能: (1) 文献 [3] 中提出的重传算法 ONC-ARQ; (2) 一种采用数据调度的重传算法 HLAR^[7]; (3) 一种基于缓存的重传算法 CMRE^[9]; (4) 本文提出的算法 NCLR。

本文使用 NS2 作为组播网络的仿真平台来进行评估。上述 4 个算法的性能都是在进行多次仿真实验后, 通过对实验数据进行统计平均所得到的结果。仿真中, 所有接收节点分布在半径为 200 m 的圆上。仿真中设置所有接收节点丢包率相同, $\gamma_1 = \dots = \gamma_N = \gamma$ 。在原始数据包传输阶段, 发送节点产生 $K = 50$ 个长度为 512 Byte 的数据包, 并广播给接收节点。在丢失恢复阶段, 发送节点分别使用上述 4 种算法恢复丢失的数据。仿真中, 每次传输占用一个时隙。在一个时隙内, 发送节点完成一次数据传输, 同时收集所有节点反馈的信息。

4.1 每个数据包的平均重传次数

图 2 给出了每个数据包的平均重传次数随丢包率增长变化的曲线图。图 2 中, 接收节点数 N 分别设为 5 和 10, 丢包率 $\gamma_1 = \gamma_2 = \dots = \gamma_N = \gamma$ 取值范围为 [0.03, 0.3], 步长为 0.03。图 3 给出了每个数据包的平均重传次数随接收节点数 N 增长变化的曲线图。图 3 中, γ 分别设为 0.2 和 0.3, N 取值范围为 [3, 15], 步长为 1。

由于 ONC-ARQ 和 HLAR 要求编码包能被所有节点成功解码, 因此会浪费大量编码机会, 导致这两种算法的性能下降最快。CMRE 算法的重传效率要高于 ONC-ARQ 和 HLAR, 但是由于 CMRE 不收集编码包的接收情况, 因而无法动态地选择编码组合。NCLR 算法优先恢复接收情况较差的数据包, 同时允许节点解码失败并收集数据包的接收情

况, 因此相比其它 3 种算法, NCLR 可以寻找到更多的编码机会, 因此在不同的丢包率和接收节点数情况下能够始终保持较高的性能。

4.2 丢失恢复时延

图 4 给出了数据平均恢复时延随丢包率增长变化的曲线图。图 4 中, 接收节点数 N 分别设为 5 和 10, 丢包率 γ 取值范围为 [0.03, 0.3], 步长为 0.03。图 5 给出了数据平均恢复时延随接收节点数 N 增长变化的曲线图。图 5 中, γ 分别设为 0.2 和 0.3, N 取值范围为 [3, 15], 步长为 1。

在 γ 较小时, ONC-ARQ 和 HLAR 算法可以保证被重传的丢包能被所有节点成功接收或解码出来。所以此时, ONC-ARQ 和 HLAR 算法的平均恢复时延最小。但是随着 γ 或 N 的增加, ONC-ARQ 和 HLAR 无法保证重传包能被所有节点正确得到, 同时其对编码机会的利用率也没有 CMRE 和 NCLR 算法高。因此在 γ 或 N 较大时, ONC-ARQ 和 HLAR 算法的平均恢复时延最大。另一方面, 因为 NCLR 要求每次重传能尽可能多的恢复数据, 同时优先重传对时延影响较大的数据包, 因此 NCLR 能使节点更快的恢复出丢包。所以 NCLR 算法的平均恢复时延要小于 CMRE 算法。

5 结束语

本文提出了一种基于机会网络编码的丢失恢复算法 NCLR。为了在每次重传中尽可能多的恢复丢包, NCLR 不仅允许生成的编码包在部分节点解码失败, 还要求接收节点在解码失败时缓存该编码包并反馈信息给发送节点。根据这些信息, 发送节点就能通过重传已编码数据包来提高编码效率。此外, 根据网络中编码机会变化的规律, NCLR 还设计了数据调度机制, 通过优先恢复较多节点需要的数据包, 来达到充分挖掘网络编码性能的目的。通过仿真各种多播网络环境, 对比其它 3 种基于机会网络编码的重传算法性能, 发现 NCLR 能有效地提高网络编码效率, 减少丢包重传次数和丢失恢复时延。在未来的工作中, 计划将信道编码技术引入到 NCLR 方法中, 进一步提高算法效率及可靠性。

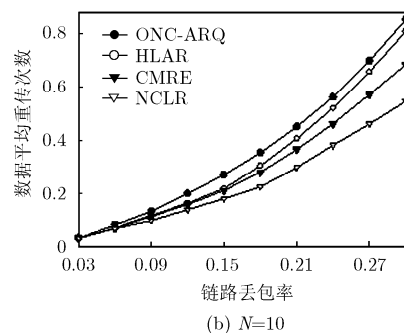
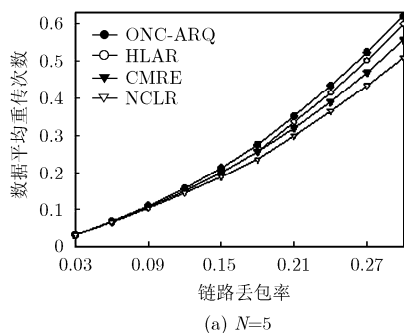


图 2 链路丢包率影响下的每个数据包的平均重传次数

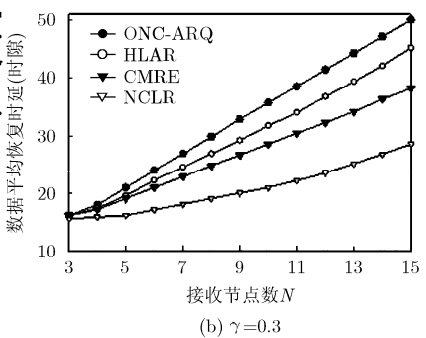
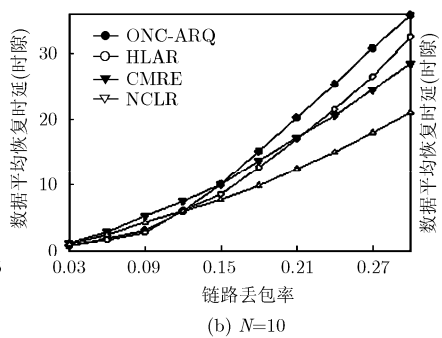
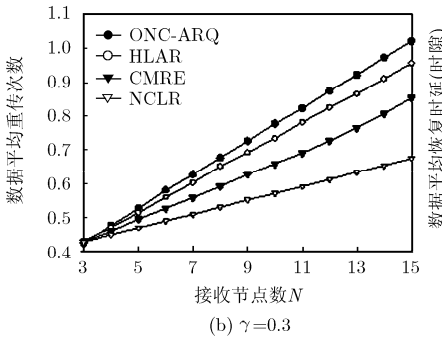
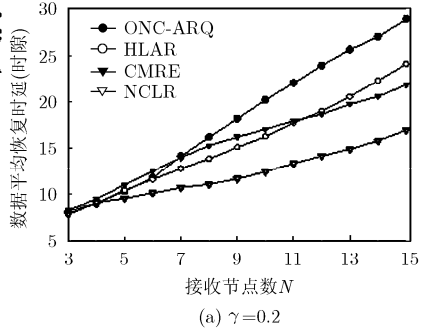
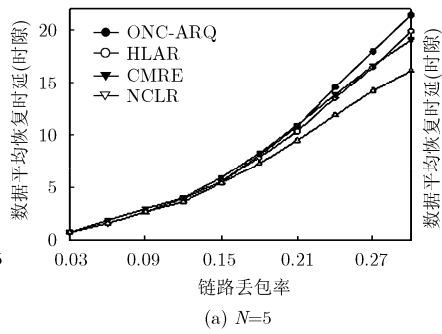
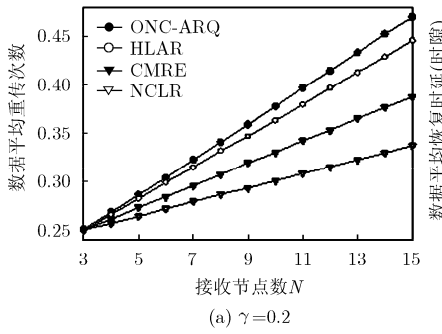


图 3 接收节点数影响下的每个数据包的平均重传次数

图 4 链路丢包率影响下的数据平均恢复时延

图 5 接收节点数影响下的数据平均恢复时延

参 考 文 献

[1] Nguyen D, Nguyen T, and Bose B. Wireless broadcasting using network coding[C]. Proceeding of Third Workshop on Network Coding, Theory, and Applications (NetCod 07), San Diego, CA, USA, Jan. 2007: 1-5.

[2] Katti S, Rahul H S, Hu Wen-jun, et al. XORs in the air: practical wireless network coding[C]. Proceeding of ACM Special Interest Group on Data Communication, Pisa, Italy, Aug. 2006: 243-254.

[3] Najjoh M, Sampei S, Morinaga N, et al. ARQ schemes with adaptive modulation/TDMA/TDD systems for wireless multi-media communication services[C]. Proceeding of IEEE International Symposium on Personal, Indoor and Mobile Radio Communications 97, Helsinki, Finland, Sept. 1997: 709-713.

[4] Rouayheb S, Chaudhry M, and Sprintson A. On the minimum number of transmissions in Single-Hop wireless coding networks[C]. Proceeding of IEEE Information Theory Workshop 07, Tahoe City, CA, USA, Sept. 2007: 120-125.

[5] Rozner E, Iyer A, Mehta P, et al. ER: efficient retransmission scheme for wireless lans[C]. Proceeding of The 7th International Conference on Emerging Networking Experiments and Technologies, Dec. 2007: 1-12.

[6] Zhang X, Ding Z, Peng M, et al. Performance of network-coding-assisted scheduling schemes and their applications in uplink time division duplexing code division multiple access systems[J]. *IET Communications*, 2011, 5(17): 2558-2568.

[7] 卢冀, 肖嵩, 吴成柯. 一种基于机会式网络编码的高效广播重传方法[J]. *电子与信息学报*, 2011, 33(4): 858-863.

Lu Ji, Xiao Song, and Wu Cheng-ke. Opportunistic network coding based delay-sensitive broadcast transmission algorithm[J]. *Journal of Electronics & Information Technology*, 2011, 33(4): 858-863.

[8] 王斌, 吴伟仁, 岩延, 等. 效率感知的无线网络编码机制[J]. *电子与信息学报*, 2012, 34(1): 69-74.

Wang Bin, Wu Wei-ren, Yan Yan, et al. Efficiency-aware localized network coding mechanism for wireless networks[J]. *Journal of Electronics & Information Technology*, 2012, 34(1): 69-74.

[9] Zhan Cheng, Xu Yin-long, Wang Jian-ping, et al. Reliable multicast in wireless networks using network coding[C]. Proceeding of IEEE International Conference on Mobile Ad-Hoc and Sensor Systems, China, Oct. 2009: 506-515.

[10] Lopes P, Silva Y, and Cavalcanti F. Efficient wireless multicast retransmission techniques based on multiple coded packets[C]. Proceeding of 2010 IEEE 72nd Vehicular Technology Conference Fall (VTC 2010-Fall), Ottawa, ON, Sept. 2010: 1-5.

[11] Sorour S and Valaee S. An adaptive network coded retransmission scheme for single-hop wireless multicast broadcast services[J]. *IEEE/ACM Transactions on Networking*, 2010, 19(3): 869-878.

[12] Zhang Zhang, Lv Tie-jun, Su Xin, et al. Dual XOR in the air: a network coding based retransmission scheme for wireless broadcasting[C]. Proceeding of 2011 IEEE International Conference on Communications, Kyoto, Japan, June 2011: 1-6.

[13] Li S Y Robert, Sun Q T, and Shao Zi-yu. Linear network coding: theory and algorithms[J]. *Proceedings of the IEEE*, 2011, 99(3): 372-387.

周志恒: 男, 1984年生, 博士, 研究方向为网络编码、无线通信与网络技术。

周亮: 男, 1961年生, 博士, 教授, 博士生导师, 研究方向为密码学、信道编码与无线通信技术。