

# 有环网络CNC译码算法优化设计

杨云飞, 朱清超\*

武警工程大学信息工程学院, 陕西 西安

收稿日期: 2021年9月10日; 录用日期: 2021年10月6日; 发布日期: 2021年10月13日

## 摘要

卷积网络编码(CNC)译码算法汇点容量区域及译码算法已知的假设条件并不能同时成立。为此在分析有环网络CNC译码机制的基础上, 指出解决CNC译码问题的关键在于初始化算法的设计和容量区域确值问题。通过简化系统状态方程, 建立发送和接收序列关系表达式, 区分节点是否具有重置/清零操作, 差异化提出与之对应的译码算法, 并推导传输矩阵满秩条件时容量区域定量表达式, 并对比分析了时延、吞吐量、丢包率、控制开销等性能指标。结果表明, 新算法吞吐量提高了3.2 Mbps, 时延、丢包率、控制开销分别降低了37%、25%、0.9 M, 验证了新算法的可行性和有效性, 为进一步提高网络安全性能提供了一定理论基础。

## 关键词

卷积网络编码, 容量区域, 有环网络, 吞吐量, 时延

# Design of Decoding Algorithm Optimization for Cyclic Network

Yunfei Yang, Qingchao Zhu\*

College of Information Engineering, Engineering University of PAP, Xi'an Shaanxi

Received: Sep. 10<sup>th</sup>, 2021; accepted: Oct. 6<sup>th</sup>, 2021; published: Oct. 13<sup>th</sup>, 2021

## Abstract

Decoding procedure of convolutional network coding (CNC) relies on multiple dimension prior information of the given network. Moreover, two assumptions, that are, known capacity region and decoding algorithm, couldn't be established simultaneously. Therefore, after analyzing the current decoding mechanism of CNC in cyclic network, algorithm optimization should be focused

\*通讯作者。

on two key factors, called as initialization algorithm design and capacity region confirmation. Thus a simplified state space representation of the system was founded, which established an input-output relationship. And two differential decoding algorithms were proposed after distinguishing whether nodes have resetting/clearing operation or not. Then a quantified expression of capacity region was derived based on full rank matrix of transfer function. At last, network performance was analyzed by delay, throughput, packet loss rate and control overhead. Results show that, for the novel algorithm, throughput raises about 3.2 Mbps, delay reduces by 37%, packet loss rate decreases by 25%, and control overhead diminishes about 0.9 M, all of which prove that the proposed algorithm is feasible and efficient, providing a theoretical basis for better improving the security level of the network.

## Keywords

Convolutional Network Coding (CNC), Capacity Region, Cyclic Network, Throughput, Delay

Copyright © 2021 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

## 1. 引言

网络编码(network coding, NC)通过中间节点编码, 汇点基于编码理论译码提高网络容量, 成为“5G+”或“6G”移动网络和类 ad hoc 带限网络(移动自组网、无线传感器网络)的关键技术[1] [2] [3]。但是任意中间节点受到攻击后, 恶意信息会随着中间节点编码进入信息分组并转发, 引发网络安全问题。文献[4]指出, 卷积网络编码(Convolutional NC, CNC)为该问题的解决提供了思路和安全保障, 成为 NC 研究的重点。

CNC 早期为规避复杂信息流问题, 假设传输无时延, 保证时延与分组信息的独立性, 针对无环网络相关编码算法展开研究, 使得无环网络编码在单会话和多会话模式中与传统路由协议“存储-转发”机制所获得的网络容量一致, 其值与网络节点数目正相关[5] [6] [7]。文献[4] [8] [9]指出有环网络为兼顾网络因果性, 必须考量时延等参数及独立会话等新特性, “多会话-单播”模式网络编码能够以大概率获得更高的网络容量, 使其成为 CNC 研究的重点。文献[10]指出对于单播、多播等无环网络, 网络最大吞吐量可达源点和汇点之间的最小割, 传统路由机制无法达到该极限值。在此基础上, 文献[11]针对有环网络提出一种 CNC 机制, 并证明了最小割边界对应最优码的存在性。但上述研究存在的共性问题节点必须获得网络拓扑先验信息, 这对于动态变化的网络而言, 感知开销和处理时延将明显增加。文献[12]指出, 有环网络中对于包含源点的所有子网  $A$ , 源点速率之和  $r$  不大于汇点和  $A$  之间的最小割能力且内核资源池足够大时, 根据大数定理, 所有汇点从中随机选择编码内核可大概率恢复出传输的符号信息, 且该随机线性网络编码算法可用于拓扑动态变化的无中心授权(用于分配编码内核)网络。文献[13] [14]以此为基础, 设计时变译码算法, 使得任意节点可以利用网络冲激响应的前  $n+L$  段用时  $n$  恢复传输的符号,  $L$  为译码延时, 但其全冲激响应过于依赖全局编码内核, 这使得其实现过程必须建立在源点容量区域已知和汇点译码算法已知的两个前提条件上。二者任一个条件不成立时, 需要设计优化机制, 以确定系统传输函数或特征多项式等参数, 便于汇点处的译码操作。文献[15]分析现有译码方法的基础上, 指出有环网络中的 CNC 译码算法有待进一步完善。

论文在此基础上, 规避以上两个条件的判断环节, 提出差异化 CNC 译码算法以及容量区域参数判定

方法。译码算法由汇点执行初始化算法预测传输矩阵, 求解差分方程实现, 根据节点初始状态是否具备置零功能, 差异化设计与之对应的译码机制; 容量区域判定算法在假设传输矩阵为满秩矩阵的基础上, 分析允许节点执行的最大传输速率。最后仿真分析了时延、吞吐量、丢包率、控制开销等性能指标, 证明了设计的 CNC 译码机制的可行性和有效性。该 CNC 译码机制的实现为缓解 NC 网络安全问题提供一定的理论基础。

## 2. 问题描述

控制理论指出, 利用冲激响应(Impulse Response, IR)可识别未知线性时不变(Linear time invariant, LTI)系统, 其方法是通过获取参数矩阵  $M$ 、 $N$ 、 $P$ 、 $Q$ , 建立系统状态空间表达式, 使系统输入输出关系满足状态方程

$$\begin{cases} s[t+1] = Ms[t] + Ng[t] \\ o[t+1] = Ps[t] + Qg[t] \end{cases} \quad (1)$$

式中  $g$ 、 $o$ 、 $s$  分别为输入向量、输出向量和状态向量, 且各值为统计方法获得的近似值。状态空间表达式确定后, 通过  $z$  变换可得系统传输函数

$$H(z) = P \cdot adj(zI - M) \cdot N / C_M(z) + Q \quad (2)$$

式中  $C_M(z) = \det(zI - M)$ 、 $H(z)$ 、 $adj(\bullet)$  分别为矩阵  $M$  的特征多项式、系统传输函数的  $z$  变换、邻接矩阵。文献[16] [17]指出, 式(2)对应的状态空间表达式也可用于表征包含 CNC 对应的有环网络。同时只要确定了  $H(z)$  和  $C_M(z)$ , 输入输出关系便唯一确定。文献[18]提出一种利用冲激响应  $R(t)$  构建 Hankel 矩阵, 获得已知 LTI 系统输入输出关系, 并通过奇异值分解(Singular value decomposition, SVD)构建系统传输函数  $H(z)$  的方法, 但方法为整个实数或复数域, 其作用域需作进一步限定。

在此基础上, 当  $C_M(z)$  已知时, LTI 系统传输函数便唯一确定。系统输出信号通过包含  $C_M(z)$  的有限冲激响应(Finite impulse response, FIR)滤波器后, 系统传输函数进一步写为

$$H(z) \cdot C_M(z) = P \cdot adj(zI - M) \cdot N \quad (3)$$

对于式(3), 文献[19]提出通过 Hankel 矩阵次对角线获得特征多项式  $C_M(z)$  的方法, 但该方法依赖系统阶数、状态向量最小实现维度等先验信息, 实现复杂度较高。若仅建立边数、最大可达速率等基本参数的关联, 将很大程度上可降低算法实现复杂度。同时式(2)、式(3)为优化算法的设计提供了基本思路。

令通信网络由有向图  $G = (N, E)$  表示,  $N$  为节点集合,  $E$  为边/链路集合, 且每条链路均用无噪声有向链路表示, 每个时间单元均可传输一个符号, 各符号从标量域  $A$  中选择。假设每条链路任意两个符号传输之间为一个单位时延, 且所有链路均已同步。

令  $S$  为源点集合,  $R$  为汇点集合。任意源点  $s \in S$  单位时间内符号发送速率为  $V_s$ , 汇点  $r \in R$  期望接收所有源点发送的符号。对于每条边  $e \in E$ ,  $u \in Up(e)$  和  $d \in Down(e)$  分别为  $e$  的两个端点, 且符号从  $u$  向  $d$  传输, 于是任意节点发送和接收符号对应的边集合可表示为  $In(n) = \{e \in E : n = Up(n)\}$  和  $Out(n) = \{e \in E : n = Down(n)\}$ 。任意时刻  $t$  链路  $e$  传输的符号记为  $s_e(t)$ 。基于上述符号定义, 链路  $e \in Out(j)$  上发送的符号为节点  $j$  上一时隙接收的符号与产生符号的线性组合, 式(1)可写为

$$s_e(t+1) = \sum_{e \in In(j)} a_e s_e(t) + \sum_{k=1}^{V_s} b_{e,k} g_{j,k}(t) \quad (4)$$

其中  $g_{j,k}(t)$  为时刻  $t$  节点  $j$  产生的第  $k$  个符号,  $\{a_e, b_{e,k}\}$  为节点  $j$  提前选定的本地编码内核。

定义序列  $\{m[t]\}_{t \in \mathbb{N}}$  时移操作符为

$$(f^i m)[t] = m[t+i]$$

那么

$$(C(f)m)[t] = \sum_{i=0}^V a_e m(t+i) \quad (5)$$

令状态列向量  $s(t)$  长度为  $|I|$ , 传输的各符号表示为  $\{s_e(t)\}_{e \in I}$ 。输入序列  $\mathbf{g}(t) = (\mathbf{g}_{s,1}^T[t], \dots, \mathbf{g}_{s,S}^T[t])$ , 其中  $\mathbf{g}_{s,i}[t] = (g_{s,i,1}[t], \dots, g_{s,i,V_s}[t])$  为源点  $s_i$  对应输入序列,  $\mathbf{g}(t)$  列向量维度为  $\dim[\mathbf{g}(t)] = \sum_{s \in S} V_s$ 。在系统传输函数已知情况下,  $t < 0$ , 令  $\mathbf{g}(t) = \mathbf{0}$ , 任意节点状态向量  $s(t)$  零状态值大概率为零, 即  $t < 0$  时,  $\mathbf{o}(t) = \mathbf{0}$ 。当系统无法获得网络拓扑、本地编码内核等先验信息时, 算法仍然需要源点和汇点在初始化开始之前获得部分网络参数, 如源点集合和传输速率/速率上边界、网络边的数目或上边界等先验参数, 那么此时  $\{\mathbf{g}(t)\}_{t=0}^{s_0}$ , 经过时延  $\tau$  后, 接收符号  $\{\mathbf{o}(t)\}_{t=0}^{o_0+\tau}$ 。上述两类参数需确定容量区域内的节点传输速率  $\{V_s\}_{s \in S}$ 。

于是有环网络 CNC 译码机制需解决三个问题: 1) 零输入状态时的 CNC 初始化, 相应节点具有重置/清零功能; 2) 非零初始状态时的 CNC 初始化, 相应节点无重置/清零功能; 3) 各源点的容量区域确值。

### 3. 方案设计

本节针对以上问题, 区分网络节点有、无重置/清零操作, 设计算法 1 (有重置操作时初始化) 和算法 2 (无重置操作时初始化) 实现域内节点参数及矩阵的初始化, 本质是在确定接收矩阵和发送矩阵后, 将其与接收向量代入输入输出关系方程, 从而获得发送向量。二者区别在于, 算法 1 通过重置操作后, 无需计算常数向量, 其执行效率优于算法 2。而后设计算法 3 通过汇点检查传输矩阵解决源点容量区域确值问题, 用于确定源点所允许的最大可达速率区间, 使汇点能够成功接收分组信息, 防止数据溢出, 而传输矩阵依赖于初始化算法, 因此算法 3 必须在算法 1 或算法 2 之后执行。于是优化机制设计如图 1 所示。算法首先判断区域内节点是否具有清零/重置功能, 如果有执行算法 1 进行初始化, 否则执行算法 2 进行初始化。当初始化完成后, 执行算法 3 完成容量区域确值。最后网络运行相关协议及数据分发, 直至完成数据分组传输。初始化算法及容量区域判定算法的执行均为汇点译码提供一定的基础数据支撑。

如前所述, 算法目的是对汇点  $r$  建立输出序列  $\mathbf{g}$  和输出序列  $\mathbf{o}$  之间的关系, 以此译码获得发送符号, 即算法的本质是使汇点建立如下差分方程

$$C_r(f)\mathbf{o} = G_r(f)\mathbf{g} \quad (6)$$

那么问题的关键在于从接收序列  $\mathbf{o}$  中确定满足式 6 的多项式  $C_r(f)$  和矩阵  $G_r(f)$ 。下面分别分析算法 1~3 的求解思路及流程。

#### 3.1. 有重置操作时初始化

令  $T$  为网络中的边数。为确保输入输出关系对所有节点适用,  $T$  足够大时, 在此增加 2 个虚拟节点和  $V_e$  个虚拟边, 不影响网络整体性能, 且虚拟节点和虚拟边不与原始网络物理互连, 不影响实际网络性能分析, 此时新网络对应的边数仍用  $T$  表征。根据  $s$ 、 $\mathbf{g}$  和  $\mathbf{o}$  的定义, 对于任意汇点  $r$ , 网络状态空间表达式可用式(1)表示。

矩阵  $\mathbf{M}$  和  $\mathbf{N}$  维度分别为  $T \times T$  和  $T \times \dim[\mathbf{g}(t)]$ , 二者均由网络结构和本地编码内核决定; 矩阵  $\mathbf{C}$  和  $\mathbf{D}$  为二值矩阵(仅包括 0 和 1)。对于任意汇点  $r$ , 式(1)对应的一般解可写为

$$o_r(t) = P_r M^t s_0 + \sum_{i=0}^{t-1} P_r M^{t-1-i} N g[i] + Q_r g[t] \quad (7)$$

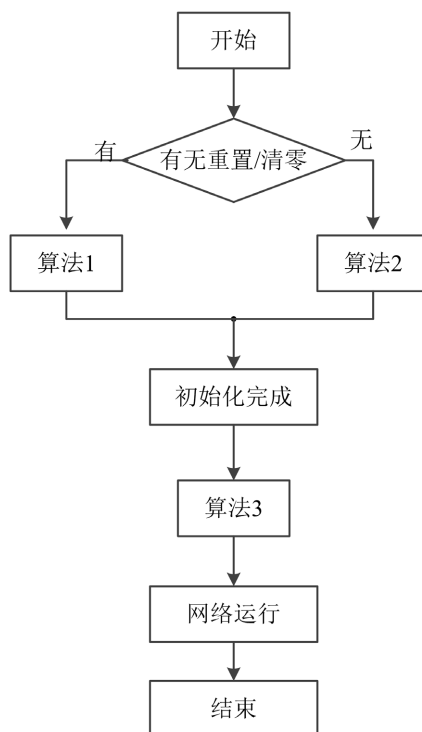


Figure 1. Flow chart of the optimization algorithm  
图 1. 优化算法流程图

当  $s_0 = 0$  时, 令输入序列  $g[t]$  为单位向量

$$g_i[t] = \begin{cases} e_i, t = 0 \\ 0, 1 \leq t \leq 2T \end{cases}$$

其中  $e_i$  为基向量, 即

$$e_i = (0, 0, \dots, p_k \dots, 0), p_k = \begin{cases} 1, k = i \\ 0, k \neq i \end{cases}$$

那么输出序列可表示为

$$o_{r,i}(t) = \begin{cases} Q_r e_i, t = 0 \\ P_r M^{t-1} N e_i, 1 \leq t \leq 2T \end{cases}$$

将输出序列组合为矩阵形式, 可表示为

$$\tilde{A}_d[t] = \begin{cases} P_r M^{t-1} N, 1 \leq t \leq 2T \\ Q_r, t = 0 \end{cases} \quad (8)$$

由于  $C_M(t)$  为矩阵  $M$  的特征多项式, 令  $\{p_k\}_{k=0}^{t-1}$  为多项式系数, 则

$$C_M(t) = t^T + \sum_{k=0}^{t-1} p_k t^k \quad (9)$$

那么下式成立

$$C_M(\mathbf{M}) = \mathbf{M}^T + \sum_{k=0}^{T-1} p_k \mathbf{M}^k = \mathbf{0}$$

那么此时需要设计满足下式的非零多项式  $C_r(\mathbf{M})$

$$\mathbf{P}_r C_r(\mathbf{M}) \mathbf{M}^\alpha \mathbf{N} = \mathbf{0}, \forall \alpha \in \mathbb{N} \tag{10}$$

线性方程(10)包含无穷多个方程式, 由于方程式之间线性相关, 那么只需确定部分方程组即可。对于  $\forall \alpha \in \{1, 2, \dots, T\}$ ,

$$\mathbf{P}_r C_r(\mathbf{M}) \mathbf{M}^{\alpha-1} \mathbf{N} = \sum_{k=0}^T a_e \mathbf{P}_r \mathbf{M}^{k+\alpha-1} \mathbf{N} = \sum_{k=0}^T a_e \tilde{\mathbf{A}}_r(k+\alpha) = \mathbf{0} \tag{11}$$

成立, 当确定  $T$  个方程组后即可从输出序列中恢复输入序列。由式(6)和式(11)可知, 若  $\text{rank}[G_r(f)] = \text{full}$  成立, 即  $G_r(f)$  为满秩矩阵时

$$C_M(f) = \sum_{k=0}^T a_e f^k \tag{12}$$

$$G_r(f) = \sum_{k=1}^T \sum_{i=k}^T a_e \tilde{\mathbf{A}}_r(i-k+1) f^{k-1} + \tilde{\mathbf{A}}_r(0) C_M(f) \tag{13}$$

与之对应的算法流程如图 2 所示。

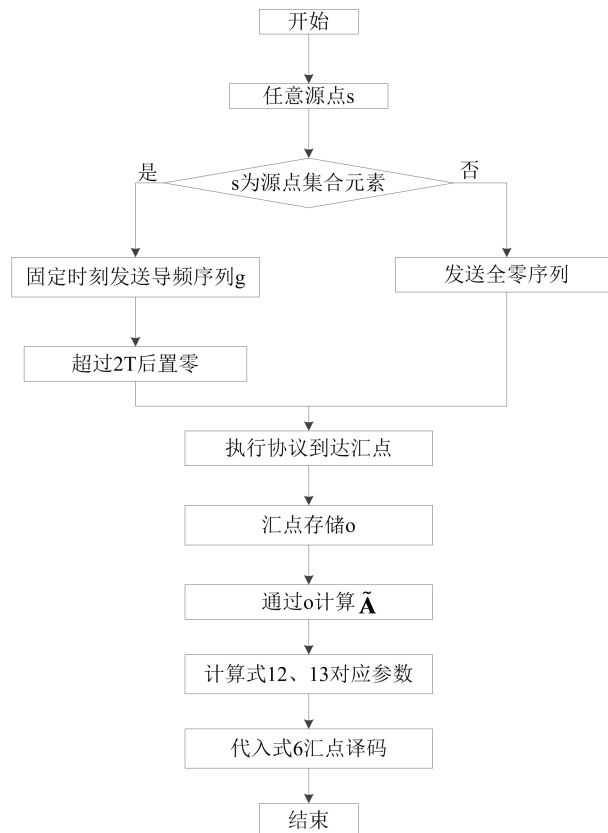


Figure 2. Flow chart of algorithm 1  
图 2. 算法 1 流程图

### 3.2. 无重置操作时初始化

无重置操作时 CNC 初始化思想与 2.1 节类似, 以获得  $C_M(f)$  和  $G_r(f)$  为目的, 进而建立  $s$  和  $o$  之间的关联, 使式 6 恒成立。仍假设  $G_r(f)$  为满秩矩阵, 那么  $C_M(f)$  和  $G_r(f)$  表达式为

$$C_M(f) = \sum_{k=0}^T a_e f^k \quad (14)$$

$$G_r(f) = \{g_{r,i}(f) : i = 1, 2, \dots, \dim[g(n)]\} \quad (15)$$

式中

$$g_{r,i}(f) = \sum_{k=1}^T \sum_{i=k}^T a_e o_r(i + (2T+1)j - k + 1) f^{k-1}, \forall j \in \{1, 2, \dots, \dim[g(n)]\}$$

具体推导过程在此不再赘述, 对应的算法流程图如图 3 所示, 其中  $V_s$  由  $G_r(f)$  非相关向量数确定。

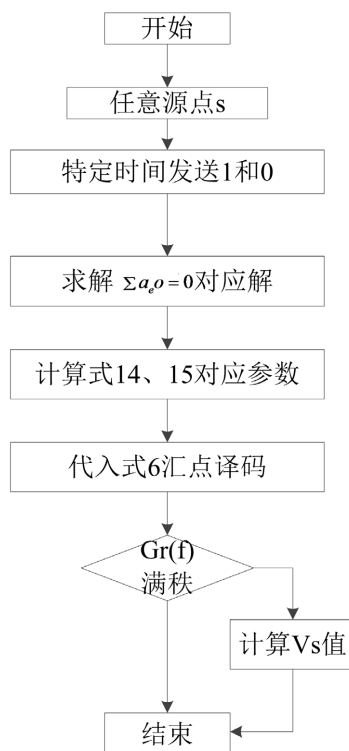


Figure 3. Flow chart of algorithm 2

图 3. 算法 2 流程图

### 3.3. 容量区域确定算法

从 3.1 节可知, 矩阵  $\{G_r(f)\}_{r \in R}$  可以获得所有源节点的既定传输速率。如果本地编码内核无法达到该传输速率时, 信息将会丢失, 使得从接收序列  $o$  中成功恢复发送序列  $g$  的概率降低, 具体原因如下。

对于任意源点  $s \in S$ , 矩阵  $G_{r,s}(f)$  中存在  $V_s'$  个线性独立的列向量  $c_{s,1}, c_{s,2}, \dots, c_{s,V_s'}$ , 那么  $(V_s')_{s \in S}$  为当前编码内核条件下汇点  $r$  的可达速率, 任意源点  $s$  能够在向量集合  $\{c_{s,1}, c_{s,2}, \dots, c_{s,V_s'}\}$  对应的链路上发送非零序列  $g_{s,i}$ , 其它链路上发送零序列, 即

$$\mathbf{g}_{s,i} = \begin{cases} \text{非零}, i \in \{c_{s,1}, c_{s,2}, \dots, c_{s,V_s}\} \\ 0, i \notin \{c_{s,1}, c_{s,2}, \dots, c_{s,V_s}\} \end{cases}$$

此时式(6)可简化为

$$C_r(f)\mathbf{o} = \tilde{G}_r(f)\tilde{\mathbf{g}} \tag{16}$$

$\tilde{G}_r(f)$  为  $V_s'$  个线性独立向量构成的矩阵,  $\tilde{\mathbf{g}}$  为除零之外的输入序列。由 3.1 节可知, 式(16)当且仅当  $\tilde{G}_r(f)$  为满秩矩阵时有解。换言之, 汇点  $r$  可达传输速率为  $V_s'$  时, 能够从输出序列  $\mathbf{o}$  中恢复发送序列  $\mathbf{g}$ 。

同时, 由可达速率定义可知, 任意源点  $s \in S$  以速率  $V_s - V_s'$  发送零值序列, 方可确保汇点  $r$  能够从接收序列  $\mathbf{o}$  中恢复发送符号  $\mathbf{g}$ , 则式(16)可解。由于  $\tilde{\mathbf{g}}$  可解析, 那么  $\tilde{G}_r(f)$  包含  $G_{r,s}(f)$  中的  $V_s'$  个线性独立的列向量。综上对任意给定网络和汇点  $r$ , 式 6 表征的输入输出序列之间的关系可写为

$$C_r(f)\mathbf{o} = G_r(f)\mathbf{g} = \sum_{s \in S} G_{r,s}(f)\mathbf{g}_s \tag{17}$$

根据式(17)可确定源点  $s$  的容量区域。

## 4. 仿真分析

### 4.1. 参数设置

算法测试基于 Linux 进行, 包括硬件、软件和算法参数三部分。硬件为算法的测试提供存储和计算功能; 软件提供仿真测试平台, 包括系统、网络仿真器(Network Simulator, NS)、图形处理软件、内置协议等, 协议采用按需距离向量路由(Ad hoc On-demand Distance Vector, AODV)协议; 模型参数确保算法顺利运行, 其中本地/全局编码内核参照文献[20]设定。主要参数设置如表 1 所示。

**Table 1.** Parameter set in simulation  
**表 1.** 仿真参数设置

硬件		软件	
参数	取值	参数	取值
处理器	i5-9500	Matlab	2009a
内存	8G	NS	allinone-3.27
硬盘	机械 1Tb	协议	AODV
		系统	Ubuntu
模型			
参数	取值	参数	取值
源信号	伯努利序列	节点数	100
仿真时间	10 s	移动模型	随机路点模型
仿真区域	1 km × 1 km	分组大小	1500 字节
频率	400 Mhz	传输速率	11 Mbps

### 4.2. 性能参数

算法性能包括时延、吞吐量、丢包率、控制开销等指标[21]。对于有环 CNC 网络而言, 时延(Delay)



可侧面反映算法效率问题, 主要包括处理时延和传播时延, 前者由 NC 编码译码产生, 后者由网络环路导致, 其值由 trace 文件往返时间统计值  $D_t$  的 1/2 确定, 在此用归一化值表征, 即统计时延平均值与最大时延  $D_{\max}$  比值; 吞吐量(Throughput)反映了网络容量的改进程度, 其值为单位时间内传输的分组数目  $P_t$ ; 丢包率(Loss)为无量纲参数, 反映节点传输速率阈值, 与容量区域密切相关, 其值为未成功接收分组  $P_l$  与总发送分组  $P_s$  的比值; 开销(Overhead)为有效数据分组以外的其它控制分组及复制分组数目, 其值为网络总分组数  $P_n$  与实际有效分组  $P_v$  的差值。各参数表达式分别为

$$\text{Delay} = D_t / 2D_{\max}$$

$$\text{Throughput} = P_t / t$$

$$\text{Loss} = P_l / P_s$$

$$\text{Overhead} = P_n - P_v$$

### 4.3. 仿真结果

在此对算法增加前(记为改进前)、增加后(记为改进后)与理想(记为理论值)性能分别进行对比分析, 结果如图 4~7 所示。

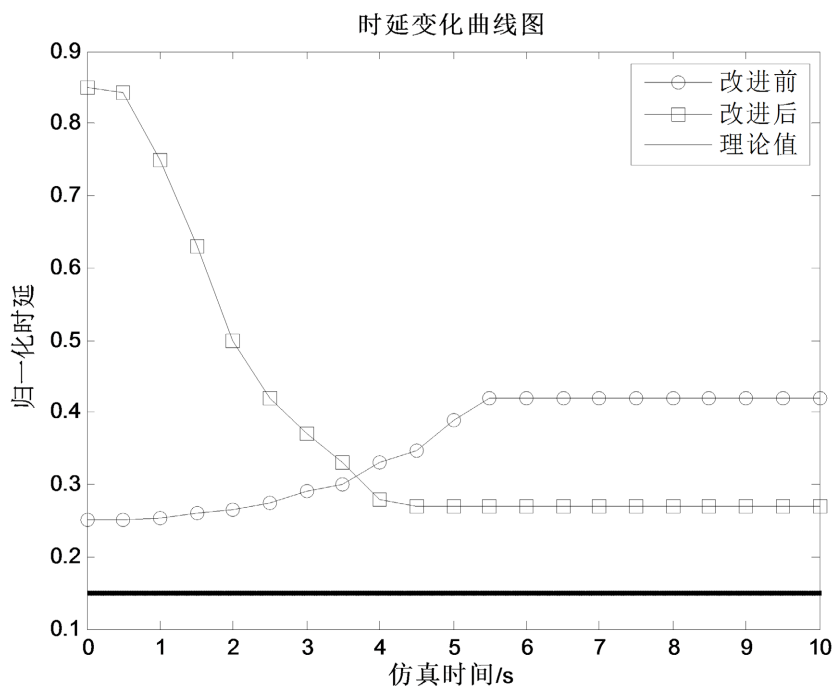


Figure 4. Diagram of normalized delay  
图 4. 归一化时延曲线图

从图 4 中不难看出, 总体而言, 改进前后算法时延均高于理想值, 且平稳状态( $t > 5$  s)改进后时延比改进前低约 37%。具体而言, 改进前、改进后时延变化趋势略有差别, 前者先递增后平稳, 后者递减至平稳状态。原因在于: 一是初始状态无论是否有数据传输, 网络需要等待全局编码内核和本地编码内核启动, 产生了高时延; 二是节点在初始化过程中, 排队和传输时延保持不变的情况下, 处理时延略有增加, 归一化时延也提高; 三是引入 CNC 后, 随着网络容量的提升, 传输的实际分组数目减少, 碰撞概率降低, 时延也随之降低; 四是引入了译码算法, 避免了有环网络的环路时延, 降低了网络时延。

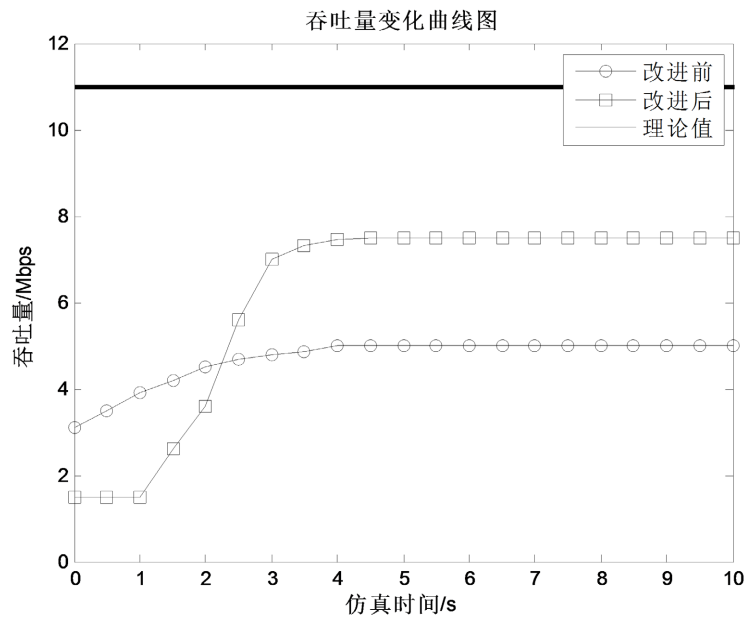


Figure 5. Diagram of throughput  
图 5. 吞吐量曲线图

从图 5 中可以看出, 吞吐量随着时间的增加, 呈现先递增、后稳定的趋势, 而且平稳时间内改进后的吞吐量比改进前高约 3.2 Mbps。原因在于: CNC 初始阶段更多数据用于处理全局和本地编码内核, 实际分组成功传输量较少, 而改进前协议在控制分组交互完成后即转发数据分组, 因此网络运行初始阶段 CNC 吞吐量低于改进前。当网络状态趋于稳定后, CNC 网络以较少的资源传输更多的分组直至稳定, 而改进前网络随着时间增加, 参与节点数递增, 分组碰撞重传概率增加, 并逐步趋于稳定值, 因而吞吐量平稳值低于 CNC 对应值。

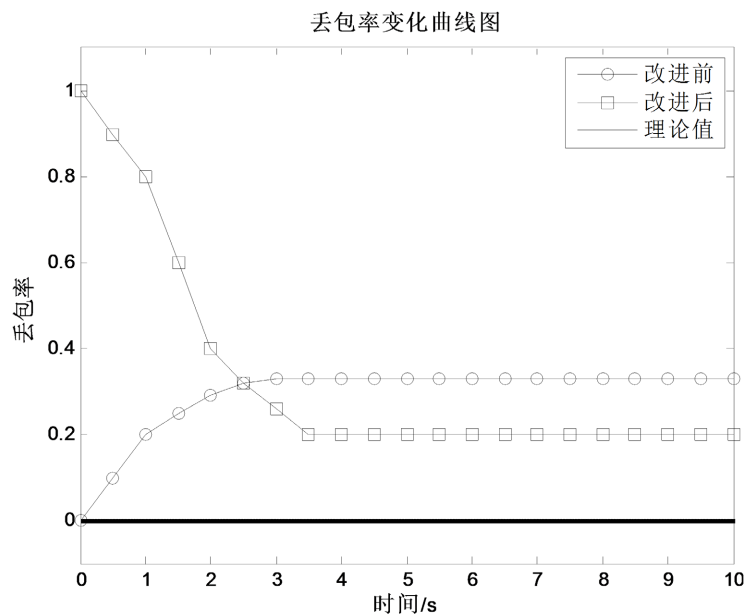


Figure 6. Diagram of packet loss rate  
图 6. 丢包率曲线图

从图 6 中可以看出, 网络运行初始阶段, 改进后网络丢包率高于改进前, 网络平稳后, 改进后丢包率比改进前低约 25%。原因在于: 初始阶段 CNC 能够正常传输的为本地和全局编码内核, 该时期内传输的所有数据分组基本上均被丢弃, 前期网络丢包率较高, 而传统网络发现路由即转发传输, 受通信链路失效性的影响, 存在部分分组丢弃的情况, 但整体低于 CNC 网络。随着时间的推移, CNC 网络 and 传统网络均趋于稳定状态, 改进前后丢包率也会趋于稳定。CNC 由于各数据分组基于内核函数作相关运算, 链路负载降低, 分组碰撞概率也随之降低, 而改进前网络通过路由协议重传机制确保分组成功传输, 分组碰撞概率高于 CNC, 与之对应的网络丢包率相对较高。

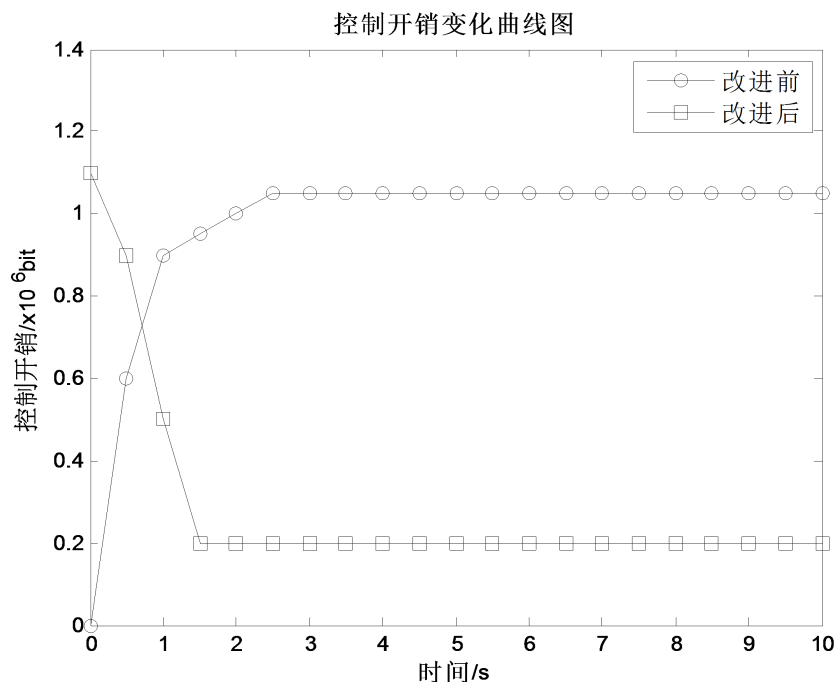


Figure 7. Diagram of control overhead

图 7. 控制开销曲线图

从图 7 中可以看出, 改进后控制开销比改进前低约 0.9 M bit, 原因在于: CNC 传输的为编码分组, 而改进前传输的为转发分组, 编码分组数量明显低于转发分组, 其原因与时延、吞吐量、丢包率等原因一致, 在此不作赘述。

## 5. 结论

论文在分析现有 CNC 译码机制的基础上, 指出存在的容量区域确值和汇点初始化问题, 并区分节点是否具有重置/清零功能, 差异化提出相应的初始化算法。为防止汇点分组溢出, 设计了容量区域参数确值算法, 并对归一化时延、吞吐量、丢包率和控制开销等性能参数进行仿真测试。但算法仍存在以下不足: 一是初始化算法和容量确值算法均假设生成矩阵为满秩, 而实际中存在非满秩情形; 二是各分组传输时延为一个时间单元, 避免了同步开销, 未考虑传输速率增加或降低时的情形。后期将针对上述不足作进一步细化研究。

## 基金项目

重大专项(XX2020F00003-XXX)、省部级课题(20191A010001)、大学基础理论课题(2021043、202121)。

## 参考文献

- [1] 欧莽, 汪继文. VANETs 中基于分布式 TDMA 的协作网络编码方法[J]. 华南理工大学学报, 2020, 48(1): 104-113.
- [2] 廖勇. 基于网络编码的多跳 D2D 通信性能研究[D]: [硕士学位论文]. 重庆: 重庆邮电大学, 2019.
- [3] Chen, Y., Wu, E.H. and Chen, G. (2017) Bandwidth-Satisfied Multicast by Multiple Trees and Network Coding in Lossy MANETs. *IEEE Systems Journal*, **2**, 1116-1127. <https://doi.org/10.1109/JSYST.2015.2406756>
- [4] 金振坤. 网络编码中优化问题研究[D]: [博士学位论文]. 武汉: 华中科技大学, 2018.
- [5] Li, X., Jiang, T., Zhang, Q. and Wang, L. (2009) Binary Linear Multicast Network Coding on Acyclic Networks: Principles and Applications in Wireless Communication Networks. *IEEE Journal on Selected Areas in Communications*, **5**, 738-748. <https://doi.org/10.1109/JSAC.2009.090614>
- [6] Kumwilaisak, W. (2010) On the Decodable Probability Bound of Linear Network Coding in Acyclic Lossy Networks. *Proceedings of TENCON 2010, IEEE Region 10 Conference*, Fukuoka, 21-24 November 2010, 835-840. <https://doi.org/10.1109/TENCON.2010.5686574>
- [7] Li, S.R. and Sun, Q.T. (2011) Network Coding Theory via Commutative Algebra. *IEEE Transactions on Information Theory*, **1**, 403-415. <https://doi.org/10.1109/TIT.2010.2090227>
- [8] Magli, E., Wang, M., Frossard, P. and Markopoulou, A. (2013) Network Coding Meets Multimedia: A Review. *IEEE Transactions on Multimedia*, **5**, 1195-1212. <https://doi.org/10.1109/TMM.2013.2241415>
- [9] Samadi Khaftari, V., Esmaeili, M. and Gulliver, T.A. (2017) Construction of MDS Convolutional Error-Correcting Network Codes over Cyclic Networks. *IEEE Transactions on Communications*, **6**, 2305-2318. <https://doi.org/10.1109/TCOMM.2017.2680441>
- [10] Li, S.Y. and Sun, Q.T. (2011) Network Coding Theory via Commutative Algebra. *IEEE Transactions on Information Theory*, **1**, 403-415. <https://doi.org/10.1109/TIT.2010.2090227>
- [11] Erez, E. and Feder, M. (2010) Efficient Network Code Design for Cyclic Networks. *IEEE Transactions on Information Theory*, **56**, 3862-3878. <https://doi.org/10.1109/TIT.2010.2050934>
- [12] Ho, T., Medard, M., Koetter, R., et al. (2006) A Random Linear Network Coding Approach to Multicast. *IEEE Transactions on Information Theory*, **10**, 4413-4430. <https://doi.org/10.1109/TIT.2006.881746>
- [13] Guo, W., Cai, N. and Sun, Q.T. (2012) Time Variant Decoding of Convolutional Network Codes. *Communications Letters*, **16**, 1656-1659. <https://doi.org/10.1109/LCOMM.2012.080312.120789>
- [14] Guo, W., Shi, X., Cai, N. and Medard, M. (2013) Localized Dimension Growth: A Convolutional Random Network Coding Approach to Managing Memory and Decoding Delay. *IEEE Transactions on Communications*, **61**, 3894-3905. <https://doi.org/10.1109/TCOMM.2013.071013.120857>
- [15] 弋改珍. 网络编码理论研究综述[J]. 无线互联科技, 2019, 16(8): 35-36, 50.
- [16] Etzion, T. and Wachter Zeh, A. (2018) Vector Network Coding Based on Subspace Codes Outperforms Scalar Linear Network Coding. *IEEE Transactions on Information Theory*, **4**, 2460-2473. <https://doi.org/10.1109/TIT.2018.2797183>
- [17] Lin, S., Fu, L., Xie, J. and Wang, X. (2016) Hybrid Network Coding for Unbalanced Slotted ALOHA Relay Networks. *IEEE Transactions on Wireless Communications*, **1**, 298-313. <https://doi.org/10.1109/TWC.2015.2472409>
- [18] Van Eeghem, F., Sørensen, M. and De Lathauwer, L. (2017) Tensor Decompositions with Several Block-Hankel Factors and Application in Blind System Identification. *IEEE Transactions on Signal Processing*, **65**, 4090-4101. <https://doi.org/10.1109/TSP.2017.2695445>
- [19] Hanna, M.T. (2017) A Discrete Fractional Hankel Transform Based on the Eigen Decomposition of a Symmetric Kernel Matrix of the Discrete Hankel Transform. *Proceedings of 60th International Midwest Symposium on Circuits and Systems (MWSCAS)*, Boston, 6-9 August 2017, 479-482. <https://doi.org/10.1109/MWSCAS.2017.8052964>
- [20] Yeung, R.W. (2008) *Information Theory and Network Coding*. Springer, Berlin.
- [21] Feng, W., Luo, H., Sun, B., et al. (2017) Performance Analysis of Sliding Window Network Coding for Energy Efficient in MANETs. *Proceedings of 7th IEEE International Conference on Electronics Information and Emergency Communication (ICEIEC)*, Macau, 21-23 July 2017, 219-222. <https://doi.org/10.1109/ICEIEC.2017.8076548>