

面向高突发的交织RLNC编译码研究

王威鹏¹, 杜伟庆¹, 陈杰², 李业³, 方毅⁴, 陈平平^{1*}

(1. 福州大学物理与信息工程学院, 福建福州 350108; 2. 厦门亿联网络技术股份有限公司, 福建厦门 361000;
3. 南通大学信息科学技术学院, 江苏南通 226019; 4. 广东工业大学信息工程学院, 广东广州 510006)

摘要: 5G新型业务场景要求传输的低时延和高可靠性, 应用层前向纠错编码(Forward Error Coding, FEC)可以快速恢复丢失数据包, 但是在高突发丢包信道的译码恢复性能不佳. 针对这个问题, 本文提出基于交织的流式随机线性网络编码(Interleaving Stream-Random Linear Network Coding, IS-RLNC)方案, 通过等间隔编码源数据包, 并将编码包分散插入源数据包的发送队列中. 该方案在保持低译码复杂度和实时性的同时, 提高突发情况下的恢复性能. 仿真实验表明, 在高突发信道环境下, IS-RLNC具有比里德-所罗门码(Reed-Solomon, RS)和滑动RLNC(Caterpillar RLNC, CRLNC)更低的译码失败概率. 此外, 在限制时延的应用场景中, IS-RLNC能够比RS码恢复更多的数据包. 最后, 为了达到最佳性能, 实验表明了不同突发长度信道具有不同的交织深度.

关键词: URLLC; 应用层FEC; RLNC; 交织编码; 流式编码

基金项目: 国家自然科学基金(No.62171135)

中图分类号: TN929.5

文献标识码: A

文章编号: 0372-2112(2025)07-2167-11

电子学报URL: <http://www.ejournal.org.cn>

DOI: 10.12263/DZXB.20250080

Research on Interleaved RLNC Coding and Decoding for High Burst

WANG Wei-peng¹, DU Wei-qing¹, CHEN Jie², LI Ye³, FANG Yi⁴, CHEN Ping-ping^{1*}

(1. School of Physics and Information Engineering, Fuzhou University, Fuzhou, Fujian 350108, China;

2. Xiamen Yealink Network Technology Company Limited, Xiamen, Fujian 361000, China;

3. School of Information Science and Technology, Nantong University, Nantong, Jiangsu 226019, China;

4. School of Information Engineering, Guangdong University of Technology, Guangzhou, Guangdong 510006, China)

Abstract: 5G new service scenarios require low latency and high reliability of transmission. Forward error coding (FEC) at the application layer can quickly recover lost data packets, but the decoding recovery performance in high burst packet loss channels is poor. To solve this problem, this paper proposes an interleaved streaming-random linear network coding (IS-RLNC) scheme, which encodes source packets at equal intervals and inserts encoded packets into source packets to send. This scheme improves the recovery performance in burst condition while keeping low decoding complexity and real-time performance. Experimental results show that IS-RLNC has a lower decoding failure probability than Reed-Solomon (RS) and Caterpillar RLNC (CRLNC) in high burst channel environments. IS-RLNC is able to recover more packets than RS codes in scenarios with tight delay constraints. Finally, experiments show that the optimal decoding performance for different burst channels has different interleaving depths.

Key words: URLLC; FEC in application layer; RLNC; interleaved coding; stream coding

Foundation Item(s): National Natural Science Foundation of China (No.62171135)

1 引言

超高可靠低时延通信^[1,2](Ultra Reliable&Low Latency Communication, URLLC)是5G系统的三大场景之一, 具有超高可靠性和极低的传输延迟, 是工业自动化、远程医疗行业和智能交通的重要通信技术^[3-6]. 在

无线数据传输过程中, 存在干扰、多径效应和衰减等因素的影响, 导致传输误码. 同时, 网络拥塞、路由表过期和缓存溢出等问题也会引起数据包丢失^[7]. 为了确保数据传输的高可靠性, 上层协议通常采用重传机制来应对数据包丢失. 然而, 在流媒体传输和控制信令传输

等对实时性要求较高的场景中,重传会引入较大的时延,在高往返时延(Round Trip Time, RTT)的信道中尤为明显^[8].

因此,为避免重传时延,需要在上层协议中加入前向纠错编码(Forward Error Coding, FEC). 目前常用的应用层 FEC 有阶梯型低密度奇偶校验码(Low Density Parity Check-staircase, LDPC-staircase)^[9]、里德-所罗门码(Reed-Solomon, RS)^[10]和喷泉码^[11]. 而在工业物联网(Industrial Internet of Things, IIoT)等低延迟需求场景中,随机线性网络编码^[12,13](Random Linear Network Coding, RLNC)是一种更合适的丢包恢复解决方案. 该类编码支持滑动窗口编码,以及根据网络状况动态调整编码参数,以此实现低延迟、良好鲁棒性和高效的数据传输^[14].

然而,传统的块 RLNC(Block-RLNC, B-RLNC)^[15,16]中分块编码的思路会引入较大解码延迟,块中靠前的包需要被阻塞到块末尾的编码包到来. 为了减小延迟, Karzand 等人^[7]提出流式 RLNC(Stream-RLNC, S-RLNC)方案,该方案引入流式编码的思想,即在源数据包的发送队列中等间隔分散插入编码数据包,能够更快恢复丢失^[17]. 同时结合无限扩展的编码窗口,编码包含所有未确认的源数据包,可以极大提高译码恢复率. 但需要通过反馈来及时关闭编码窗口,因此不适用于广播这类无反馈信道的应用场景. 因此,文献[18]将 S-RLNC 结合滑动窗口机制,提出滑动 RLNC(Caterpillar RLNC, CRLNC)方案,通过不断滑动固定尺寸的编解码窗口,该方案不依赖反馈,同时能实现更低的按序时延^[19]. 但 CRLNC 牺牲了一定可靠性,因此在文献[18]的工作基础上,文献[20]提出结合反馈的 CRLNC(Caterpillar RLNC with FeedBack, CRLNC-FB),针对性重传源包来提高可靠性. 目前,RLNC 的许多研究仍聚焦在努力优化 S-RLNC 的编码策略^[8,21-24]和完善理论性能的分析上^[25,26].

在实际网络传输中,存在设备快速移动、突发高流量等信道恶化的情况,导致数据流高突发丢失. S-RLNC^[7,17]会因为长时间突发丢包阻塞解码窗口,影响实时性;CRLNC 则在高突发丢包下几乎无法恢复丢失,虽然可以通过增大编码窗口改善,但会带来较高的计算复杂度. 对于该类差错,一个有效的改进方法是对数据包进行交织编码^[27,28],文献[27]将冗余插入每个源数据包,其中冗余包间隔地包含前面数据包的部分数据;文献[28]提出实时的 B-RLNC 交织方案,其不改变源数据包原本的发送顺序,通过改变编码的顺序来避免交织时延.

因此,本文在 CRLNC 中结合交织的思想,提出基于交织的流式随机线性网络编码(Interleaved Streaming-Random Linear Network Coding, IS-RLNC),通过等间隔

地选取源数据包参与编码,生成的编码包等间隔地插入于发送队列中,既扩大了编码的覆盖范围,同时保证计算复杂度不变. 仿真结果表明,在不同的突发丢包环境下,IS-RLNC 相较于 RS 码和 CRLNC 的丢包恢复性能更优,并且在限制最大按序时延的场景中,IS-RLNC 能够比 RS 码更快地恢复大部分数据包. 同时,适当增加 IS-RLNC 的交织深度能够进一步提高恢复性能,且相比于 CRLNC 的增大编码窗口方案,IS-RLNC 的计算复杂度更低,即更少的平均编码时间.

2 RLNC 相关工作

RLNC^[12,13]编码随机选取伽罗瓦域 GF(q)内的元素作为编码系数,并根据编码系数对源数据包进行加权组合得到编码包. 设 \mathbf{X} 为 n_s 个源数据包集合,编码矩阵 \mathbf{C}_e 包括对应的编码系数向量,大小为 $n_c \times n_s$. 最终发送的 n_s 个数据包集合 \mathbf{Y} 为

$$\mathbf{Y} = \mathbf{C}_e \mathbf{X} \quad (1)$$

其中, \mathbf{X} 为长度为 n_s 的列向量; \mathbf{Y} 为长度为 n_c 的列向量.

解码器在接收到的数据包中,取 n_s 个编码系数线性独立的数据包,设集合为 \mathbf{Y}_d . 根据 \mathbf{Y}_d 对应的编码系数生成解码矩阵 \mathbf{C}_d , 大小为 $n_s \times n_s$. 通过式(2),即可恢复出 \mathbf{X} :

$$\mathbf{X} = \mathbf{C}_d^{-1} \mathbf{Y}_d \quad (2)$$

其中 \mathbf{Y}_d 为长度为 n_s 的列向量.

由于编码包的编码系数是随机生成的,不同编码包可能线性相关. 选择的伽罗瓦域越大,越不容易产生线性相关,但是编解码的计算复杂度会更高,且解码产生的时延也会增加. 因此 RLNC 编码通常采用 GF(2^8), 可以实现极小的解码失败概率^[7],同时每个字节数据视为该有限域中的一个符号,可以进行简单高效的异或编码操作. 不同 RLNC 码型的编码如图 1 所示.

B-RLNC^[15]编码方案如图 1(a)所示. 对于收到的源数据包,会先将其进行分块处理,每块包含 k 个源数据包,先传输这些源数据包,然后对源数据包进行编码,生成 r 个编码包,编码码率 R 即为

$$R = \frac{k}{k+r} \quad (3)$$

S-RLNC^[7,17]编码方案如图 1(b)所示,编码器会维护一个无限制长度的编码窗口 E_w . 发送端每发出一个源数据包,编码器都会将其存入 E_w ,当且仅当接收端对源数据包按序反馈确认后,才能从 E_w 中将其移除. 每发送 l 个源数据包后,会紧接着发送一个编码包,其为 E_w 中所有源数据包的线性组合. 因此 S-RLNC 的码率 R 为

$$R = \frac{l}{l+1} \quad (4)$$

在这种编码策略下,接收端收到的编码包会包含

未确认的源数据包. 在时间足够长的情况下,若信道的擦除概率为 ε , 只要码率 $R > 1 - \varepsilon$, 那么源数据包便能完全恢复^[18]. 但是在高 RTT 的信道或高突发信道下, E_w 无法及时更新, 即较早存入 E_w 内的源数据包无法被移除, 导致编码的计算复杂度过高, 因此该编码策略要求信道能提供及时的反馈.

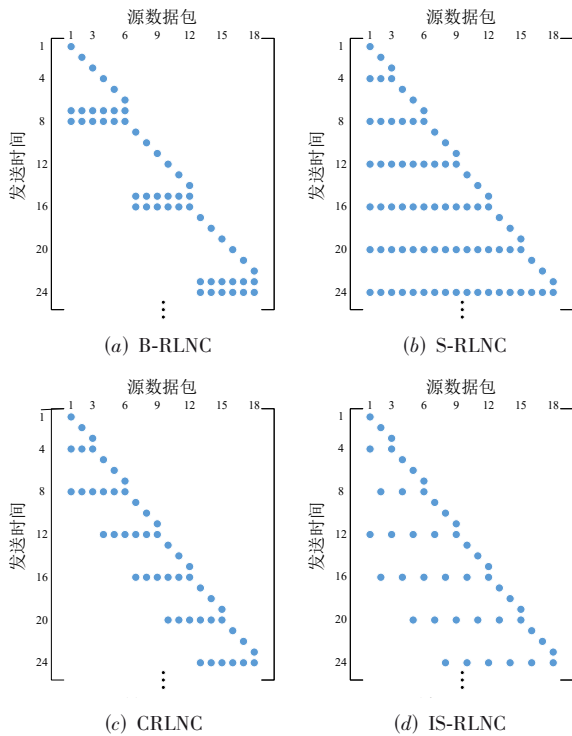


图 1 不同 RLNC 码型的编码示意图

如图 1(c) 所示, CRLNC^[18] 也采用了流式编码的思想, 即发送过程中等间隔 l 插入一个编码包, 码率如式 (4) 中所示. 与 S-RLNC 相反, 其限制 E_w 为固定尺寸, 因此不依赖于反馈, 编码的计算复杂度可以保持恒定. 同时, 在信道发生高突发丢失无法恢复时, 解码窗口 D_w 会进行推窗防止阻塞^[18], 即不再尝试译码恢复 D_w 起始位置的数据包, 更适用于实时性要求高的场景. 但是如果没结合重传方案, CRLNC 与 B-RLNC 一样, 无法保证完全恢复. 因此, CRLNC-FB^[20] 引入反馈, 提高了数据包恢复率. 但是, 仍存在 ARQ 反馈确认不及时导致的阻塞问题, 相对于 CRLNC, 实时性方面略有下降.

CRLNC 和 B-RLNC 类似, 编码包都是由固定数量的源数据包组合而成, 但是 B-RLNC 的各编码包的编码范围并不重叠, 而 CRLNC 的编码窗口允许重叠, 类似卷积码, 即编码包含当前块和前面块的源数据包. 因此, CRLNC 的丢包恢复速度比 B-RLNC 更快^[14]. 如果编码包涵盖的源数据包数量一致, 两者的恢复能力并无显著差别^[18].

3 系统模型

如图 2 所示, 本文研究应用层中的数据包 FEC 编译码. 编码器将应用程序传来的数据包缓存进队列, 并不断从队头中取出数据包进行编码发送. 同理, 解码器也设置队列缓存收到的数据包, 并不断取出数据包送入解码器进行译码恢复, 成功恢复的数据包将按序上传. 在编解码器中间的信道包含了传输层、网络层、链路层和物理层等. 数据包在接收端向上传递时, 错误的数据包无法通过上层 crc 校验, 因此接收端在应用层收到的数据包是正确的, 可将编解码器中间的信道等效为一个二元擦除信道^[29].

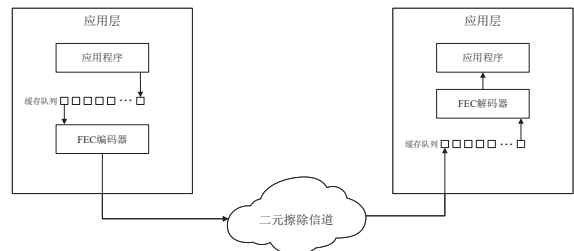


图 2 系统模型框图

3.1 编码方式

图 1(d) 展示了本文所提出的 IS-RLNC 编码方案. 编码包在发送队列中等间隔 l 插入, 定义 l 为插入编码包的步长. 编码窗口的大小设置为固定尺寸, 定义为 w_c . 参与编码的源数据包为等间隔选取, 定义交织深度的大小为 v , 则参与编码的数据包之间间隔为 $v-1$. 文献[25]中, 保持码率不变, 对随机插入编码包和等距插入编码包 2 种方案进行了对比. 其中等距插入编码包带来的对所有源数据包均等保护, 效果要比随机插入编码包带来的不等保护性能要更好. 在 IS-RLNC 中, 要保证每个源数据包都能受到相同的保护程度. 首先, 编码窗口的大小 w_c 应为步长 l 的整数倍. 否则会出现 2 个编码包之间的 l 个源数据包, 后发送的包要比先发送的包多被保护 1 次. 其次, 交织深度 v 与步长 l 必须呈现互质的关系.

定理 若每个源数据包受到的保护均等, 则需要 v 和 l 满足方程 $lx \equiv m \pmod{v}$, $m \in \{1, 2, \dots, v-1\}$ 对应任意 m 均有不重复的解 x , 即 v 和 l 互质.

证明 假定一个编码包跟在第 i 个源数据包后发送, 即该编码包保护了第 i 个源数据包. 由于交织深度为 v , 因此前面 $v-1$ 个源数据包, 即第 $i-v+1, i-v+2, \dots, i-1$ 个包还没被保护. 若要后续的 $v-1$ 个编码包保护这 $v-1$ 个源数据包, 即需要第 $i+lx, x \in \{1, 2, \dots, v-1\}$ 个源数据包后面紧跟的编码包能保护到第 $i-v+1, i-v+2, \dots, i-1$ 个源数据包. 为简化数学模型, 假定 $i=v$, 每个源数据包均受

到保护需要满足 $lx \equiv m \pmod{v}$, $m \in \{1, 2, \dots, v-1\}$ 对每个 m 取值均有对应的解 x . 若要均等保护, 则这 $v-1$ 个源数据包不能被相同的编码包保护, 即对应不同 m 的方程 $lx \equiv m \pmod{v}$, $m \in \{1, 2, \dots, v-1\}$ 的解 x 不重复.

若 v 和 l 互质, 则 $\gcd(v, l) = 1$, 可得 $\gcd(v, l)$ 能够整除任意的 m , $m \in \{1, 2, \dots, v-1\}$. 根据线性同余方程解的存在性定理, 可得对应任意的 m 均有对应的解 x .

设方程 $lx \equiv m \pmod{v}$, $m \in \{1, 2, \dots, v-1\}$ 中 m 取不同的值 m_1 和 m_2 , 对应的解为 x_1 和 x_2 . 若不同的源数据包被相同的编码包保护, 说明 $x_1 = x_2$, 则 $l(x_1 - x_2) \equiv 0 \pmod{v}$. 由于 l 与 v 互质, 则 $(x_1 - x_2) \equiv 0 \pmod{v}$, 说明 x_1 与 x_2 同余. 因此 x 在 $1, 2, \dots, v-1$ 范围内, 不存在方程 $lx \equiv m \pmod{v}$, $m \in \{1, 2, \dots, v-1\}$, m 取不同的值却有相同的 x 值可以求解. 因此, 要满足每个源数据包受到均等保护, 需要 v 和 l 互质. 证毕.

3.2 数据包设计

对于源数据包和编码包, 研究采用相同的包头结构, 如图3所示. 其中 `source_id` 表示进入编码器的第 i 个源数据包, 该数据包定义为 s_i , `repair_id` 表示由编码器生成的 j 个编码包, 该编码包定义为 c_j . 其中 `repair_id` 也作为随机数种子, 用于编解码器随机生成相同的编码系数. 随机数的生成可采用梅森选择算法^[30], 其具有周期长、分布均匀、生成快等特点. 当发送源数据包时, `repair_id` 置为 -1 , 而发送编码包时, `source_id` 置为 -1 . `first_pkt_id` 为编码包涵盖的最早的源数据包序号, `last_pkt_id` 为编码包涵盖的最晚的源数据包序号, 若是发送源数据包, 则 `first_pkt_id` 和 `last_pkt_id` 都置为 -1 . `pl_size` 为封装包头前的数据大小, 当应用程序传输大小不一致的数据包时, 这样能够取得编码窗口 E_w 中最大的数据包尺寸, 便于对不同大小的数据包进行基于 $GF(2^8)$ 的线性组合. `payload` 为从应用程序中获取的数据内容.

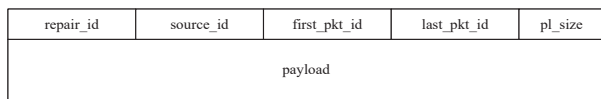


图3 编码包头结构图

3.3 解码方式

接收端维护一个解码器, 解码器会记录最近成功上传的源数据包的 `source_id`, 记为 i_{ord} . 若解码器新接收到的数据包的 `source_id` 不为 $i_{ord} + 1$, 即新到达接收端的包并非按序到达 (一般是过程中发生了突发丢失或者乱序), 则会将解码器激活.

解码采用动态高斯消元^[31]. 具体来说, 接收端根据数据包的 `repair_id` 和 `source_id` 生成数据包对应的编码系数, 并通过行变换后插入解码矩阵 C_d , 其中 C_d 始终

保持上三角形式. 这样做有2个好处, 一是可以方便判断 C_d 是否满秩, 以及收到线性相关的编码包可以尽早发现并丢弃; 二是可以提前开始解码的高斯消元操作. 相比而言, 常规的高斯消元先把 C_d 化成上三角后再化成单位矩阵, 动态解码则可以在接收数据包插入的步骤中先把 C_d 化成上三角, 从而将解码耗时分散到每个编码包的接收时刻.

图4展示了IS-RLNC解码器的算法流程, 在接收到对端传来的FEC数据包后, 通过解析包头信息, 对数据包进行相应的解码操作, 直到解码矩阵满秩后进行动态高斯消元译码, 并按序上传恢复的源数据包. 该解码算法可以避免网络乱序、数据包重复等影响的同时有效提升RLNC的解码效率.

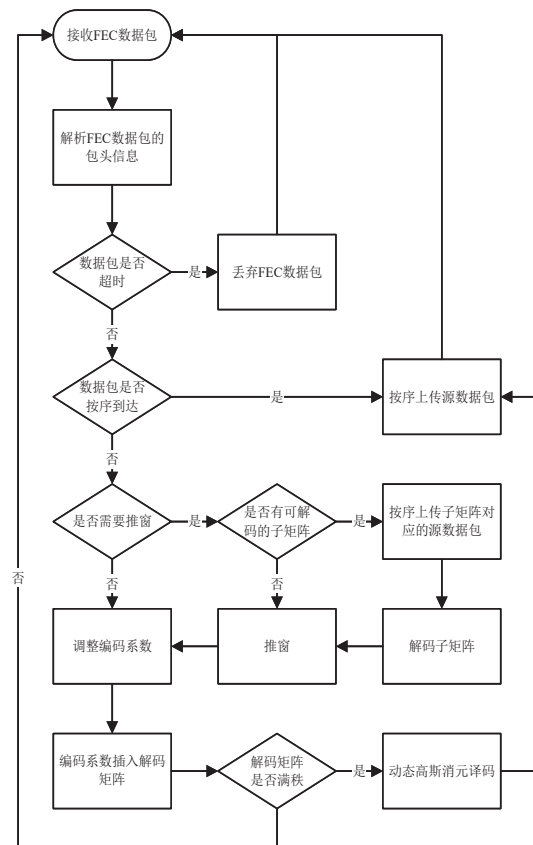


图4 解码算法流程图

如图5所示, 详细解释该解码过程, 假定时隙6~8连续丢包. 编码器参数与图1(d)中一致, 即编码窗口大小 w_c 为6, 步长 l 为3, 交织深度 v 为2. 假定每个数据包的发送和接收都在一个时隙内完成, 发送端按序发送数据包且每个包传输时延一致, 不会发生乱序现象. 在时隙6~8中, 信道中 s_5, s_6, c_2 数据包发生丢失, 在接收端, 解码器按序收到数据包 $s_1 \sim s_4$ (c_1 包无用, 直接丢弃) 并成功上传, 此时 $i_{ord} = 4$. 由于信道突发擦除, 接收端在

时隙9才收到新数据包 s_7 . 新数据包 source_id 为7, 不等于 $i_{ord} + 1$, 因此解码器激活, 接收端生成数据包对应编码系数, 并将其插入 C_d 中, 如图6(a)所示. 随后的数据包 s_8 和 s_9 也以相同方式按序插入 C_d 中, 如图6(b)所示.

在解码器中缓存着部分已经从解码窗口 D_w 中上传的源数据包, 编码系数会根据这些源数据包调整至当前 D_w 范围内再插入, 即让 D_w 范围外的源数据包不参与该编码包的线性加权组合. 若缓存的数据包不足以将编码系数调整至该范围内, 则丢弃该编码包. 接收端在 $s_7 \sim s_9$ 插入 C_d 后, 随后的编码包 c_3 会调整编码系数后插入 C_d , 如图6(c)所示. 后续按序到达的源数据包 $s_{10} \sim s_{12}$ 以及编码包 c_4 以同样方式插入 C_d 中, 如图6(d)所示. 此时 C_d 满秩, 高斯消元解码恢复出源数据包 $s_5 \sim s_{12}$ 并按序传至上层.

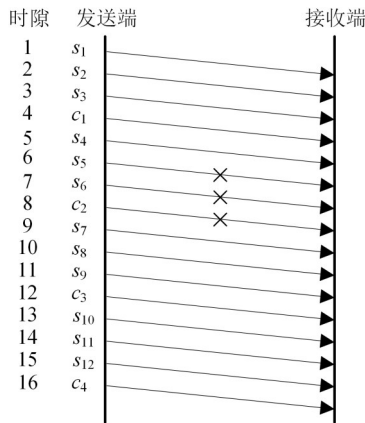
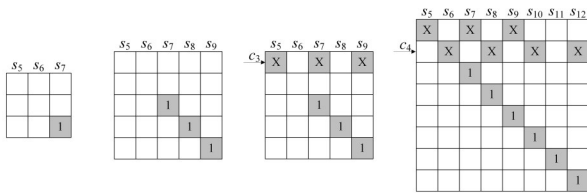


图5 信道突发示例1



(a) 收到 s_7 后 (b) 收到 s_8 和 s_9 后 (c) 收到 c_3 后 (d) 收到 c_4 后

图6 解码矩阵变化过程

解码窗口 D_w 的大小是动态的, 只有在 D_w 插入新数据包时, 才会实际分配内存空间, 可以一定程度提高解码端存储空间利用. 为提高实时性, 需要防止数据包长时间阻塞在 D_w 内, 因此 D_w 的大小设置了最大尺寸 w_d . 若新到来的数据包插入后, D_w 大小将超过 w_d , 便会进行推窗, 即保持解码窗口结束位置不动, 移动起始位置使 D_w 大小不超过 w_d . 推窗后, 仍阻塞在 D_w 的数据包可能可以译码恢复上传, 因此解码器会在推窗之后进行判断, 从新的起始位置开始是否存在可以解码的子矩阵 C_s . 若存在可解码的 C_s , 会将该 C_s 解码, 并将解码恢复的源数据包按序传至上层, 并再次推进 D_w .

为详细解释该推窗流程, 如图7所示, 我们假定时隙6, 10~12发生丢包, 编码器参数仍与图1(d)中一致. 设 D_w 的最大尺寸 w_d 为8, 不考虑乱序到达的情况下, 接收端一直接收到数据包 c_4 时 D_w 对应的解码矩阵 C_d 如图8(a)所示. 而后解码器收到 source_id 为13的数据包 s_{13} , 此时 D_w 大小已经为最大尺寸 $w_d=8$. 因此需要推掉数据包 s_5 所在的位置, 而后插入数据包 s_{13} , 此时 C_d 如图8(b)所示. 可以看出, 推窗之后 s_6 和 s_7 之间对应的 C_s 是满秩的, 因此解码器会先将这块 C_s 单独高斯消元解码, 并将恢复出来的 s_6 和 s_7 传至上层, 并移动 D_w 起始位置至 s_8 , 此时 C_d 如图8(c)所示.

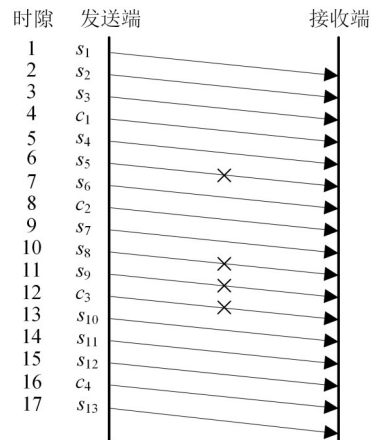
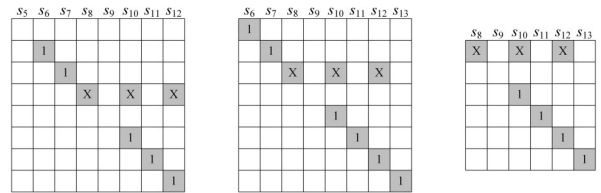


图7 信道突发示例2



(a) 收到 s_{12} 后 (b) 收到 s_{13} 后 (c) 子矩阵解码后

图8 子矩阵解码过程

3.4 二元擦除信道

目前常用的仿真分析的网络丢包模型有 Bernoulli 丢包模型^[32]、Gilbert 丢包模型^[33]、Gilbert-Elliot 丢包模型^[34]以及四状态 Markov^[35]模型等. 在实际的网络传输环境中, 信道的丢包通常不是均匀分布的, 而是具有突发性^[36]. 因此, 本文主要研究突发丢包的连续二元擦除信道, 丢包机制选用 Gilbert 模型, 具体如图9所示, 该模型是一个两态马尔可夫链. 其中, 状态 G 表示“好”状态, 该状态下不会产生丢包; 状态 B 表示“坏”状态, 该状态必然产生丢包; β 表示“坏”状态到“好”状态的转移概率; γ 表示“好”状态到“坏”状态的转移概率. 信道突发差错的期望 $E[B]$ (坏状态下的期望停留时间) 为

$$E[B] = \frac{1}{\beta} \quad (5)$$

信道的总体丢包率 ε (坏信道的稳态概率) 为

$$\varepsilon = \frac{\gamma}{\gamma + \beta} \quad (6)$$

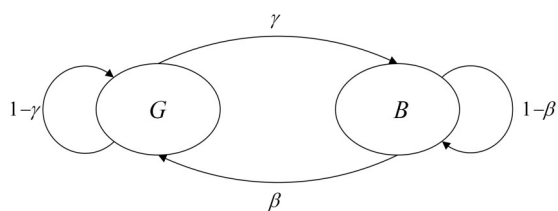


图9 Gilbert信道状态转换图

4 仿真实验

在本文的仿真实验中,硬件基于CPU i5-10400,软件系统基于Ubuntu 20.04平台测试.发送的数据包的内容为随机生成,数据包尺寸保持为200 Byte,每次测试随机生成 10^6 个包按顺序传入编码器.每个数据进行100次测试,最后结果取平均得到.

4.1 评价指标

评价应用层FEC码型的性能,主要从恢复性能、实时性能、对设备计算能力的要求3个方面考虑,对应译码失败概率、按序时延、平均编码时间3个指标.

译码失败概率:设原始数据包数量为 N ,经过解码器恢复后仍然丢失的数据包数量为 N' ,译码失败概率为 N'/N ,译码恢复概率即为 $(1 - N')/N$.

按序时延:假定每个数据包的发送和接收都在一个时隙内完成.发生突发擦除后,收到的数据包会被解码器缓存来恢复丢失的源数据包,其中源数据包在解码器中被缓存的时隙即为按序时延^[37].

平均编码时间:为平均每个源数据包经过编码器的用时,该时间用Wall time来表示,该指标可以很好反映IS-RLNC编码的计算复杂度^[38].

4.2 与RS码的性能比较

本文比较RS码和IS-RLNC的性能,码率 R 均为 $4/5$,其中RS码以国际互联网工程任务组(The Internet Engineering Task Force, IETF)发布的标准化文档RFC 5510定义的框架设计,没有结合交织技术.RS码的码长为255,每块源数据包数量为204,编码包数量为51.为保证每个编码包保护的源数据包数量与RS码一致,论文设置IS-RLNC的编码窗口大小 $w_c = 204$,步长 $l = 4$.仿真不同的突发期望长度 $E[B]$ 来观察译码失败概率的变化,测试结果如图10所示.图10的丢包率 ε 设置分别为5%、10%和15%.首先可以看到,随着 $E[B]$ 的增加,译码失败概率上升.这是因为突发错误越集中,突发错误越容易超出译码恢复能力,导致译码失败概率接近于Gilbert信道的稳态丢包率.其次,在相同信道条件下,IS-RLNC译码失败概率都显著低于RS码,且交织深度 v 越高,译码失败概率越低.但是在图10(c)的开始阶段中, $E[B] < 5$ 时, $v = 5$ 的IS-RLNC译码失败概率高于 $v = 3$ 的.这是因为在低突发的环境下,交织深度未必越高越好,在后续4.2节将会进一步分析说明.

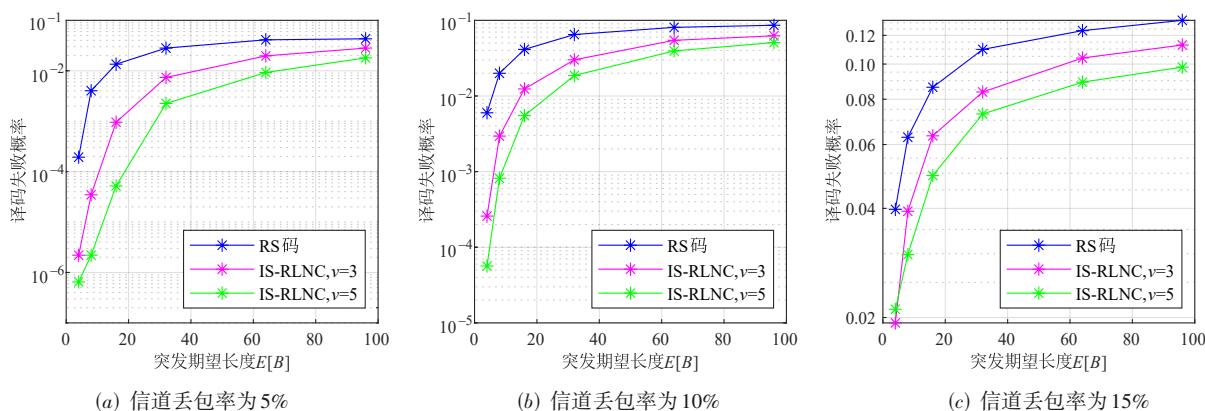


图10 不同突发期望长度下的译码失败概率比较

不同应用程序对数据包的按序时延的容忍度不同,如实时流媒体传输对实时性要求较高,而像网页响应这类对实时性要求更低,可以容忍更高的按序时延.研究设置了接收端允许的最大按序时延 T_s ,当接收到的数据包在 D_w 中被阻塞,超过了 T_s 后,该数据包会被推出并传至上层.因此,图11和图12比较了在不同 T_s 下,RS码和 $v = 3$ 的IS-RLNC的译码性能.图11信道丢

包率 $\varepsilon = 10\%$, $E[B]$ 分别为4、8、16;图12中 $E[B] = 8$, ε 分别为5%、10%、15%.可以看出,当 $T_s < 200$ 时,IS-RLNC能够恢复出更多的数据包.这是因为它将编码包分散发送,在发生少量错误时,能更快地恢复丢失的数据包.而RS码由于编码包都集中在数据块的末尾,当块中前面的源数据包丢失时,需要等到块末尾的编码包到达才能恢复,因此RS码的恢复性能较差.

当 $T_s > 300$ 时,由于 IS-RLNC 通过交织扩大编码范围,能够恢复出更长的突发丢失,因此仍表现出更好的性能. 在 T_s 足够大时,RS 码结合交织也能提升译码恢复概率. 但是结合交织的 RS 码由于需要缓存数据包以改变发送顺序,会引入额外的交织时延^[39],严重影响实时性. 在本文方案中,无须改变发送顺序,而是通过间隔编码来扩大编码范围,有效避免了额外的交织时延. 然而,在 T_s 为 200~300 时,RS 码恢复性能略好. 这是因为 RS 码长设置 255,源数据包在延迟 200~300 时间内,

能被块末尾的 51 个编码包都保护,而 IS-RLNC 由于交织编码的存在,对数据包的保护较为稀疏,在延迟 200~300 时间内,每个源数据包仅能被少量编码包保护,而 RS 码在该范围内充分利用了所有的编码包,因此恢复能力更强. 同时,图 11 和图 12 可以观察到,随着 $E[B]$ 或 ε 的增加,IS-RLNC 在低按序时延条件下的优势逐渐下降. 这主要是由于交织带来的稀疏保护,IS-RLNC 在高突发错误的实时性下降,但是依然对 RS 码具有性能优势.

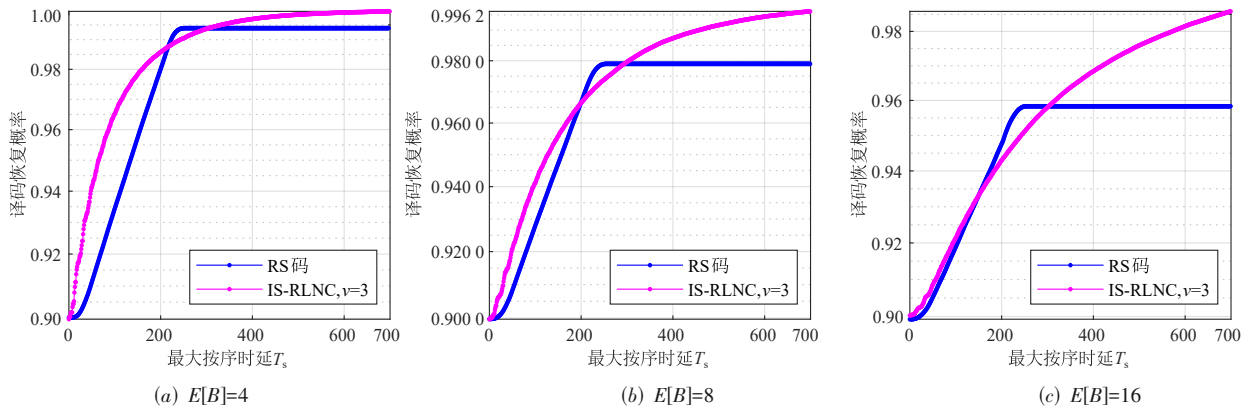


图 11 不同 T_s 下译码恢复概率的比较($\varepsilon=10\%$)

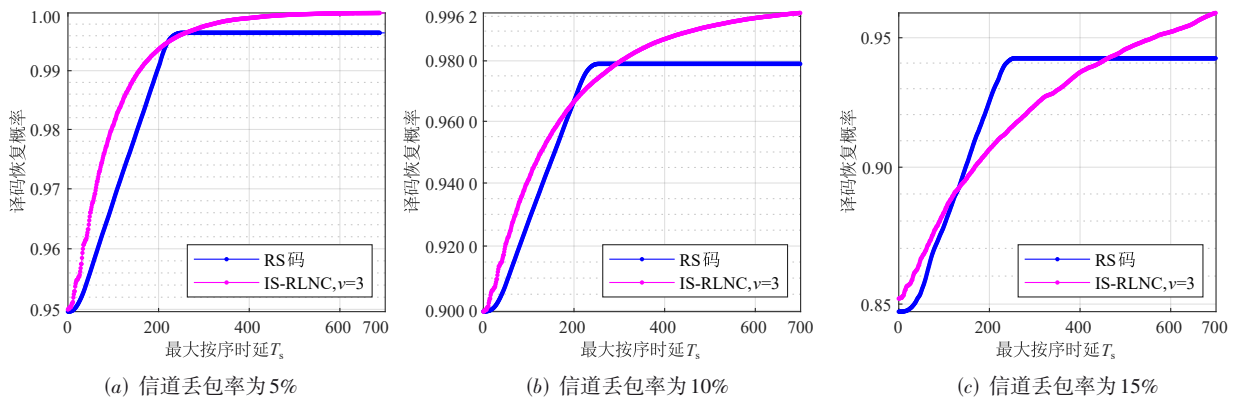


图 12 不同 T_s 下译码恢复概率的比较($E[B]=8$)

4.3 与 CRLNC 的性能比较

比较不同交织深度下的 IS-RLNC 性能. 保持编码窗口大小 $w_c=32$, 解码窗口最大尺寸 $w_d=48$ (即 w_c 的 1.5 倍). 需要注意的是,交织深度为 1 的流式 RLNC 也等效于 CRLNC. 图 13 和图 14 测试了不同码率(码率分别为 1/2、2/3、4/5、8/9、16/17)的译码失败概率,由式(4)可得码率 R 和编码步长 l 线性相关,因此图 13 和图 14 也可以看作不同编码步长下的译码失败概率变化. 其中图 13 突发期望长度 $E[B]=32$,信道丢包率 ε 分别为 3%、5%、7%. 图 14 中 $\varepsilon=5\%$, $E[B]$ 分别为 16、32、48. 首先可以看出,随着码率的提高,即编码步长的增加,译码失败概率增大. 因为编码窗口大小不变而码率增高时,每个

源数据包受到的编码包保护减少,发生错误时更难恢复. 其次,在不同码率下,IS-RLNC 要比 CRLNC 有更低的译码失败概率,并且交织深度越高,译码失败概率越低. 在图 13(a)中,即 $\varepsilon=3\%$, $E[B]=32$,当码率为 1/2 时, $v=1$ 时译码失败概率约为 10^{-2} ;而 $v=3$ 时,译码失败概率降低至 10^{-3} ; $v=5$ 时,译码失败概率降低至 10^{-4} .

图 15 测试了不同编码窗口大小 w_c (分别为 48、64、80、96、112、128) 的译码失败概率,其中解码窗口大小 w_d 保持为 w_c 的 1.5 倍,可以平衡译码的可靠性和实时性^[18]. 信道的丢包率 ε 设为 5%,突发期望长度 $E[B]$ 分别为 16、32、48. 可以看出,随着 w_c 变大,译码失败概率有着明显的下降. 这是因为 w_c 增大,而码率 R (即编码

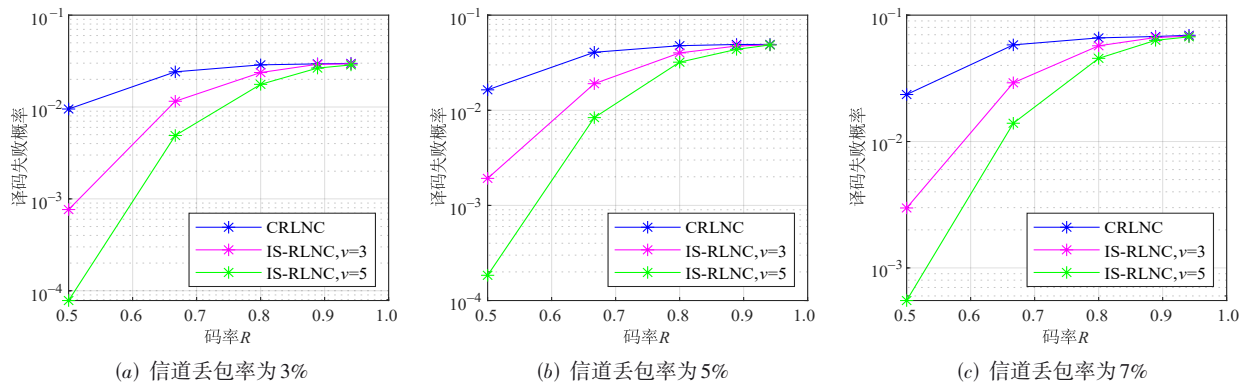


图13 不同码率下译码失败概率的变化($E[B]=32$)

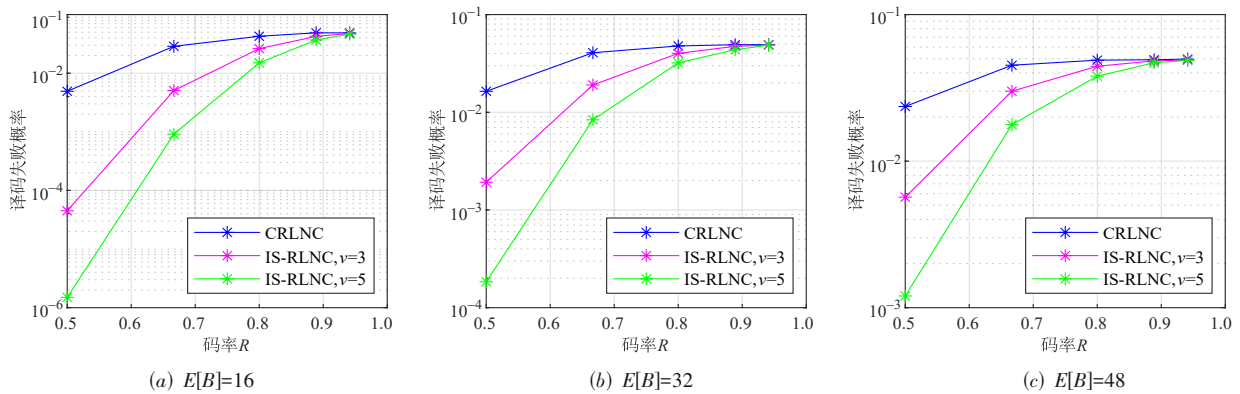


图14 不同码率下译码失败概率的变化($\epsilon=5\%$)

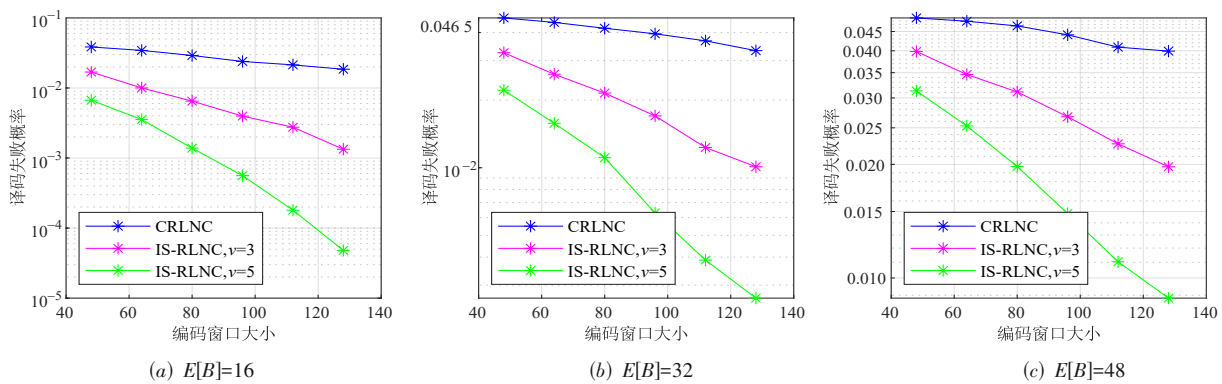


图15 不同编码窗口大小下译码失败概率的变化($\epsilon=5\%$)

步长 l) 不变, 每个编码包会保护更多的源数据包, 同时源数据包可以被更多的编码包保护. 其次, 在不同的突发丢包环境下, 在相同 w_e 下, IS-RLNC 均比 CRLNC 具有更低的译码失败概率, 有效提高了突发信道下的丢包恢复能力.

4.4 编码复杂度比较

IS-RLNC 相比 CRLNC 具有更好的抗突发恢复性能, 这是由于扩展了编码包的编码范围, 容忍更长的突发. CRLNC 通过增大编码窗口大小 w_e , 也能扩展编码包的覆盖范围. 表1对比了增大 w_e 的 CRLNC 和 IS-RLNC 这2种方案, 测试不同码率 $R(1/2, 2/3, 4/5, 8/9, 16/17)$ 的不

同平均编码时间和译码失败概率. 其中基准 CRLNC 参数为 $w_e = 32, w_d = 48$, IS-RLNC 的 $v=3$, 其他参数相同. 增大 w_e 的 CRLNC 参数为 $w_e = 96, w_d = 144$, 其他不变.

从表1中可以看出, CRLNC 通过增大 w_e , 译码失败概率与 IS-RLNC 基本一致, 但是平均编码时间明显增加. 如果上层应用程序的发送速率过快, 较高的平均编码时间将导致编码器无法及时编码. 当缓存上层数据的队列溢出时, 编码器将被迫丢弃上层传来的数据包. 相比之下, 结合交织的 IS-RLNC 的 w_e 保持不变, 能够保持较低的平均编码时间, 适配更高的上层发送速率. 同时其较低的计算复杂度, 也更加适用于性能受限的边

表 1 各编码方案的译码失败概率与平均编码时间对比

编码方案	指标	$R = 1/2$	$R = 2/3$	$R = 4/5$	$R = 8/9$	$R = 16/17$
基准 CRLNC	平均编码时间/ μs	38.84	21.30	12.33	7.92	5.78
	译码失败概率	4.70×10^{-3}	2.71×10^{-2}	4.27×10^{-2}	4.79×10^{-2}	4.84×10^{-2}
ISRLNC, $v = 3$	平均编码时间/ μs	38.59	21.43	12.62	8.19	5.93
	译码失败概率	7.20×10^{-5}	5.21×10^{-3}	2.59×10^{-2}	4.32×10^{-2}	4.74×10^{-2}
增大编码窗口的 CRLNC	平均编码时间/ μs	89.71	46.38	24.90	14.98	8.86
	译码失败概率	6.60×10^{-5}	4.22×10^{-3}	2.52×10^{-2}	4.15×10^{-2}	4.61×10^{-2}

缘设备.

4.5 最佳交织深度

图 10 的分析中发现并非交织深度 v 越高,译码失败概率越低.图 16 对不同 v 下的译码失败概率进行测试.信道丢包率 $\varepsilon = 5\%$,码率 $R = 8/9$,编码窗口大小 $w_e = 32$,解码窗口最大尺寸 $w_d = 48$.可以看出,随着 v 的提高,译码失败概率先是下降,随后会缓慢上升,表明 v 存在一个最佳值.同时在不同的突发长度 $E[B]$ 下,最佳交织深度并不一致.在低突发环境下,最佳交织深度更小.这是因为 $E[B]$ 变小,而 ε 不变的情况下,从式(5)和式(6)可知,Gilbert 信道中状态之间转换的概率 γ 和 β 都会变高,信道状态转换的会更频繁,即擦除信道的“记忆”比较短,使得上一段突发丢失的包尚未恢复,下一段突发丢失的包已经开始.此时,IS-RLNC 由于其稀疏保护特性, v 越高,在遭遇突发时需要更长的“好”状态来恢复.因此,在低突发的环境下,过高的 v 反而会降低 IS-RLNC 的恢复性能.

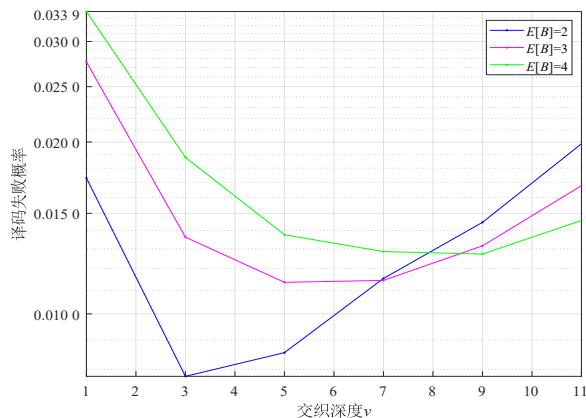


图 16 译码失败概率关于交织深度的变化

5 结论

为实现在高突发环境下,应用层 FEC 仍能保持高可靠性和低时延的优异表现,本文提出了交织 IS-RLNC 的编解码方案,通过等间隔编码以及编码包分散发送,在保证低计算复杂度的同时,能够有效应对高突发丢失,并保持良好的译码恢复能力.实验结果表明,相较于 RS 和 CRLNC 码,IS-RLNC 在恢复能力上均具有明显

优势.同时,本文还分析了不同突发长度对最佳交织深度选择的影响.未来的工作中,我们将尝试优化解码方案,结合反馈机制,引入信道丢包率预测算法来调整编码交织深度,提升解码速度和性能.

参考文献

- [1] 许辰人,马翔天,徐昊天,等.5G 抗干扰技术综述[J].电子学报,2023,51(3):765-778.
- [2] XU C R, MA X T, XU H T, et al. A survey of 5G anti-interference technology[J]. Acta Electronica Sinica, 2023, 51(3): 765-778. (in Chinese)
- [3] OSSEIRAN A, BOCCARDI F, BRAUN V, et al. Scenarios for 5G mobile and wireless communications: The vision of the METIS project[J]. IEEE Communications Magazine, 2014, 52(5): 26-35.
- [4] BENNIS M, DEBBAH M, POOR H V. Ultrareliable and low-latency wireless communication: Tail, risk, and scale[J]. Proceedings of the IEEE, 2018, 106(10): 1834-1853.
- [5] LI Z X, UUSITALO M A, SHARIATMADARI H, et al. 5G URLLC: Design challenges and system concepts[C]// 2018 15th International Symposium on Wireless Communication Systems. Piscataway: IEEE, 2018: 1-6.
- [6] 罗洪斌,张珊,王志远,等.对互联网体系结构创新的认识与思考[J].电子学报,2024,52(4):1411-1420.
- [7] LUO H B, ZHANG S, WANG Z Y, et al. Understanding and thinking about the innovation on Internet architecture[J]. Acta Electronica Sinica, 2024, 52(4): 1411-1420. (in Chinese)
- [8] 李斌,刘文帅,谢万城,等.智能超表面赋能移动边缘计算部分任务卸载策略[J].电子与信息学报,2022,44(7):2309-2316.
- [9] LI B, LIU W S, XIE W C, et al. Partial computation offloading for double-RIS assisted multi-user mobile edge computing networks[J]. Journal of Electronics & Information Technology, 2022, 44(7): 2309-2316. (in Chinese)
- [10] KARZAND M, LEITH D J, CLOUD J, et al. Design of FEC for low delay in 5G[J]. IEEE Journal on Selected Areas in Communications, 2017, 35(8): 1783-1793.
- [11] KARETSI F, PAPAPETROU E. A low complexity net-

- work-coded ARQ protocol for ultra-reliable low latency communication[C]//2021 IEEE 22nd International Symposium on a World of Wireless, Mobile and Multimedia Networks. Piscataway: IEEE, 2021: 11-20.
- [9] ROCA V, NEUMANN C, FURODET D. Low density parity check (LDPC) staircase and triangle forward error correction (FEC) schemes[EB/OL]. (2008-06-01) [2025-01-01]. <https://www.rfc-editor.org/rfc/rfc5170>.
- [10] LACAN J, ROCA V, PELTOTALO J, et al. Reed-Solomon forward error correction (FEC) schemes[EB/OL]. (2009-04-01)[2025-01-01]. <https://www.rfc-editor.org/rfc/rfc5510.html>.
- [11] BOURAS C, KANAKIS N, KOKKINOS V, et al. Evaluating RaptorQ FEC over 3GPP multicast services[C]//2012 8th International Wireless Communications and Mobile Computing Conference. Piscataway: IEEE, 2012: 257-262.
- [12] HO T, MEDARD M, KOETTER R, et al. A random linear network coding approach to multicast[J]. IEEE Transactions on Information Theory, 2006, 52(10): 4413-4430.
- [13] MAYMOUNKOV P, HARVEY N J A, LUN D S. Methods for efficient network coding[C]//44th Annual Allerton Conference on Communication, Control, and Computing. Urbana: University of Illinois, 2006: 482-491.
- [14] ROCA V, TEIBI B, BURDINAT C, et al. Block or convolutional AL-FEC codes A performance comparison for robust low-latency communications[R/OL]. (2017-01-01)[2025-01-01]. <https://inria.hal.science/hal-01395937>.
- [15] CLOUD J, LEITH D, MÉDARD M. A coded generalization of selective repeat ARQ[C]//2015 IEEE Conference on Computer Communications. Piscataway: IEEE, 2015: 2155-2163.
- [16] CLOUD J, MÉDARD M. Network coding over SATCOM: Lessons learned[M]//Wireless and Satellite Systems. Cham: Springer International Publishing, 2015: 272-285.
- [17] KARZAND M, LEITH D J. Low delay random linear coding over a stream[C]//2014 52nd Annual Allerton Conference on Communication, Control, and Computing. Piscataway: IEEE, 2015: 521-528.
- [18] WUNDERLICH S, GABRIEL F, PANDI S, et al. Caterpillar RLNC (CRLNC): A practical finite sliding window RLNC approach[J]. IEEE Access, 2017, 5: 20183-20197.
- [19] WUNDERLICH S, GABRIEL F, PANDI S, et al. We don't need no generation-a practical approach to sliding window RLNC[C]//2017 Wireless Days. Piscataway: IEEE, 2017: 218-223.
- [20] GABRIEL F, WUNDERLICH S, PANDI S, et al. Caterpillar RLNC with feedback (CRLNC-FB): Reducing delay in selective repeat ARQ through coding[J]. IEEE Access, 2018, 6: 44787-44802.
- [21] TOURNOUX P U, LOCHIN E, LACAN J, et al. On-the-fly erasure coding for real-time video applications[J]. IEEE Transactions on Multimedia, 2011, 13(4): 797-812.
- [22] KARETSI F, PAPAPETROU E. Lightweight network-coded ARQ: An approach for ultra-reliable low latency communication[J]. Computer Communications, 2022, 185: 118-129.
- [23] KARETSI F, LIASKOS C, IOANNIDIS S, et al. On the impact of coding depth in sliding window random linear network coding schemes[C]//2023 IEEE 24th International Symposium on a World of Wireless, Mobile and Multimedia Networks. Piscataway: IEEE, 2023: 205-214.
- [24] LI Y, CHEN X M, HU Y D, et al. Low-complexity streaming forward erasure correction for non-terrestrial networks[J]. IEEE Transactions on Communications, 2023, 71(12): 6870-6883.
- [25] LI Y, ZHANG F F, WANG J, et al. On streaming coding for low-latency packet transmissions over highly lossy links[J]. IEEE Communications Letters, 2020, 24(9): 1885-1889.
- [26] GABRIEL F, CHORPPATH A K, TSOKALO I, et al. Multipath communication with finite sliding window network coding for ultra-reliability and low latency[C]//2018 IEEE International Conference on Communications Workshops. Piscataway: IEEE, 2018: 1-6.
- [27] MARTINIAN E, SUNDBERG C E W. Burst erasure correction codes with low decoding delay[J]. IEEE Transactions on Information Theory, 2004, 50(10): 2494-2502.
- [28] STOLPMANN D, PETERSEN C, EICHHORN V, et al. Extending on-the-fly network coding by interleaving for avionic satellite links[C]//2018 IEEE 88th Vehicular Technology Conference. Piscataway: IEEE, 2019: 1-5.
- [29] ROCA V, CUNCHE M, THIENOT C, et al. RS + LDPC-Staircase codes for the erasure channel: Standards, usage and performance[C]//2013 IEEE 9th International Conference on Wireless and Mobile Computing, Networking and Communications. Piscataway: IEEE, 2013: 638-644.
- [30] MATSUMOTO M, NISHIMURA T. Mersenne twister: A 623-dimensionally equidistributed uniform pseudo-random number generator[J]. ACM Transactions on Modeling and Computer Simulation, 1998, 8(1): 3-30.
- [31] BIOGLIO V, GRANGETTO M, GAETA R, et al. On the fly Gaussian elimination for LT codes[J]. IEEE Commu-

nications Letters, 2009, 13(12), 953-955.

- [32] KLEINROCK L. Queuing Systems[M]. New Jersey: Wiley-Interscience, 1975.
- [33] GILBERT E N. Capacity of a burst-noise channel[J]. Bell System Technical Journal, 1960, 39(5): 1253-1265.
- [34] ELLIOTT E O. A model of the switched telephone network for data communications[J]. Bell System Technical Journal, 1965, 44(1): 89-109.
- [35] JURGELIONIS A, LAULAJAINEN J P, HIRVONEN M, et al. An empirical study of netem network emulation functionalities[C]//2011 Proceedings of 20th International Conference on Computer Communications and Networks(ICCCN). Piscataway: IEEE, 2011: 1-6.
- [36] LUTZ E, CYGAN D, DIPPOLD M, et al. The land mo-

bile satellite communication channel-recording, statistics, and channel model[J]. IEEE Transactions on Vehicular Technology, 1991, 40(2): 375-386.

- [37] BADR A, KHISTI A, TAN W T, et al. FEC for VoIP using dual-delay streaming codes[C]//2017 IEEE Conference on Computer Communications. Piscataway: IEEE, 2017: 1-9.
- [38] FRAGOULI C, SOLJANIN E. Network coding applications[J]. Foundations and Trends in Networking, 2007, 2(2): 135-269.
- [39] SCHMIDT G, SIDORENKO V R, BOSSERT M. Collaborative decoding of interleaved reed-Solomon codes and concatenated code designs[J]. IEEE Transactions on Information Theory, 2009, 55(7): 2991-3012.

作者简介



王威鹏 男, 2001年12月出生于福建省南平市. 福州大学物理与信息工程学院硕士研究生. 主要研究方向为应用层FEC编码、网络传输协议优化与无线通信.
E-mail: weipeng@163.com



李业 男, 1986年9月出生于江苏省南通市. 南通大学信息科学技术学院副教授、博士生导师. 主要研究方向为卫星通信、纠错编码. 中国电子学会会员编号: E190023555M.
E-mail: yeli@ntu.edu.cn



杜伟庆 男, 1985年10月出生于福建省泉州市. 福州大学物理与信息工程学院助理研究员. 主要研究方向为无线通信、视频编解码与传输.
E-mail: momilang1985@aliyun.com



方毅 男, 1986年8月出生于浙江省义乌市. 广东工业大学信息工程学院教授、博士生导师. 主要研究方向为信息论与信道编码、无线通信、数据存储编码. 中国电子学会会员编号: E190028682M.
E-mail: fangyi@gdut.edu.cn



陈杰 男, 1983年5月出生于福建省厦门市. 亿联网络高级项目总监, 博士、高级工程师. 主要研究方向为音视频人工智能处理技术.
E-mail: chjie@yealink.com



陈平平 男, 1986年12月出生于福建省泉州市. 福州大学物理与信息工程学院教授. 主要研究方向为信道编码、无线通信. 中国电子学会会员编号: E190021215M.
E-mail: ppchen.xm@gmail.com