组会报告

徐益

2018 年 7 月 9 日

1 工作内容

1. 更新 DPDK
2. 学习 LDPC 相关内容及编码部分代码

2 更新 DPDK18.05

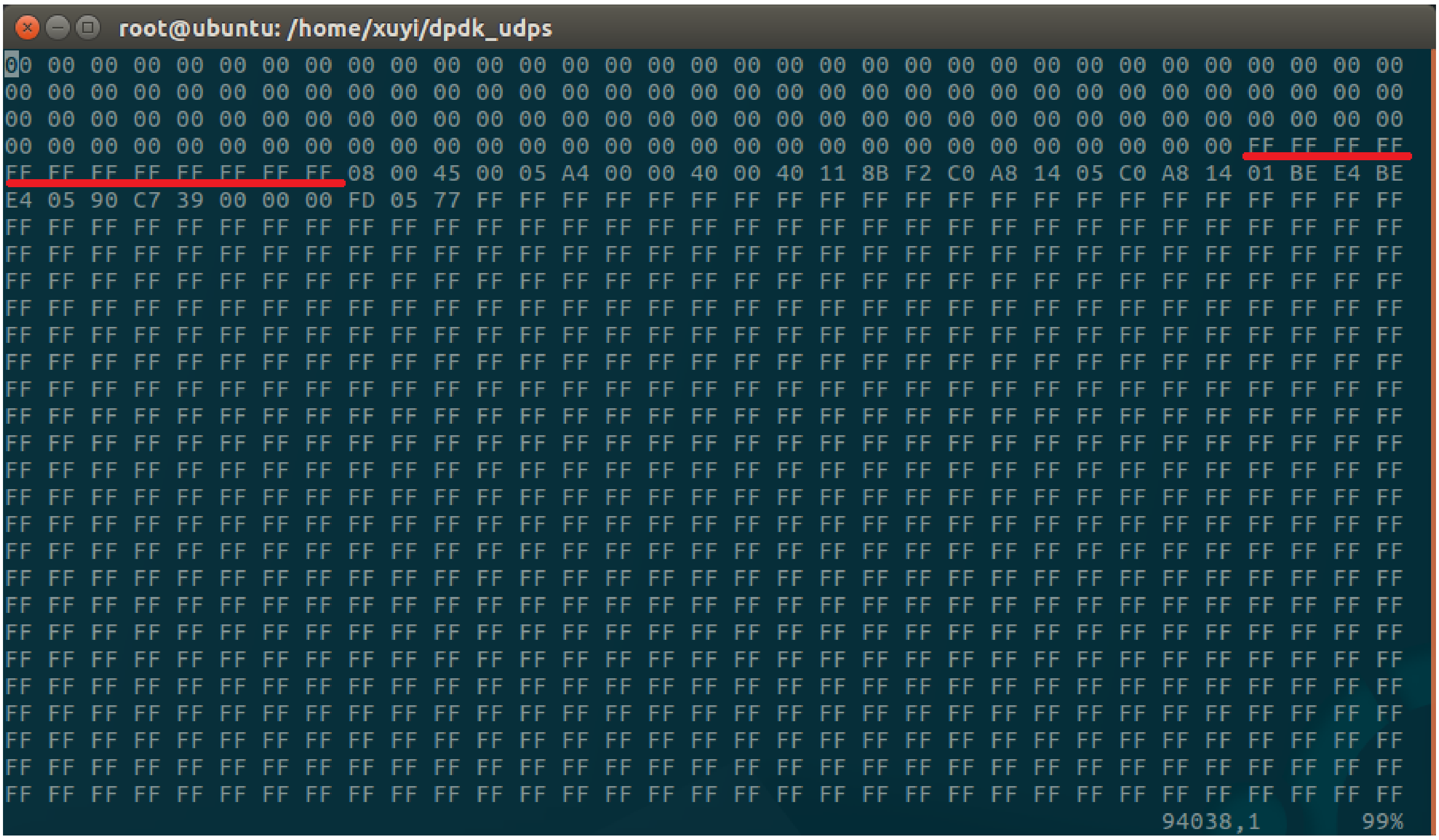


图 1: 更新后仍存在内存覆盖、丢包问题

3 LDPC 相关内容及编码部分代码学习

3.1 校验矩阵的构造

3.1.1 随机构造法

1. Gallager 构造法
2. 旋转矩阵构造法
3. PEG 构造法

3.1.2 结构化构造法

– 准循环 (Quasi-Cyclic) 构造法

A QC-LDPC code is given by the null space of an array of sparse circulants of the same size. For two positive integers c and t with $c \le t$, consider the folowing $c \times t$ array of $b \times b$ circulants over GF(2):

$$\mathbf{H}_{qc} = \begin{bmatrix} \mathbf{A}_{1,1} & \mathbf{A}_{1,2} & \cdots & \mathbf{A}_{1,t} \\ \mathbf{A}_{2,1} & \mathbf{A}_{2,2} & \cdots & \mathbf{A}_{2,t} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{A}_{c,1} & \mathbf{A}_{c,2} & \cdots & \mathbf{A}_{c,t} \end{bmatrix} \tag{1}$$

3.2 编码算法

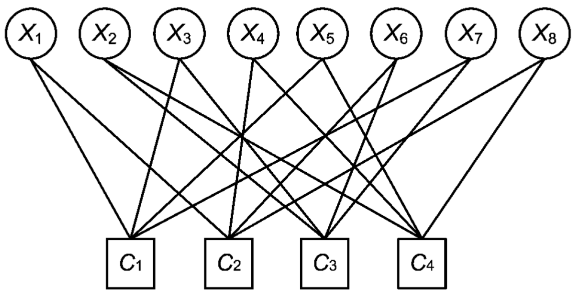
Let $\mathbf{H}_{qc} = \begin{bmatrix} \mathbf{H}_1 & \mathbf{H}_2 \end{bmatrix}$ be the partitioned base parity check matrix, where $\mathbf{H}_1$ is an $(N-M) \times M$ matrix, and $\mathbf{H}_2$ is an $(N-M) \times (N-M)$ matrix. Let $\mathbf{c} = \begin{bmatrix} \mathbf{m} & \mathbf{p} \end{bmatrix}$ be a codeword block, where $\mathbf{m}$ and $\mathbf{p}$ denote information and parity bit sequences, respectively. From the property that the correct codeword satisfies the parity check equation, the parity bit sequence $\mathbf{p}$ can be derived as follows,

$$\mathbf{H}_{qc} \cdot \mathbf{c}^{\mathrm{T}} = \mathbf{H}_1 \cdot \mathbf{m}^{\mathrm{T}} + \mathbf{H}_2 \cdot \mathbf{p}^{\mathrm{T}} = 0 \tag{2}$$

$$\mathbf{p}^{\mathrm{T}} = \mathbf{H}_2^{-1} \cdot \mathbf{H}_1 \cdot \mathbf{m}^{\mathrm{T}} \tag{3}$$

Since $\mathbf{H}_1$ is a sparse matrix, and $\mathbf{H}_2^{-1}$ has a regular pattern, the matrix-vector multiplications of (3) have linear complexity.

3.3 min-sum 译码算法



1) Initialize the iteration counter, i, to 1 and let I_M be the maximum number of iterations allowed.
2) Initialize $z_{mn}^{(0)}$ to the a posteriori LLR, $\lambda_n = \log(P(v_n = 0|y_n)/P(v_n = 1|y_n))$ for $1 \le n \le N, m \in M(n)$.
3) Update the check nodes, i.e., for $1 \le m \le M, n \in N(m)$, calculate:

$$\epsilon_{mn}^{(i)} = \min_{n' \in N(m)\backslash n} |z_{mn'}^{(i)}| \prod_{n' \in N(m)\backslash n} \mathrm{sgn}(z_{mn'}^{(i)}). \tag{4}$$

4) Update the variable nodes, i.e., for $1 \le n \le N, m \in M(n)$, calculate:

$$z_{mn}^{i} = \sum_{m' \in M(n)\backslash m} \epsilon_{m'n}^{(i)}. \tag{5}$$

5) Apply a hard decision, i.e., compute $\hat{W} = (\hat{w}_1, \hat{w}_2, ..., \hat{w}_N)$ where element $\hat{w}_n = \begin{cases} 0,\text{if}\lambda_n + \sum_{m \in M(n)} \epsilon_{mn}^{(i)} \ge 0, \\ 1,\text{otherwise.} \end{cases}$
If $\hat{W}H^T = 0$ or $i \ge I_M$, stop decoding and go to step 6. Otherwise set $i = i + 1$ and go to step 3.
6) Output $\hat{W}^{(i)}$ as the decoder output.

4　下阶段计划

1. 完成 LDPC 译码 matlab 仿真
2. 尝试 C 语言实现