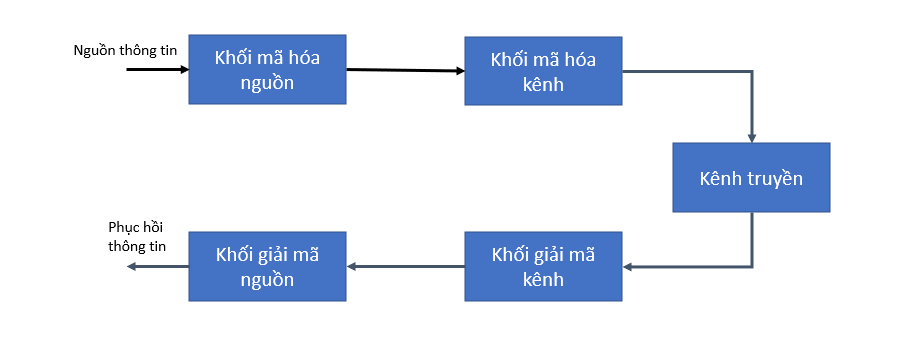
**Mã hóa kênh**: Đối với bất kỳ kênh truyền dẫn nào, tín hiệu thu sẽ bị ảnh hưởng bởi nhiễu trên đường truyền. Tùy vào độ lớn của nhiễu, có thể thu được bit tín hiệu ngược hoàn toàn với tín hiệu phát (phát bit 0, thu bit 1 và ngược lại). Để giảm hoặc loại bỏ ảnh hưởng của nhiễu trên kênh truyền dẫn, phải sử dụng một khối chức năng ở phía phát đó là bộ mã hóa kênh ở phía máy phát và bộ giải mã hóa kênh ở phía thu.



Mô hình truyền thông tin điểm – điểm

Bên cạnh truyền thông tin nguồn đi, cần truyền thêm một số bit dư thừa giúp phía thu phát hiện lỗi hoặc sửa lỗi. Trong đó:

* k: Số bit thông tin (data bits)
* n: Tổng số bit của mã codeword sau khi mã hoá
* r = n - k: Số bit kiểm tra chẵn lẻ (parity bits)
* Code rate (r= k/n): Tỷ lệ mã hóa thể hiện mức độ dư thừa
* r= 1/2 : mã có tính bảo vệ lỗi cao (nhiều bit dư)
* r= 3/4 : mã nhẹ hơn, phù hợp với kênh ít lỗi

Mã chẵn lẻ C (3,2) (chẵn; ; ) là một ví dụ đơn giản nhất của việc mã hóa kênh với tỷ lệ mã R=2/3. Cứ 2 bit thông tin nguồn được truyền đi, sẽ truyền thêm một bit dư thừa để đảm bảo rằng tổng số bit 1 trong bất kỳ từ mã nào trong họ mã C(3,2) luôn là một số chẵn như ở bảng dưới đây.

|  |  |
| --- | --- |
| Bit thông tin () | Từ mã () |
| 00 | 00 0 |
| 01 | 01 1 |
| 10 | 10 1 |
| 11 | 11 0 |

Giả sử ở bên phát truyền đi từ mã 011 và do nhiễu trên đường truyền làm đảo giá trị của bit thứ nhất từ 0 thành 1. Như vậy, máy thu sẽ nhận được từ mã 111 chứ không phải từ mã 011. Lúc này, số 1 trong từ mã thu là 3 và là một số lẻ – không đúng với quy tắc mã hóa kênh, do đó phía thu biết có lỗi xảy ra trên đường truyền từ phía phát đến phía thu.

Trong số các kỹ thuật mã hóa kênh hiện đại nhằm phát hiện và sửa lỗi hiệu quả trên kênh truyền nhiễu, mã LDPC (Low-Density Parity-Check) nổi bật như một giải pháp mạnh mẽ với khả năng tiến gần đến giới hạn Shannon.

**Mã LDPC** là một mã khối mà ma trận kiểm tra H có mật độ số 1 rất thấp. Mật độ số 1 thấp giúp giảm độ phức tạp của thuật toán giải mã vòng lặp. Khả năng sửa lỗi ký tự trong một từ mã được xác định bởi khoảng cách tối thiểu Dmin. Với Dmin là số lượng bit khác biệt nhỏ nhất giữa hai từ mã hợp lệ trong một mã khối.

Ứng dụng:

LDPC được sử dụng trong nhiều tiêu chuẩn truyền thông hiện đại như:

* IEEE 802.11n/ac/ax: WiFi (giao thức truyền mạng không dây tốc độ cao)
* IEEE 802.16e: Truyền thông băng thông rộng không dây
* DVB-S2/S2X : Truyền hình số vệ tinh, tốc độ cao, độ trễ thấp
* CCSDS (Chuẩn của NASA): Truyền dữ liệu trong không gian
* 5G NR (New Radio): Giao tiếp vô tuyến di động cho kênh dữ liệu.

Trong đó, **ma trận kiểm tra H(**parity-check matrix) đóng vai trò trung tâm, xác định cấu trúc và hiệu suất của bộ mã. Ma trận H là ma trận nhị phân có kích thước r x n dùng để kiểm tra tính hợp lệ của codeword. Nếu c ⋅ = 0 (mod 2), c là một codeword hợp lệ.

Một số dạng ma trận H phổ biến:

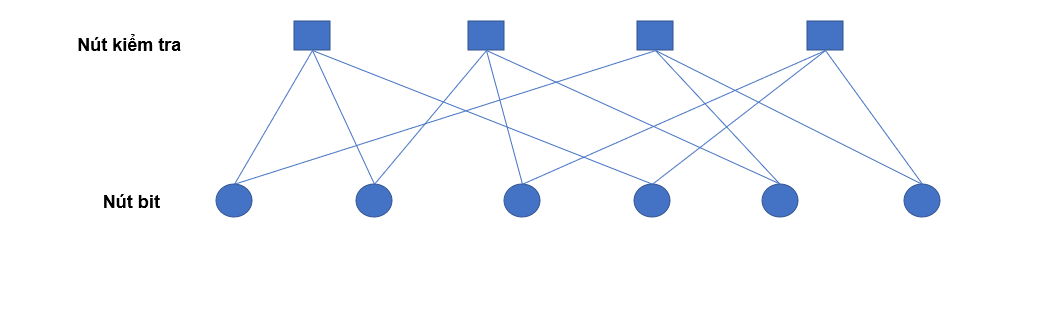
* Quasi-Cyclic LDPC: Là ma trận tuần hoàn, xây dựng từ các block xoay phải
* Protograph-based LDPC: Sinh từ đồ thị cơ sở nhỏ, sau đó nhân lên thành ma trận lớn
* Irregular LDPC: Các cột có số bit 1 không đồng đều
* Repeat-Accumulator (RA): Mã lặp kết hợp với mạch tích lũy (accumulator), mỗi bit thông tin được lặp lại nhiều lần, sau đó đưa vào mạch cộng dồn modulo-2.

Ma trận G (generator matrix) thường được suy ra từ ma trận H dưới dạng: G = [|] nếu H = [|P] (dạng hệ thống). Thoả mãn điều kiện: H ⋅ = 0. Ma trận G là ma trận sinh dùng để tạo các từ mã hợp lệ từ các thông tin gốc, có kích thước k × n.

Ma trận kiểm tra chẵn lẽ H dưới đây là một ví dụ cho 1 mã LDPC có độ dài từ mã là 6 bit (Số cột của ma trận H) và có số phương trình kiểm tra chẵn lẻ là 4 (Số hàng của ma trận H). Ngoài việc có mật độ số 1 trong ma trận chẵn lẻ ít thì mã LDPC không khác gì các khối mã tuyến tính khác.



Ma trận có kích thước (n-k) hàng tương đương với số phương trình kiểm tra chẵn lẻ (check nodes trên giản đồ Tanner) và có n cột tương đương với số bit trong từ mã (bit nodes trên giản đồ Tanner). Ma trận có thể biểu diễn bằng giản đồ Tanner như hình dưới



Giản đồ Tanner của mã LDPC

Giản đồ Tanner được sử dụng để xây dựng bộ giải mã LDPC dựa trên thuật toán lan truyền độ tin cậy - đó là thông tin mềm về các bit mã được truyền qua truyền lại giữa node bit và node kiểm tra.

Nếu từ mã thu được dưới dạng c (dạng véc-tơ hàng) được gọi là từ mã hợp lệ khi : H.=0. Trong đó 0 là véc-tơ với tất cả các phần tử bằng 0.

Như vậy, ma trận kiểm tra chẵn lẻ đóng một vai trò quan trọng trong mã hóa và giải mã LDPC.

**Mã hóa LDPC**

1. Mã hóa sử dụng ma trận sinh

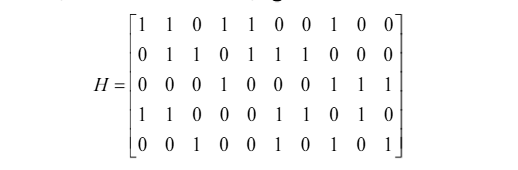
Phương pháp này đưa ma trận H về dạng:

H = [|P] với P là ma trận nhị phân (n-k) x k và là ma trận đơn vị.

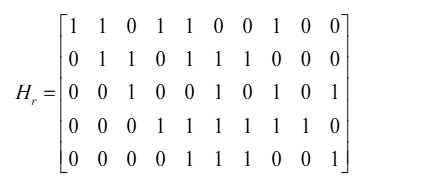
Ma trận sinh được xác định: G = [|]

Các bit thông tin được ký hiệu , , …, được viết dưới dạng véc-tơ hàng b= được mã hóa bởi một ma trận sinh (generation matrix) G để tạo ra từ mã c như sau: c= b.G. Mối quan hệ giữa ma trận sinh G và ma trận kiểm tra chẵn lẻ H là: G.=0.

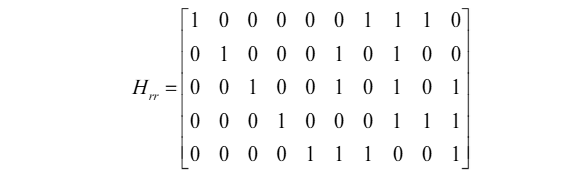
Ví dụ: Thực hiện mã LDPC với chiều dài từ mã bằng 10, tỷ lệ mã hóa ½ ma trận H có dạng



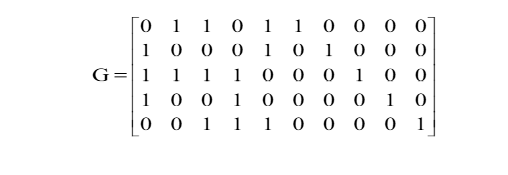
* Đưa ma trận H về dạng bậc thang hàng bằng các phép toán trên các phần tử trong trường nhị phân. Kết quả ta nhận được



* Đưa ma trận về dạng bậc thang thu gọn (reduce row-echelon)



* Từ đây suy ra ma trận sinh G

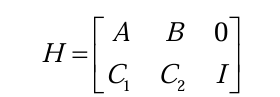


1. Mã hóa trực tiếp bằng ma trận H

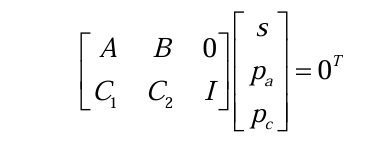
Mã LDPC có thể được mã hóa trực tiếp bằng ma trận kiểm tra chẵn lẻ H bằng cách chuyển nó thành dạng tam giác dưới bằng các phép biến đổi chỉ sử dụng hoán vị hàng và cột, để định dạng lại ma trận H thành ma trận thưa. Do đó, phương pháp này có thể giảm độ phức tạp so với phương pháp nhân ma trận sinh G.

Cho từ mã C = [s ], trong đó s biểu thị phần hệ thống, được chia thành nhóm gồm Z bits vì mô hình cơ sở có = - cột bit thông tin. Hơn nữa, s= [,…, ], trong đó mỗi phần tử s là một véc tớ có độ dài Z bits. Tổng số bit chẵn lẻ: = g+(−g). Giả sử phần chẵn lẻ của mỗi thông tin *p* được chia thành 2 phần như sau: g=5 bit chẵn lẻ đầu tiên = [,…, ] và phần còn lại gồm −g bit kiểm tra = [,…, ]. Cụ thể từ mã được biểu diễn như sau:

C = [,…, , ,…, , ,…, ]

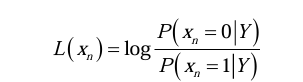
Ma trận kiểm tra chẵn lẻ H có thể được chia thành 6 ma trận và được trình bày ở dạng sau 

Trong đó: A là ma trận có kích thước g ×, B là ma trận có kích thước g × g, là ma trận có kích thước (−g )× và là ma trận có kích thước −g) ×g. Ngoài ra, I là một ma trận đơn vị có kích thước là (−g)×( −g). Việc mã hóa các mã LDPC bằng cách sử dụng phương trình sau: H.= 0 hay cũng có thể biểu thị bằng phương trình dưới:



**Giải mã LDPC**

**Sum – product** là tên chung cho 1 lớp thuật toán giải mã Maximum Likelihood (ML). Thuật toán sử dụng thông tin kênh truyền và các giá trị từ kênh truyền. Tạo ra một giá trị xác suất cho mỗi bit nhận được và làm mới giá trị này sau nhiều lần lặp để tìm ước lượng cho bit đó. Mã LDPC (N, K) là mã nhị phân được đặc trưng bởi ma trận kiểm tra chẵn lẻ thưa trong đó M = N - K có thể được biểu diễn bằng đồ hình Tanner của các nút biến n∈{1,…, N } và các nút kiểm tra m∈{1,…, M }. Thuật toán Sum – product xử lý lặp đi lặp lại các bit nhận được theo các bước nối liền nhau có thể được nhìn thấy trên đồ thị Tanner dưới dạng bước ngang tiếp theo là bước dọc để cải thiện độ tin cậy của mỗi bit được giải mã.Các thước đo độ tin cậy được tính toán của các bit ở cuối bất kỳ lần lặp giải mã nào được sử dụng làm đầu vào của lần lặp tiếp theo. Thuật toán giải mã lặp này tiếp tục cho đến khi thỏa mãn một tiêu chí dừng nào đó. Để minh họa, xét độ tin cậy của một bit đã giải mã được đo bằng xác suất posteriori P(|Y) với 1 n N. Sau đó, Log-Likelihood Ratio (LLR) của mỗi bit mã được tính bởi



Ý nghĩa:

* là bit thứ n trong chuỗi đã giải mã
* Y là tín hiệu nhận được từ kênh truyền
* LLR biểu thị độ tin cậy rằng bit là 0 hay 1, dựa trên quan sát Y
* Nếu L (>0), bit đó có nhiều khả năng là bit 0
* Nếu L (<0), bit đó có nhiều khả năng là bit 1

Syndrome H (với là chuyển vị của từ mã đã được giải mã) sau đó được tính toán và kiểm tra để quyết định giải mã thành công nếu Syndrome bằng 0 hoặc tiến hành lặp lại tiếp theo nếu điều kiện Syndrome không được thỏa mãn. Quá trình này tiếp tục cho đến khi từ mã được giải mã thành công hoặc số lần lặp tối đa đã hết.