**MÔ PHỎNG TRIỂN KHAI CDMA\_DSSS**

1. **Mục tiêu**

Trong bài lab này, chúng ta sẽ tìm hiểu và triển khai hệ thống truyền thông Direct Sequence Spread Spectrum Code Division Multiple Access (DSSS-CDMA) sử dụng Python. Bài lab tập trung vào quá trình mã hóa, trải phổ, nhúng tín hiệu vào âm thanh, và giải mã tín hiệu để khôi phục dữ liệu gốc.

1. **Yêu cầu đối với sinh viên**

Kiến thức cơ bản về hệ điều hành Linux.

Hiểu biết về ngôn ngữ lập trình Python.

Nắm bắt các khái niệm cơ bản về xử lý tín hiệu số và truyền thông.

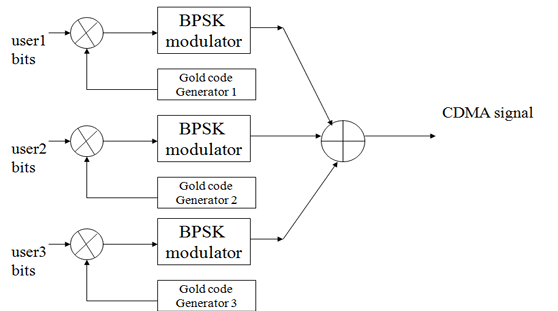
1. **Giới thiệu chung về bài thực hành**

DSSS (Trải phổ trực tiếp): Dữ liệu gốc (ví dụ: tin nhắn hoặc giọng nói) được nhân với một chuỗi mã giả ngẫu nhiên (PN code) có tốc độ cao hơn, gọi là chip. Quá trình này làm tín hiệu trải rộng trên một băng tần lớn, giúp giảm ảnh hưởng của nhiễu và tăng tính bảo mật.

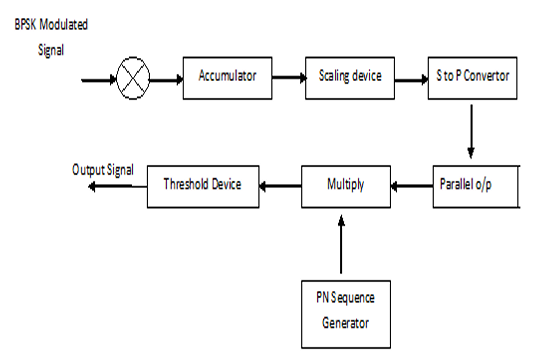
CDMA (Đa truy cập phân chia theo mã): Mỗi người dùng được gán một mã PN riêng, cho phép nhiều người dùng truyền tín hiệu đồng thời trên cùng băng tần mà không gây xung đột. Mã PN đóng vai trò như “chìa khóa” để phân biệt tín hiệu.

**Nguyên lý hoạt động:**

* **Phía truyền(Trasmitter):**



* + **Tạo tín hiệu BPSK:** Chuyển đổi dữ liệu của n người dùng thành tín hiệu BPSK (bit 1 → +1, bit 0 → -1). Dữ liệu người dùng được mã hóa thành dạng nhị phân có pha, sẵn sàng cho trải phổ.
  + **PN sequence generator**: Tạo chuỗi mã Gold cho n người dùng để trải phổ tín hiệu tương tự như tạo mã giả ngẫu nhiên. Tương ứng với khối PN Sequence Generator, sử dụng hai LFSR để tạo mã Gold dài 127 bit, đảm bảo tính phân biệt giữa các người dùng.
  + **Signal spreader**: Nhân tín hiệu BPSK với Gold code. Tín hiệu chip được tạo bằng cách nhân dữ liệu với mã PN, làm tín hiệu trải rộng trên băng tần.
  + **Embed\_audio:** Nhúng tín hiệu vào âm thanh làm sóng mang, tạo tín hiệu ẩn. Tín hiệu trải phổ được cộng vào biên độ âm thanh mà không làm thay đổi cấu trúc cơ bản.
* **Phía nhận(Receiver):**



* + **BPSK demodule**: Trích xuất tín hiệu từ file âm thanh và giải điều chế BPSK. Tín hiệu trải phổ được tách khỏi sóng mang âm thanh và chuyển thành dạng nhị phân (+1/-1).
  + **Chuyển đổi serial sang parallel**: Ghép từ Accumulator, Scaling device, S to P convertor, parallel o/p để xử lý tương quan và chuyển đổi chuỗi trải phổ từ tuần tự sang song song. Dữ liệu được gom thành vector 128 từ (độ dài mã PN + 1) để chuẩn bị cho bộ lọc phù hợp.
  + **Nhân với mã PN (Gold code):** Nhân tín hiệu parallel với Gold code để giải trải phổ
  + **Giải trải phổ tín hiệu**: Tích lũy với kết quả nhân để khôi phục tín hiệu gốc, tạo giá trị tương quan.
  + **Phát hiện và giải mã:** So sánh đầu ra với ngưỡng để phát hiện và giải mã bit dữ liệu, xác định bit 1 hoặc 0 dựa trên giá trị tương quan.

1. **Nội dung thực hành**

Dựa trên bài lab bạn cung cấp, dưới đây là giải thích chi tiết từng bước mô phỏng hệ thống DSSS-CDMA sử dụng Python, ánh xạ với các khái niệm

**Tại máy alice**

1. **Tạo tín hiệu BPSK**

**python3 generate\_bpsk.py -n 4 -f user1.txt user2.txt user3.txt user4.txt**

Max là 63 người dung, có thể tự thêm.

**Kết quả**: Tệp bpsk\_output.txt chứa tín hiệu BPSK của 4 người dùng.

1. **Tạo mã Gold**

**python3 generate\_gold\_codes.py -n 4**

**Kết quả**: Tệp gold\_codes\_output.txt chứa mã Gold.

1. **Trải phổ tín hiệu DSSS**

**python3 spread\_dsss.py -b bpsk\_output.txt -g gold\_codes\_output.txt -n 4**

**Kết quả**: Tệp spread\_output.txt chứa tín hiệu trải phổ.

1. **Nhúng tín hiệu vào âm thanh và vẽ biểu đồ**

**python3 embed\_audio\_and\_plot.py -n 4 -i input1.wav -b bpsk\_output.txt -g gold\_codes\_output.txt -s spread\_output.txt**

**Kết quả**: Tệp total\_signal\_output.txt, hình ảnh embedded\_signal.png, và tệp parameters.txt chứa thông số.

1. **Phân tích âm thanh**

**sox input1.wav output\_with\_message.wav -n stat**

**Kết quả**: Thống kê về sự khác biệt giữa hai file âm thanh.

1. **Nén và truyền file**

**tar czf file.tar.gz output\_with\_message.wav parameters.txt**

**ncat –send-only 172.10.0.11 4444 < file.tar.gz**

**Kết quả**: File file.tar.gz được gửi/nhận qua mạng.

**Tại máy bob**

**ncat -l 4444 > file.tar.gz**

**tar -xzf file.tar.gz**

1. **Trích xuất và giải điều chế**

**python3 extract\_and\_demodulate.py -p parameters.txt -i output\_with\_message.wav**

**Kết quả**: Tệp demodulated\_output.txt chứa tín hiệu trải phổ.

cat demodulated\_output.txt

1. **Chuyển đổi serial sang parallel**

**python3 serial\_to\_parallel.py -p parameters.txt -d demodulated\_output.txt**

**Kết quả**: Tệp parallel\_output.txt chứa các vector parallel.

cat parallel\_output.txt

1. **Nhân với mã PN**

**python3 multiply\_pn\_sequence.py -p parameters.txt -pr parallel\_output.txt**

**Kết quả**: Tệp multiplied\_output.txt chứa vector sau khi nhân.

cat /multiplied\_output.txt

1. **Giải trải phổ tín hiệu**

**python3 despread\_signal.py -p parameters.txt -m multiplied\_output.txt**

**Kết quả**: Tệp despreaded\_output.txt chứa tín hiệu đã giải trải phổ.

cat despreaded\_output.txt

1. **Phát hiện và giải mã**

**python3 detect\_and\_decode.py -p parameters.txt -d despreaded\_output.txt**

**Kết quả**: Tệp despreaded\_user1234.txt, decode\_user1234.txt, và hình ảnh cdma\_bpsk\_results.png.

cat despreaded\_user1234.txt

cat decode\_user1234.txt

feh cdma\_bpsk\_results.png