1. Tổng quan về mã nguồn usrsctp

**usrsctp** là một thư viện mã nguồn mở cung cấp khả năng triển khai Stream Control Transmission Protocol (SCTP) trong không gian người dùng. Các tính năng chính của usrsctp:  
- Hoạt động trong user space: Không yêu cầu thay đổi nhân hệ điều hành, phù hợp cho việc thử nghiệm và phát triển.

- Đa nền tảng: Thư viện hỗ trợ nhiều hệ điều hành như Linux, macOS và Window.

- Hỗ trợ API socket chuẩn: Cung cấp API tương tự với socket truyền thống, giúp cac nhà phát triển dễ dàng tích hợp SCTP vào ứng dụng hiện tại.

- Đa luồng và đa kết nối: Hỗ trợ nhiều luồng dữ liệu song song trong một phiên kết nối, giúp tối ưu hóa truyền tải và giảm độ trễ.

usrsctp thường được sử dụng trong các ứng dụng truyền thông yêu cầu độ tin cậy cao như:

* Truyền thông đa phương tiện thời gian thực (video/audio conferencing).
* Các dịch vụ mạng yêu cầu phục hồi nhanh và quản lý tốt luồng dữ liệu.

Thư viện này cho phép nghiên cứu, phát triển, kiểm thử và mô phỏng các tính năng của SCTP trong môi trường độc lập, đồng thời tối ưu hóa các hệ thống mạng phức tạp

1. Mục đích kiểm thử

Quá trình kiểm thử thư viện usrsctp nhằm đảm bảo tính đúng đắn, độ ổn định và hiệu suất của giao thức SCTP trong không gian người dùng. Các mục tiêu chính của việc kiểm thử bao gồm:

* Tính đúng đắn của chức năng:  
  - Kiểm tra xem các chức năng của usrsctp có hoạt động đúng như thiết kế của giao thức SCTP hay không?

- Đảm bảo rằng các tính năng chính như thiết lập kết nối, truyền tải dữ liệu qua nhiều luồng, đảm bảo thứ tự và độ tin cậy của dữ liệu được thực hiện chính xác.

* Hiệu suất: Đo lường các chỉ số hiệu suất như độ trễ, băng thông, thời gian thiết lập kết nối trong các tình huống mạng khác nhau.
* Tính ổn định:

- Kiểm thử sự ổn định của kết nối trong các tình huống tải cao và trong các môi trường có điều kiện mạng không ổn định (mất gói tin, độ trễ cao, jitter).

- Đảm bảo rằng usrsctp có khả năng phục hồi sau các lỗi mạng hoặc khi kết nối bị gián đoạn.

* Tính toàn vẹn của dữ liệu

- Xác minh rằng dữ liệu được truyền qua usrsctp không bị thay đổi hoặc mất mát trong quá trình truyền.

- Đảm bảo rằng các cơ chế kiểm soát lỗi của SCTP được thực hiện đầy đủ và chính xác.

1. Kiểm thử
2. Kết nối cơ bản

* Kịch bản kiểm thử:
* Thiết lập: Xây dựng server và client giao tiếp với nhau qua giao thức SCTP dựa trên thư viện usrsctp

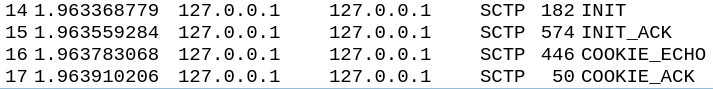
Server: Buffer có kích thước 1KB.  
Client: Buffer có kích thước 1KB.

Giao tiếp thông qua địa chỉ Loopback.

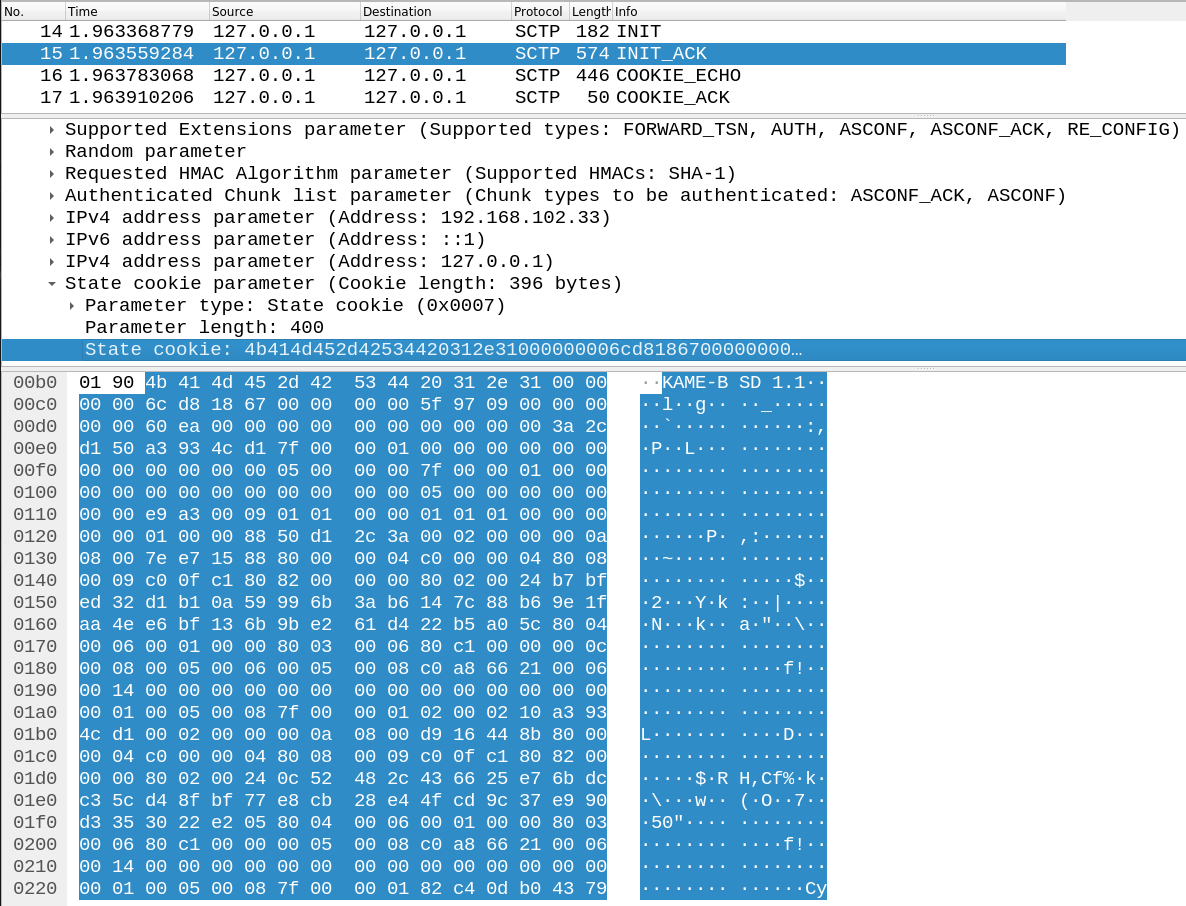
* Công cụ:   
  Wireshark: bắt các gói tin giao tiếp giữa server-client.

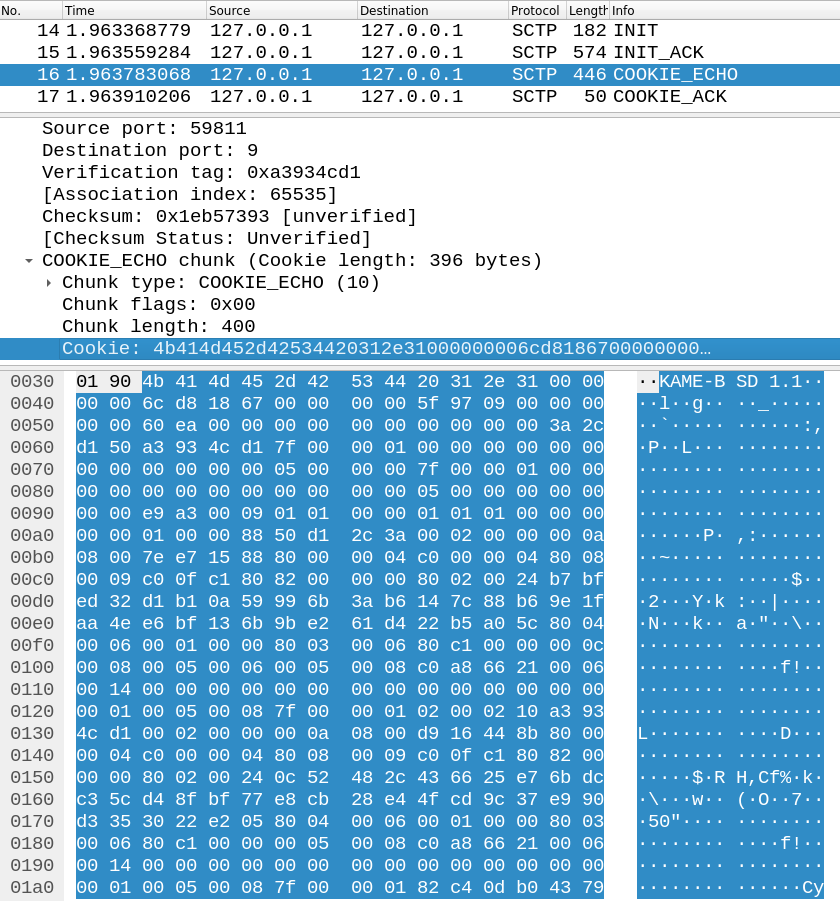
Valgrind + Massif Visualizer: đo lường bộ nhớ cấp phát.  
Pmap: đo lường bộ nhớ sử dụng của tiến trình.

* Kết quả kiểm thử:
* Server và Client đã thực hiện tạo kết nối theo đúng cơ chế bắt tay 4 bước.



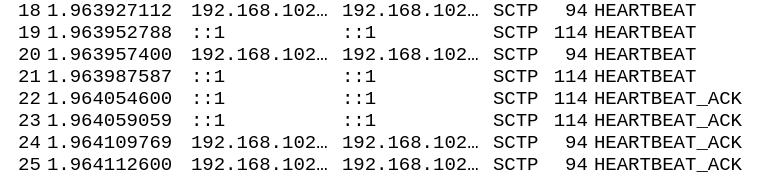
* Để tăng tính bảo mật, Server đã tạo cookie và gửi cho Client trong gói tin INT ACK; Client sao chép và gửi lại đúng cookie cho Server trong gói tin COOKIE ECHO.

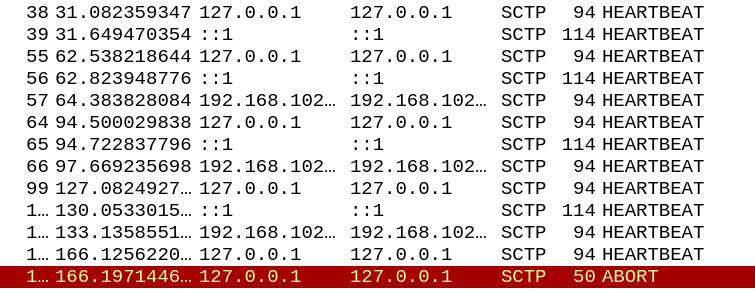




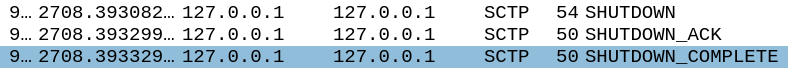
- Client gửi DATA và nhận được phản hồi SACK từ Server.



* Trong quá trình giao tiếp, Server có thể gửi các gói tin HEARTBEAT để kiểm tra trạng thái của đường truyền. Nếu không nhận ACK thì Server sẽ gửi ABORT để hủy kết nối  
  



* Quy trình đóng kết nối theo 3 bước:



* Kết quả đo lường bộ nhớ cấp phát
* Kết quả đo lường bộ nhớ tiến trình sử dụng

1. UDP encapsulation

* Kịch bản kiểm thử

- Thiết lập: Xây dựng server và client giao tiếp với nhau qua giao thức SCTP có đóng gói UDP, dựa trên thư viện usrsctp.

Server: Buffer có kích thước 1KB.  
Client: Buffer có kích thước 1KB.

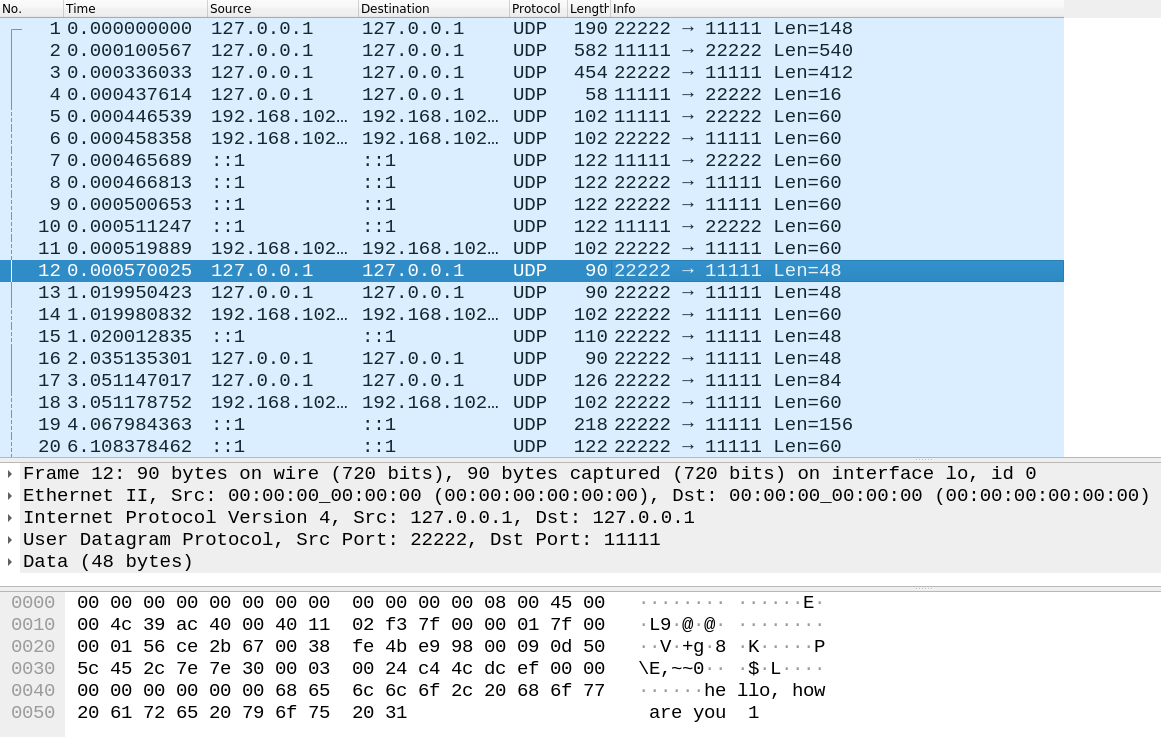
Giao tiếp thông qua địa chỉ Loopback.

- Công cụ:

Wireshark: bắt các gói tin giao tiếp giữa server-client.

Valgrind + Massif Visualizer: đo lường bộ nhớ cấp phát.  
Pmap: đo lường bộ nhớ sử dụng của tiến trình.

* Kết quả kiểm thử



1. SCTP Multi-Streaming
2. Packet Loss
3. Lỗi bit
4. Delay
5. Gói tin có kích thước lớn.