

N.E.R.F.

Network Emulator Robust-radical Freeware



Introduction

This tool works in a client-server model. The server waits for connections and the clients connect to the remote server to begin the experiment. There are two communication channels between each client and the server.

- The signaling one, which is used for information exchange and signaling between the server and the client. This communication channel uses TCP
- The experiment channel. This channel transfers data over UDP that are part of the network measurement process

Parameters

Server/Client parameters

- -a: In server mode, this argument specifies the IP address of the network interface that the program should bind. If none given the program binds in all available

interfaces. In client mode this argument provides the IP address of the server to connect to

- -p: In server mode this argument indicates the listening port of the primary communication TCP channel of the server. In client mode, the argument specifies the server port to connect to

- -i: The interval in seconds to print information for the progress of the experiment. The information displayed maybe different depending on the experiment setup. You are free to produce any kind of meaningful result.

- -f: Specifies the file that the results will be stored. This option is for your own convenience. Use an output format that will help for plotting the results.

Server parameters

- -s: The program acts like server

Client parameters

- -c: The program acts like client

- -l: UDP packet size in bytes

- -b: The bandwidth in bits per second of the data stream that the client should sent to the server. The program should take inconsideration overhead of the lower TCP/IP stack layers in order the measurement to be as accurate as possible

- -n: Number of parallel data streams that the client should create. For each one of these streams a new thread is assigned in both server and client

- -t: Experiment duration seconds The client should send for the specified amount of time and stop the datastream. Before It Exits, it informs properly the server in order the later to print the measurement results. If this argument is not specified, the data stream should be continuously sending data until a user termination signal occurs

- -d: Measure the one way delay, instead of throughput, jitter and packet loss

- -w: Wait duration in seconds before starting the data transmission

Files

MakeFile

client.h

client.cpp

Measurements.h

Measurements.cpp

Nerf.h

Nerf.cpp

NerfPacket.h

NerfPacket.cpp

Server.h

Server.cpp

Utilities.h

Utilities.cpp

Requirements

g++ compiler