

U10814

# LAB 6: Understanding Network Performance

---

Author	<b>Danny Werb</b>
Contributors	<b>George Harvey, Alexios Louridas &amp; Seb Blair</b>
Last Update	<b>2023-04-20</b>
Number of pages	<b>4</b>

---

## 1 Introduction

This lab will introduce you to *Jperf/Iperf*. You will need the following for this lab:

- *Jperf* and *Iperf* (in this exercise you will use a zip file which has both components already).
- *Wireshark* installed and ready for capture.

*Jperf* is a GUI frontend for an application called *Iperf* that measures network performance between a client and server with the ability to alter many parameters. There are actually three implementations of *Iperf*, namely the original *Iperf* which is not maintained any more and pretty much obsolete, *Iperf2* which is maintained and the successor to the previous one, and *Iperf3* which is a complete rewrite of the original and incompatible with both *Iperf* and *Iperf2*.

*Jperf* comes bundled with the original *Iperf* and should only be used in a lab setting such as this one. For critical analysis using *Iperf2* or *Iperf3* is recommended. However, *Iperf2* can largely work with *Jperf*, which means you are welcome to replace the `iperf.exe` binary bundled with *Jperf* with the newer *Iperf2* binary (see Section 5, later on).

## 2 Launching *Jperf*

Locate the file `jperf-2.0.2.zip` on the forensic network under X: Practical Files *jperf* and copy it to your Desktop. Now extract the zip archive, then locate the file `jperf.bat` (see Figure 1) and execute it.

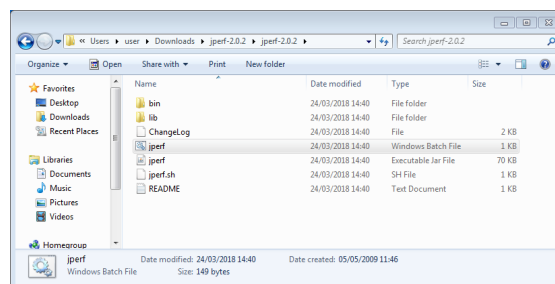


Figure 1: Launch the file `jperf.bat` to run the *Jperf* GUI.

## 3 Setup of *Jperf/Iperf* Server and Client

*Jperf* can control both client and server modes of *Iperf*. For this lab you can pair up with a classmate and one of you will run *Iperf* in server mode and the other in client mode (see Figure 3).

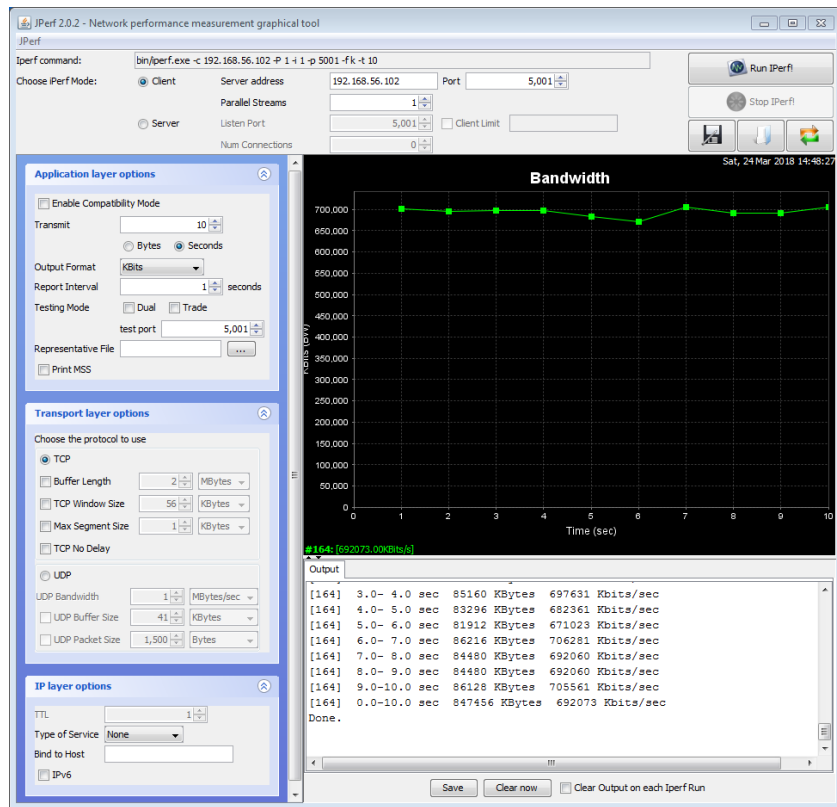


Figure 2: *Jperf* running a 10 second test to a server.

Investigate the performance of TCP/UDP by changing a single parameter at a time (taking screenshots as you go and researching each of these parameters and what they do):

- Buffer size
- Packet size
- Window size
- TCP No-delay
- Max Segmentation Size (MSS)
- Parallel streams

---

#### QUESTIONS:

- 🔗 What is the **buffer size** and how does it impact on TCP/UDP?
  
- 🔗 What is the **packet size** and how does it impact on TCP/UDP?
  
- 🔗 What is the **TCP No-Delay** option and what does it do to TCP/UDP?
  
- 🔗 What is the **window size** and how does it impact on TCP/UDP?
  
- 🔗 What is **MSS** and how does it impact on TCP/UDP?

Alter the time/amount of data being sent by the client attempt to observe any saturation effects. In groups, observe performance impact of the server when multiple clients are querying it. Attempt to create relationships between changes in different parameters and performance and derive an optimal setting for a heavy loaded server.

## 4 *Jperf/Iperf and Wireshark*

Ensure that *Wireshark* is ready to capture traffic and set up *Jperf/Iperf* to run with the settings you want in place. Now start capturing traffic and run *Jperf/Iperf* and stop the capture as soon as *Jperf/Iperf* is done with its run. Repeat this for each of the settings you want to observe.

Analyse the capture and the packets with respect to the settings you have chosen previously and note down your observations.

## 5 *Iperf2 CLI*

Copy over the folder for *Iperf2* to your Desktop from X: ▶ Practical Files ▶ iperf2 (note that the *cygwin\** files need to be in the same directory as the *iperf2.exe*). Try to use *Iperf2* in the Windows terminal shell (*cmd.exe*) and observe the options available to you as well as connecting to the *Iperf* server run by your instructor (the IP address should be provided to you).

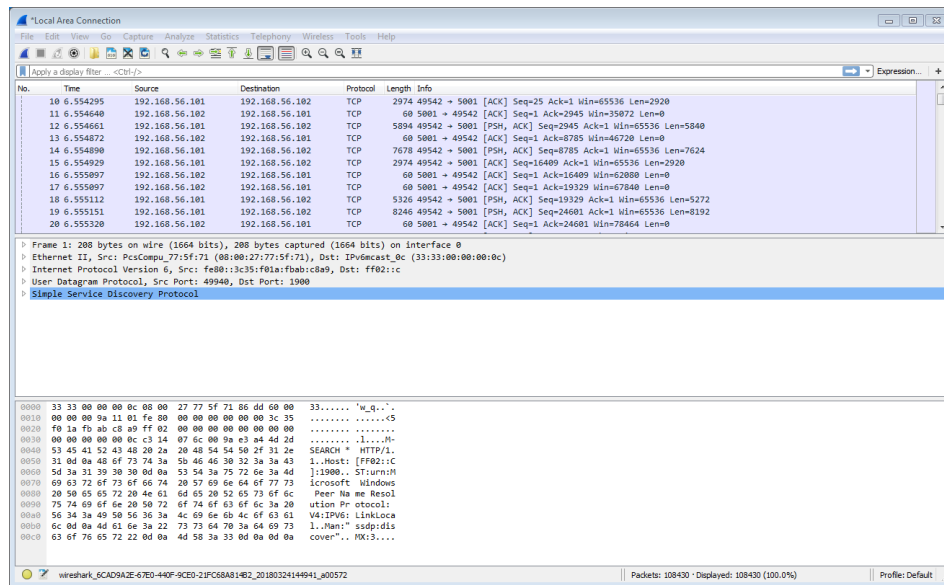


Figure 3: Wireshark capture of an Iperf test.