

Bandwidth Monitor Application

User guide

1. Compilation directions

1.1 Compilation in Eclipse IDE

Open Eclipse IDE, and import the project in your workspace:

- File -> Import
- General -> Existing Project into Workspace and click “next”.
- Select: “Select archive file” and browse the “BWMonitor_SourceCode.zip” archive.
- In the “Project” view, Select the “BandwidthMonitor” project.
- Click “Finish”.

1.2 Application execution in Eclipse IDE

In order to connect to the network devices and be able to capture packets, the application need root privilèges. So, to execute it, please follow these directions:

- Right click on the BandwidthMonitor Project, then “Export” -> “Java” -> “Runnable JAR File”, and follow the wizard.
- Execute the JAR file in a terminal using “sudo” (for macos/linux users):

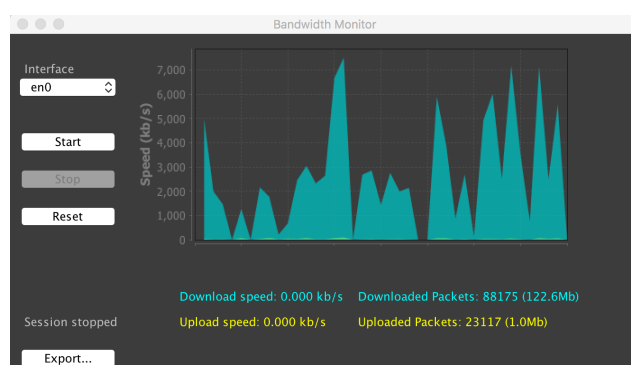
```
sudo java -jar BM_runnable.jar
```

Or in a prompt command windows, launched with “run as admin” for the Windows users.

2. Execution direction

To Monitor network activities:

- At start, the application will automatically browse for the available network interface to monitor.
- Use the dropdown list to select the device you would like to monitor.
- Press “Start” button to begin/resume the monitoring session.
- Press the “Stop” button to stop/pause the monitoring session.
- Press the “Reset” button to reset all the session variables (downloaded packets number, downloaded data size etc...).
- If you want to save all the datas recorded during the session, click on “Export” button and choose the destination folder to save the .csv file.



3. Source code explanation

3.1 Dependencies

- Jpcap:

The application is using Jpcap library to capture the network packets. Jpcap is based on libpcap (Linux & MacOS) or winpcap (Windows). libjpcap.jnilib and jpcap.jar are included with the source code, and is made for work and MacOS. If you're a linux or windows user, please read the jpcap documentation and either download the binaries or recompile the sources.

- JFreeChart:

This is the library used to display the download/upload chart. The .jar file is included in the source code. The lib also needs the jcommon lib as dependency (included).

3.1 Source code explanation

- BandwidthMonitor.java

This is the main class of the project. It uses JpcapCaptor in the discoverInterfaces() method in order to enumerate the available network interface, and create the area chart to display the download/upload speed in real time.

A timer is used to refresh each seconds the graph, fetching the network information from the PacketCatcher.java class.

- PacketCatcher.java

This class implement the "PacketReceiver" class from the Jpcap library, so it's callback method "receivePacket" will be called each time a packet is captured, after the call of the "loopPacket()" method (called in the PacketLookupThread).

- PacketLookupThread.java

This thread, launched in the main class after clicking the "Start" button, connects to the desired network device, and launch the "JpcapCaptor.loopPacket()" that initiate the packet lookup infinite loop.