

Data Usage Monitor for a Shared Wireless Network Connection

Akshay Aggarwal¹, Shantanu Kharia²

¹(Department of Information Technology, VIT University, India)

²(Department of Information Technology, VIT University, India)

Abstract: This paper proposes a novel method to monitor and control the network usage of a shared wireless connection. The main objective of this technique is to balance and restrict the shared network data. It facilitates monitor of data usage by each user individually. This technique is demonstrated through a user-friendly command line interface. The implementation is tested with Android Mobile Hotspot which exhibited satisfactory results as discussed in the paper.

Keywords: Batch Scripting, Command Line Interface, Data Usage Monitor, Limiting Data Usage, Network Profile, Windows Powershell, Windows Task Scheduler

I. Introduction

Disputes in a workplace over a shared internet connection are a common thing. Most of the disputes are due to the petty reason of unfair usage of data over a connection that is shared by multiple users. We have made an attempt to try to find a solution to the same problem. In this interface, we have used *Batch Files* (a type of script file containing commands that are processed by Windows Command Prompt.) that helped us make use of other Windows functionalities like *Task Scheduler* (component of Microsoft Windows that provides the ability to schedule the launch of programs or scripts at user defined time-intervals or triggers), and *PowerShell* (a task automation and configuration management framework from Microsoft that also helps to execute command line programs and scripts) to help us achieve our functionality of monitoring data usage over a particular network profile.

The interface allows the user to monitor their data usage and put a limit on the same. The data limit is recorded at the start of monitor period (which we have set for a month as most data network plans commercially available work on a monthly basis) and the same is recorded in a text file. The interface then runs and monitors the data usage. When the data usage surpasses the specified limit, the profile is blocked and the user isn't allowed to connect to the network profile until he modifies the limit or the monitor period is finished, as and however the user chooses to.

Let us take an example to illustrate the point. Suppose there exist an Internet source, W1 (say, a portable Wi-Fi Adapter). We take the case when the expense of data plans on the network source is being shared equally by the users in the group. Let A, B, C, D, E, F be six users in the network. We assume each user has their own device (laptop) through which they establish connection to access data services. The expense, being shared by all dictates the ethical rule that all users should have access to equal amount of data. Also, each user should be able to monitor his/her data usage over the connection so as to approximate and adjust (if needed) their data usage. We seek to provide an interface for the same where the user has the ability to change/modify monitor period, change limits of data transfer. Added to this is the functionality that once the data usage is exceeded, the connection should terminate automatically, unless mentioned otherwise. All the features mentioned above have been implemented by the authors in Windows 7 Operating System wherein no extra software(s) were needed to be installed. In-built features of Windows OS, like Task Scheduler, Command Line Interface and Powershell combined with Batch Files were used to implement the system.

II. Currently Existing Technologies

2.1 Data Usage Monitors

All Major Telecom Service Providers send a text message after every session of Data Network Usage. However, while using a Wireless System with usage spanned over multiple user devices, the data usage for individual systems is hard to calculate. This becomes even harder when using Desktops or Laptops. The Android and iOS all have the feature of calculating the data used based on a daily basis. Also, some apps on the respective platform allow data usage monitoring over custom time periods as well.

Usage Monitors for Laptops or Desktops currently in the market are fuzzy and require the user to download third-party softwares for the suite to function. Although Windows 8.1 allows users to keep a tab on the data usage, the functionality isn't backwards compatible. Also, most of the interfaces that provide the facility of monitoring data usage feature advertisements (usually push-ads) which the user often finds as irritating and/or

misleading.

2.2 Limiting Data Usage

While Android OS has the in-built feature by which it allows the user to limit the data usage, similar features are also available in iOS too. Talking about desktops and laptops, Windows 8.1 features an upgrade that allows users to limit their data usage. However, this feature is present just in Windows 8.1 and not in any previous version of Windows.

There are very less third-party softwares that allow users to put a limit on their data usage without having them battle against hefty downloads for suites and against advertisements.

Commercial Wi-Fi providers provide an interface allowing the user to monitor their data usage as well as limit it (not user-defined limit) by the means of login systems. However, we seek to provide an interface that doesn't need user to login but still allows them to monitor and limit the data usage.

III. Implementation

We have used the batch files to implement the data usage monitor and limiter. The suite consists of six batch files and as few as three text files for monitor of a Wireless Network Profile.

The following are the file names and the functionalities included in each one of them.

3.1 run_1.bat

First File of the suite, this file is run only at the first instance when the system is first used. The functionalities included in this file are- Creating a Limit (in MBs), creating a schedule based on a monthly basis on the date that is fed by user. The schedule is created using Windows Task Scheduler. The Limit Value is stored in a text file which is called as limit.txt.

3.2 run_2.bat

Second File of the suite, this file is run by another file called counter.bat when the limit set up by the user is exceeded. It allows the user to extend the limit in case he/she wants to. After modifying the limit, it unblocks the network profile that was blocked by counter.bat file so that the user can access the network again.

3.3 run_3.bat

Third File of the suite, this file is the one that is scheduled to run at the end of each monitor period (a month usually), in case the user has not exhausted the limit set by himself. This file also allows user to modify the limit for the new monitor period or select a new monitor period altogether.

3.4 Count_init.bat

Fourth file of the suite, this file is the one that logs the starting value of the data used as per system statistics and stores it in a file called Start.txt. Since only 32-bit integer data can be manipulated using Batch files, this file takes help of Windows PowerShell to do the necessary computation.

3.5 Counter.bat

This is the file that forms the mind of the limiter. This file reads the current usage value of the data and subtracts it from the value stored in Start.txt to calculate the data used. As soon as the limit is exceeded, it blocks the network profile and calls upon run_2.bat to allow the user to modify the limit in case he/she wants to. Like Count_init.bat, this file uses Windows Powershell for the computation part.

3.6 View.bat

The last file of the suite, it borrows data from Start.txt, limit.txt and system statistics and manipulates them using Windows PowerShell to tell the user the data they have used and how much data they have left until they overrun the self-imposed limit.

3.7 Start.txt

Contains the start value from which the calculations are to be done.

3.8 limit.txt

Contains limit value in Bytes that is self-imposed by the user.

3.9 Profile.txt

Contains the name of Profile the user wants to monitor.

IV. Input and Output Screens

The following are the screenshots of the screens with the inputs and the corresponding outputs-

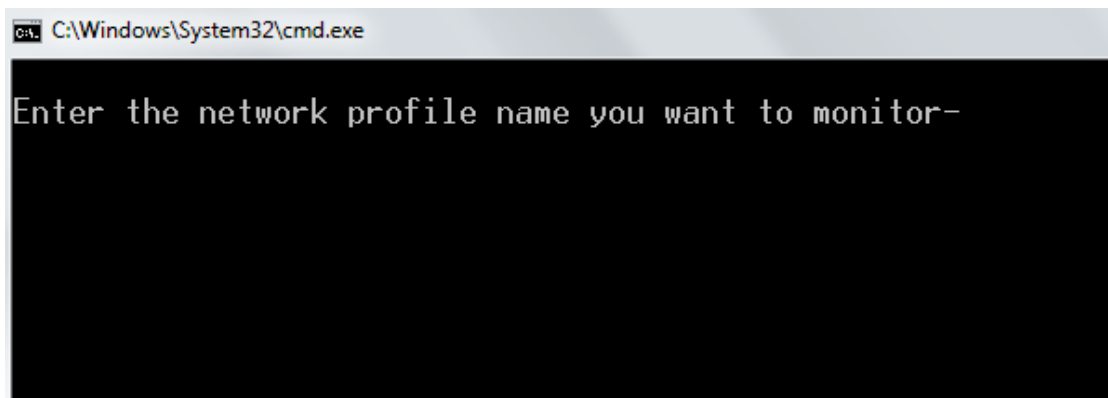


Fig. 1(a). run_1.bat. First Screen. Asks the user about the Network Profile Name the user wants to monitor using the system.

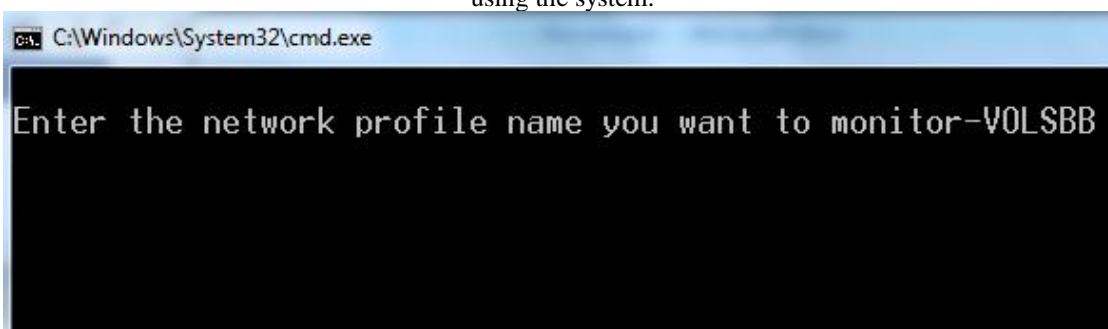


Fig. 1(b). run_1.bat. Feeding the network profile name.

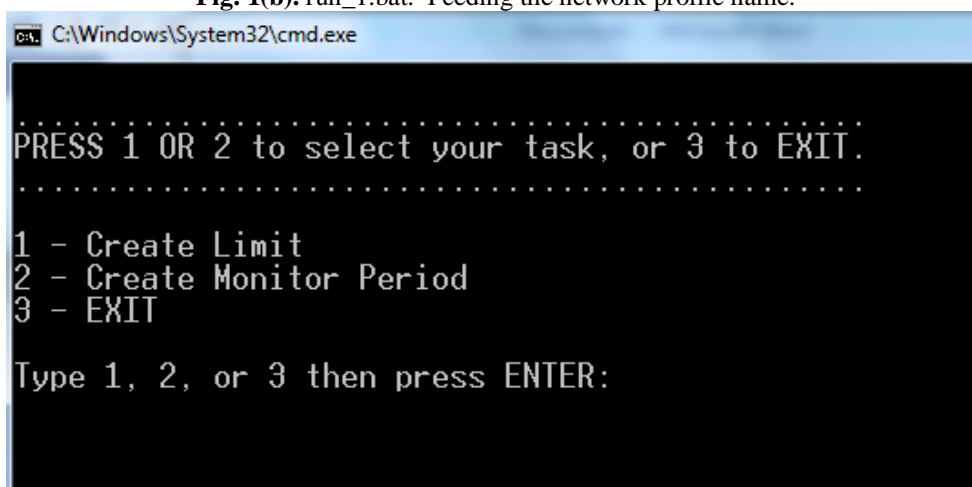


Fig. 2. run_1.bat. First Usage of Interface.

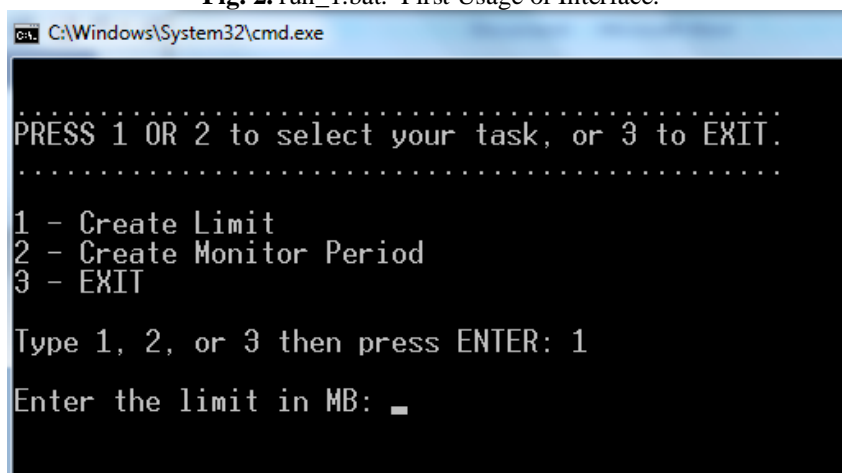


Fig. 3(a). run_1.bat. Prompting the user to create the limit.

```
C:\Windows\System32\cmd.exe

.....
PRESS 1 OR 2 to select your task, or 3 to EXIT.
.....

1 - Create Limit
2 - Create Monitor Period
3 - EXIT

Type 1, 2, or 3 then press ENTER: 1

Enter the limit in MB: 20

Limit Created/Modified

Press any key to continue
_
```

Fig. 3(b). run_1.bat. User gives input to create the limit.

```
C:\Windows\System32\cmd.exe

.....
PRESS 1 OR 2 to select your task, or 3 to EXIT.
.....

1 - Create Limit
2 - Create Monitor Period
3 - EXIT

Type 1, 2, or 3 then press ENTER: 2

Enter Date (not month or year): _
```

Fig. 4(a). run_1.bat. Creating a new schedule – asking user for date.

```
C:\Windows\System32\cmd.exe

.....
PRESS 1 OR 2 to select your task, or 3 to EXIT.
.....

1 - Create Limit
2 - Create Monitor Period
3 - EXIT

Type 1, 2, or 3 then press ENTER: 2

Enter Date (not month or year): 16
Deleting Old Schedule..
SUCCESS: The scheduled task "Revoke" was successfully deleted.
SUCCESS: The scheduled task "Logger" was successfully deleted.
Creating New Schedule..
SUCCESS: The scheduled task "Revoke" has successfully been created.
SUCCESS: The scheduled task "Logger" has successfully been created.

Limit monitor date changed to date = 16 for every month
You will be notified every month on 16

Press any key to continue
_
```

Fig. 4(b). run_1.bat. New Schedule Created. Makes Use of Task Scheduler to schedule the task for the specified date every month.

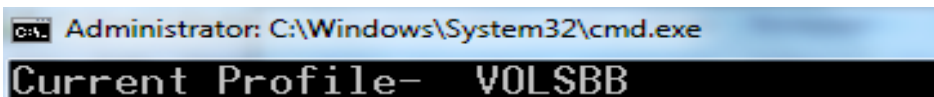


Fig. 5. Count_init.bat. Deriving the profile name and modifying the Start.txt file. This file is called by run_1.bat as soon as the schedule is created.

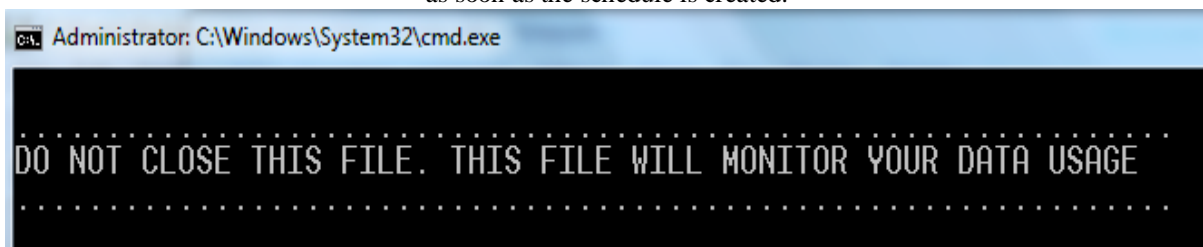


Fig. 6. counter.bat. Keeps a tab on data usage. This file shouldn't be closed.



Fig. 7(a). run_2.bat. As soon as the limit gets exceeded, counter.bat calls on run_2.bat to allow user to modify the limit. Option 2 and 3 are used to exit the program.

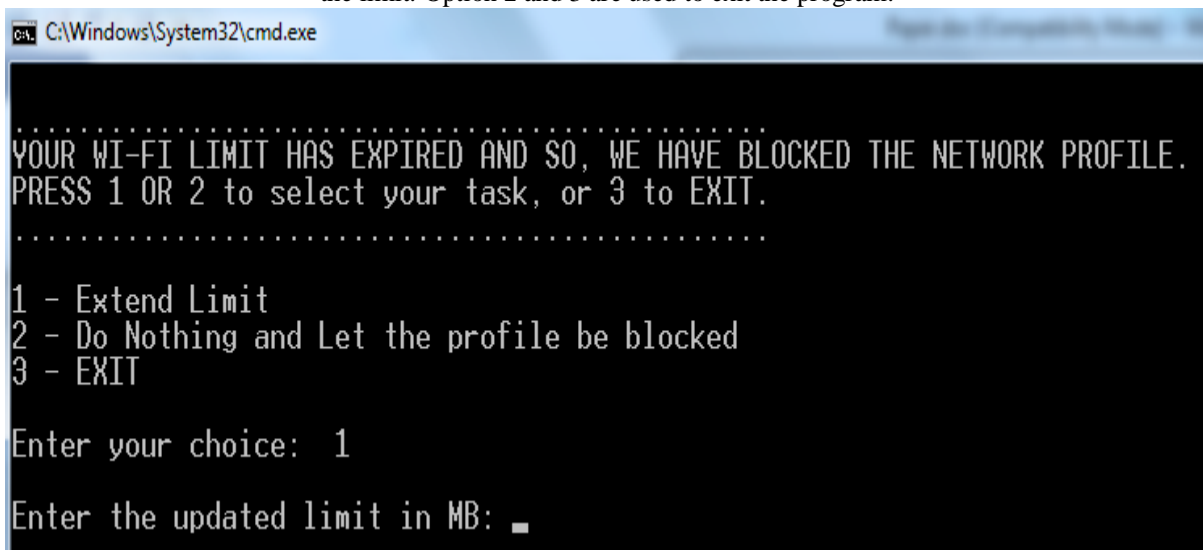


Fig. 7(b). run_2.bat. Prompt the user for modified limit.

```
C:\Windows\System32\cmd.exe

.....
YOUR WI-FI LIMIT HAS EXPIRED AND SO, WE HAVE BLOCKED THE NETWORK PROFILE.
PRESS 1 OR 2 to select your task, or 3 to EXIT.
.....

1 - Extend Limit
2 - Do Nothing and Let the profile be blocked
3 - EXIT

Enter your choice: 1

Enter the updated limit in MB: 25

Limit Modified

Unblocking network..
The filter is removed from the system successfully.
Network Profile unblocked
```

Fig. 7(c). run_2.bat. As soon as the limit is modified, the network profile is unblocked and again the counter starts.

```
C:\Windows\System32\cmd.exe

.....
PRESS 1 OR 2 to select your task, or 3 to EXIT.
.....

1 - Modify Limit
2 - Change Monitor Period
3 - EXIT

Enter your choice:
```

Fig. 8(a). run_3.bat. As soon as network profile is unblocked, run_2.bat calls run_3.bat, allowing the user to modify limit or monitor period. run_3.bat runs by itself upon completion of monitor period when the limit has not been exceeded.

```
C:\Windows\System32\cmd.exe

.....
PRESS 1 OR 2 to select your task, or 3 to EXIT.
.....

1 - Modify Limit
2 - Change Monitor Period
3 - EXIT

Enter your choice: 1

Enter the limit in MB:
```

Fig. 8(b). run_3.bat. At the end of monitor period, allow user to modify limit for next period or change the monitor period. This screenshot shows the interface prompting the user for a new limit.

```

C:\Windows\System32\cmd.exe

.....
PRESS 1 OR 2 to select your task, or 3 to EXIT.
.....

1 - Modify Limit
2 - Change Monitor Period
3 - EXIT

Enter your choice: 1

Enter the limit in MB: 30

Limit Modified

Unblocking network..
The filter does not exist on the system.
Network Profile unblocked
Press any key to continue
    
```

Fig. 8(c). run_3.bat. After modifying the limit, in case the user didn't remove the filter before, the network profile filter, if present, will be removed.

```

C:\Windows\System32\cmd.exe

.....
PRESS 1 OR 2 to select your task, or 3 to EXIT.
.....

1 - Modify Limit
2 - Change Monitor Period
3 - EXIT

Enter your choice: 2

Enter Date (not month or year): _
    
```

Fig. 8(d). run_3.bat. Prompting the user for changing the monitor period.

```

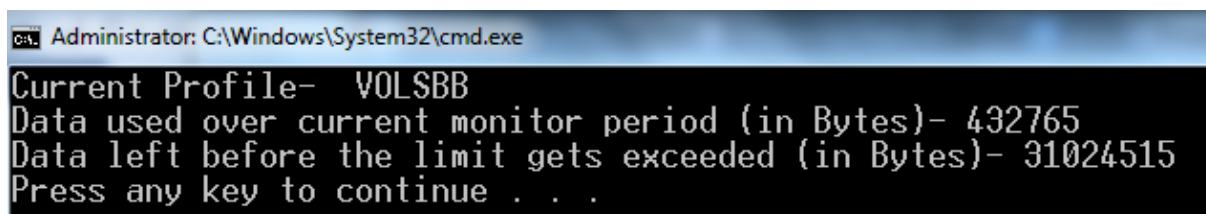
Administrator: C:\Windows\System32\cmd.exe

Current Profile- VOLSBB
Press any key to continue . . .
Deleting Old Schedule..
SUCCESS: The scheduled task "Revoke" was successfully deleted.
SUCCESS: The scheduled task "Logger" was successfully deleted.
Creating New Schedule..
SUCCESS: The scheduled task "Revoke" has successfully been created.
SUCCESS: The scheduled task "Logger" has successfully been created.

Limit monitor date changed to date = 17 for every month
You will be notified every month on 17

Press any key to continue
    
```

Fig. 8(e). run_3.bat. In case the monitor date is same as system date, it calls on Count_init.bat to record the start date.



```
Administrator: C:\Windows\System32\cmd.exe
Current Profile- VOLSBB
Data used over current monitor period (in Bytes)- 432765
Data left before the limit gets exceeded (in Bytes)- 31024515
Press any key to continue . . .
```

Fig. 9. view.bat. This file allows the user to keep a track on the data used and also know about the remaining data usage unless the limit gets exceeded.

V. Conclusion And Further Scope of the Paper

The current monitor works for a single network profile, i.e. it can be used to monitor data usage over only one connection profile at the moment. The authors are working on to extend it to monitor data usage over multiple connection profiles simultaneously.

Also, we have just handled wireless source of Internet Connection through this interface and in no manner whatsoever, touched upon a wired or a USB source like a dongle. We are trying to inculcate and merge other sources viz. wired and USB connections in addition to Wireless System to make it a full-fledged monitor system.

The current interface works on Windows OS and was developed and tested on Windows 7 Home Premium version of Windows and later versions and proved successful on each of them. The command line arguments used in the batch files have limited backward compatibility and so the interface may not work on Windows NT or even Windows Vista. However, given the limited scope of the older edition of Windows, we decided to neglect on that part.

The system uses a command line interface, which in turn requires user to get admin privileges. We seek to make the interface GUI to make it even more user-friendly.

The system is not portable across various platforms like Linux and OSX as of yet. To make the interface platform independent, we have started to develop the versions for Mac as well as Bash too. However, our primary interest lies in the expansion of current interface to be able to include other functionalities mentioned in the scope of the paper.

The system is not fool-proof yet. To be able for it to be effectively usable among a group of say 4 people, we need to introduce security in the interface so that the system itself cannot be hacked by the user and the prime functionality of limiting the data usage bypassed.

The interface saves the user from having the trouble to schedule the task monitor again and again as it is taken care of manually. Since, minimum level of human interaction is needed, the script is easy to maintain. Thus, we were successfully able to create a monitor that is

1. Semi-automated
2. Doesn't require any installation of any kind
3. Hassle-free, and
4. Effective in doing its job.

For future, our focus would be on maintaining and upgrading the system so as to include the points covered in scope of the paper.

Acknowledgements

The authors wish to thank first and foremost, our Faculty Guide, Ms. S. Menaka, Assistant Professor (SG), Operating Systems Division, School of Information Technology, VIT University, Vellore, India for being available to us all the time and for providing us support and help at all times. We also would like to thank numerous people on various forums and blogs who helped us when we were stuck while coding.