Resume sftp / scp cancelled (interrupted) network transfer - Continue (large) partially downloaded files on Linux / Windows

Author : admin

I've recentely have a t**ask to transfer some huge Application server long time stored data (about 70GB) of data after being archived between an old Linux host server and a new one to where the new Tomcat Application (Linux) server will be installed to fit the increased sites accessibility (server hardware overload)**.

The *two systems are into a a paranoid DMZ network and does not have access between each other via SSH / FTP / FTPs and even no Web Access on port (80 or SSL - 443) between the two hosts*, so in order to move the data I had to use a third HOP station Windows (server) which have a **huge SAN network attached storage of 150 TB** (as a Mapped drive **I:/**).

On the *Windows HOP station* which is giving me access via *Citrix Receiver* to the DMZ-ed network I'm using **mobaxterm** so I have the basic UNIX commands such as sftp / scp already existing on the Windows system via it.
Thus to transfer the **Chronos Tomcat application** stored files **.tar.gz** archived I've **sftp-ed** into the Linux host and used get command to retrieve it, e.g.:

```
sftp UserName@Linux-server.net
```

```
Password:
Connected to Linux-server.
sftp> get Chronos_Application_23_04_2015.tar.gz
....
```

The *Secured DMZ Network seemed to have a network shaper limiting my get / Secured SCP download to be at* ***2.5MBytes / sec***, thus the **overall file transfer seemed to require a lot of time about 08:30 hours to complete**. As it was the middle of day about *13:00* and my work day ends at *18:00* (this meant I would be able to keep the file retrieval session for a maximum of 5 hrs) and thus file transfer would cancel when I logout of the HOP station (after 18:00). However I've already left the file transfer to continue for *2hrs* and thus about *23%* of file were retrieved, thus I wondered whether **SCP / SFTP Protocol file downloads could be resumed**. I've checked thoroughfully all the options within **sftp** (interactive SCP client) and the *scp command manua*l itself however none of it doesn't have a way to do a resume option. Then I thought for a while **what I can use to continue the interrupted download** and I remembered good old **rsync (versatile remote and local file copying tool)** which I often use to create customer backup stragies has the **ability to resume partially downloaded files** I wondered whether this partially downloaded file resume could be done only if file transfer was only initiated through rsync itself and luckily ***rsync*** **is able to continue interrupted file transfers no matter what kind of HTTP / HTTPS / SCP / FTP program was used to start file retrieval**. **rsync** is *able to continue cancelled / failed transfer due to network problems or user interaction activity*), that turned even pretty easy **to continue failed file transfer download** from where it was interrupted I had to change to directory where file is located:

```
cd /path/to/interrupted_file/
```

and issue command:

```
rsync -av --partial username@Linux-server.net:/path/to/file .
```

the **--partial** option is the one that does the file resume trick, **-a** option stands for **--archive** and turns on the archive mode; equals **-rlptgoD** (no -H,-A,-X) arguments and **-v** option shows a file transfer percantage status line and an avarage estimated time for transfer to complete, an easier to remember rsync resume is like so:

```
  rsync -avP username@Linux-server.net:/path/to/file .
Password:
receiving incremental file list
chronos_application_23_04_2015.tar.gz
  4364009472   8%    2.41MB/s    5:37:34
```

To **continue a failed file upload with *rsync*** (e.g. if you used **sftp put** command and the upload transfer failed or have been cancalled:

```
  rsync -avP chronos_application_23_04_2015.tar.gz username@Linux-server.net:/path/where_to/upload
```

Of course for the rsync resume to work remote Linux system had installed **rsync** (package), if rsync was not available on remote system this would have not work, so before using this method make sure remote *Linux / Windows server* has **rsync** installed. There is an ***rsync** port also for Windows* so to **resume large Giga or Terabyte file archive downloads easily between two Windows hosts** use cwRsync.