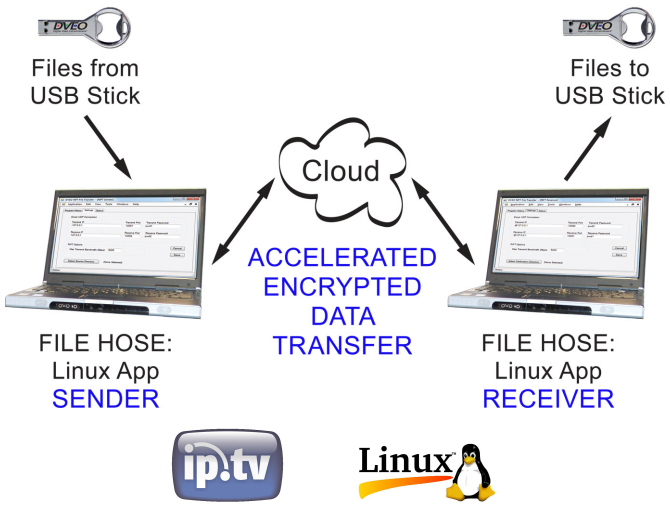
FILE HOSE™ Linux App

Linux® based high speed, highly secure file transfer application designed to bring professional quality file transfer capabilities to small to midsize users who do not wish to engage with high end service providers. This application is available to be run on sets of computers that are secured behind firewalls and have reliable internet connections. This application is available for both end users and OEMs who wish to integrate this capability into their daily data center activities.

Features

- 1000x faster than standard FTP transfer
- Our patented DOZER™ technology eliminates the bottlenecks of TCP and delivers maximum transfer speed regardless of internet conditions
- Utilizes watch folder to move files
- Automatic retry and resumes if connection is interrupted
- Creates an UDP encrypted one way tunnel and performs the file transfer through it. All content is encrypted via AES 128.
- Works with any file type or file size
- Easy to set up and operate. Drag and drop the files to the Sender folder and watch them transfer to the Receiver folder on the destination system.
- Sends one or many files at the same time
- Supports transmission of partial files. This allows starting the transmission even before the file has uploaded or creation has been completed.
- Transfer speed will not vary with network latency. Works as fast via fiber as it does on a satellite internet link, provided the bandwidths are similar.



Overview

Large file transfers between data centers or servers via the internet are essential tasks performed by all organizations with some anxiety. Most of the time transfers are successful but evidence is there that as the data sets being transferred get larger the likelihood of failure increases. Such data sets are invaluable to the organization and must be sent via completely encrypted connections between the two computers. Reliability, security, and speed are essential functions to this type of activity.

There are several large multinational firms that specialize in providing this as a service. DVEO on the other hand wishes to disrupt this market by offering the capability as an affordable application to be installed and used on any two computers that can connect via the internet. We assume that both computers are secure and reside behind corporate firewalls. If this is the case we can make the argument that equipping both computers with our software can connect and reliably and securely transfer very large data or image files between them. We also anticipate that some of these file transfers will need to be schedule driven and thus have incorporated a scheduler that will perform the file transfers on a time-driven basis.

Applications that Require AFT* with Security

- Media and Entertainment
- Government
- Financial Services
- Teleradiology
- Oil and Gas
- Publishing
- Cinema
- CDN

*AFT = Accelerated File Transfer

DVEO
Digital Video ExtraOrdinaire™
Computer Modules, Inc.
11409 West Bernardo Court
San Diego, CA 92127
Tel: 858-613-1818 Fax: 858-613-1815
www.dveo.com

How it Works

FILE HOSE: Linux App combines our own patented error correcting technology and the very fast UDP Internet data protocol to send and receive files at the fastest possible speed your Internet connection supports, with 100% accuracy. It's ideal for automatic distribution of huge media files and document archives from site to site.

Our layer 2 (‡) UDP protocol (RIFT, which stands for Rapid Internet File Transfer) has many advantages over traditional TCP based (FTP, SCP, SMB, NFS, HTTP, SSMTP) file transfer protocols which you almost undoubtedly use to transfer your files now. First of all, each UDP packet can be many times larger than a TCP packet, meaning we send fewer packets. Secondly, TCP was designed to be slow (for accuracy). Each TCP send/receive is a transaction which requires many packets to announce that something is coming, acknowledge it was received, to wait a specified time-out-period if it doesn't arrive, and to re-request when not received (remember that even a top quality NOC connection has some packets that drop). With every re-request, the TCP connection lowers the speed of all subsequent packets.

UDP has no acknowledgements built in. Our error correction protocol, to remedy this, inserts a sequence number and a checksum into every packet header. At the receiving end, we immediately detect any missing packet by a gap in the sequence, re-request it, and the transmitting side resends it without slowing down. Likewise if the packet doesn't match the checksum (meaning it was damaged during transmission), we also re-request.

You're probably familiar with the speed tests that your ISP for your home connection provides. You're also probably aware that your download speeds never quite approach the numbers those speed tests say your connection has. Those tests use UDP. Your ISP flings a lot of bytes at you, without regard to any getting dropped, then divides the total of bytes that made it by the duration of the test, and that's your speed. Our protocol does the same thing *BUT* it checks that every single byte made the journey correctly, and re-requests any missing.

If you need to move huge files from NOC to NOC, site to site, across town or across the world, this protocol will move it significantly faster, even under the worst conditions. We "torture-test" our protocol in the lab in network-emulated situations such as 4-5% dropped packets (you'd probably call 2% an Internet "bad hair" day), and maximum jitter. The proof is in the fact that our streaming protocol, using the exact same technology, is used to transport live TV events such as professional sports and breaking news around the world, to places such as New Zealand and Russia, traversing networks and submarine cable routings and re-routings where Internet transport may not be ideal.

Though FILE HOSE may sound complicated, it's ridiculously easy to configure. Once you've configured the network settings and placed it in your network, just one web page for the transmitting side, one for the receiving side. Specify a couple of IP addresses and ports, type in a passphrase or two, tell it what directory, and you'll be mirroring your huge file(s) to the other site as fast as an Internet speed test. And you can attach to your local network (SMB or NFS), so that designating a file for mirroring is as easy as dragging that huge file, using your workstation, into a designated folder on your NAS.

How it Works – Continued

In keeping with the Keep-It-Simple-Stupid nature of this single-task product, the hardware is small and dependable (we literally run live HD feeds over the same hardware in un-air-conditioned, dusty rooms for months at a time, though we expect you to put them in proper data centers; *we're supposed to torture test them!*). They run a secure Linux® operating system, including the always reliable IP tables based firewall. They'll taking a beating, such as disconnecting the power repeatedly (though again, we don't want to encourage you!); and of course if you do pull the plug, it will resume as soon as it's reconnected! File Blaster is a true single-minded, single-task product that will fulfill your need for the fastest possible transfer of huge files over the Internet.

(‡) In the OSI layer model, layer 2 is the transport layer just above physical hardware, meaning that it is the (theoretically) fastest transport possible outside of sending electrons over bare copper.

Better than FTP

Traditionally FTP is the principal protocol for file transfer. It is suitable for smaller file transfers but suffers from (1) random failures, (2) lack of strong encryption capability, and (3) lack of speed.

Over the years FTP has been enhanced several times to support PASV signaling and up to date data modes including stream modes, block modes, and compressed modes. In spite of its improvements it is hobbled by the round trip delays of sending commands and awaiting responses. FTP is also hampered by the need for a control connection which can be dropped. FTP was not designed to be a secure protocol and has many weaknesses. FTPS is an extension to the FTP standard that allows clients to request that FTP sessions be encrypted.

Software Capabilities

- Very fast UDP file transfer protocol replaces slow TCP based transfers such as FTP
- Our own patented error-correcting technology ensures 100% accuracy even under highly unreliable Internet conditions
- Based on our mature DOZER™ Layer 2 protocol used by professional broadcasters and IPTV professionals to transport video streams around the world
- Comparable to enterprise file transfer products such as the IBM® Aspera FASP protocol
- Protocol bandwidth management option provides allows sizing of the transfer pipe bandwidth (*)
- Protocol security provides AES encryption with user specified pass phrase(s) (†)
- Provides directory synchronization option such that changed files on the origin side are automatically transferred to the mirror copy
- Safe-copy; does not delete from mirror if a file is deleted from the original
- Web-GUI management
- UPTIME II failover (optional)

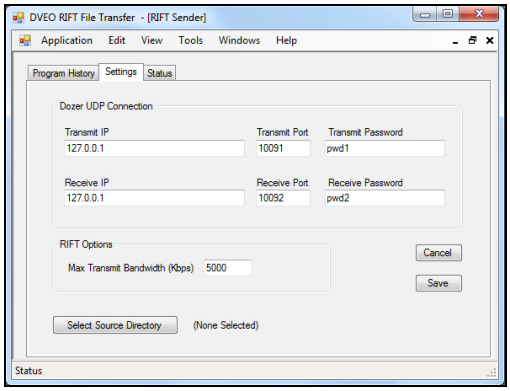
(*) The protocol was originally designed as a private line emulation protocol. In addition to specifying a precise bandwidth size, it is also possible to specify "auto" for automatic maximum-rate management based on initial ping time, ongoing transfer rate measurements and the number of re-requests necessary.
(†) Optionally specify separate passphrases for each direction for additional security.

RIFT

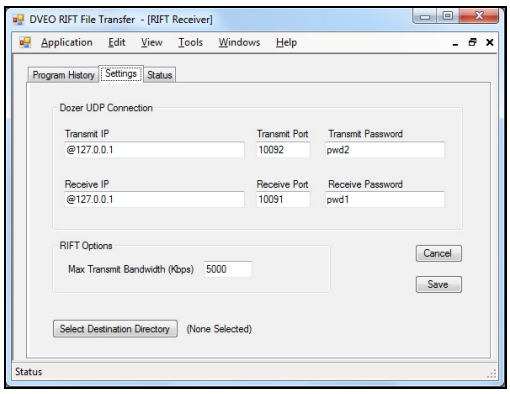
RIFT sends and receives your data (or any type of files) over UDP. But rather than just ignore any lost packets, it adds an error correction process. It sends your files in big packets, and moreover, tags each packet with a sequence number and a checksum. On the receiving side, the receiver checks that there it has every number in the sequence. If it skips from 131 to 133, it knows it has to request that the sender re-send #132, even as it continues to send additional packets – it never stops.

Additionally, the receiver checks the payload data of each packet received against the checksum. If valid, the receiver knows it received it with 100% accuracy. If not, the receiver requests the packet and the transmitter resends it, again, even as it receives additional packets. Given a short buffer time, the receiver has time to insert a resent missing or corrupt packet before writing it to disk.

Sample of GUIs



Sender GUI

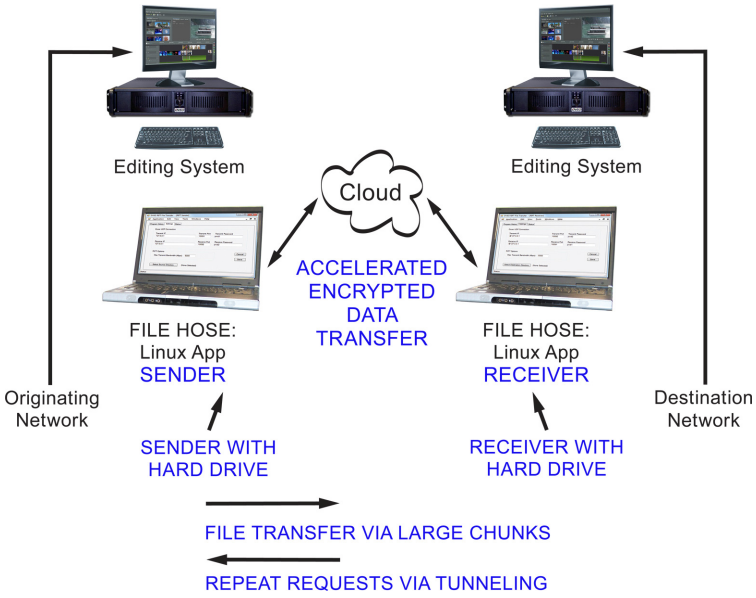


Receiver GUI

Ordering Information

FILE HOSE: Linux App
LIC: RIFT OEM License for RIFT SDK

File Transfer Acceleration



Specifications

File Input

IP Input:	Files of any type via drag and drop

IP Inputs/Outputs

IP Output Protocols:	UDP or RTP

Administration

Access:	Web interface, ssh interface, with passwords
SNMP:	Monitoring and alerts
UPTIME II™:	Failover software (Option)

System Requirements

OS:	Linux® Versions for MacOS™ or Windows® 7, 8, or 10 available

Other

Latency:	About six times the Ping Time
Bandwidth Overhead:	7% typical, but depends on network issues

DVEO
Digital Video ExtraOrdinaire™
Computer Modules, Inc.
11409 West Bernardo Court
San Diego, CA 92127
Tel: 858-613-1818 Fax: 858-613-1815
www.dveo.com