



www.internet2.edu

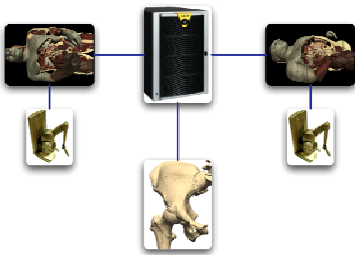
VFER is a portable, user-space tool for high-performance data transport. VFER, which enables the use of advanced congestion control without kernel modifications, uses advanced TCP-friendly congestion control that, by taking delay into account, quickly recovers from non-congestive packet loss.

Overview

VFER is a congestion-controlled, connection-based, bi-directional transport level protocol that is designed to work on top of IP (Internet Protocol). This TCP-friendly bulk transport level protocol enables high-performance data transfers without kernel changes.

Two modes of use are supported: 1) a stand-alone file transfer tool and 2) use as a library for multimedia and/or other complex applications that currently use simple UDP and could benefit from congestion control. The stand-alone file transfer tool uses normal SSH credentials that users already have. VFER is in use by the electronic-Very Long Baseline Interferometry (e-VLBI) community and the Visible Human Project (VHP):

“The University of Wisconsin - La Crosse Visualization Lab is developing the Immersive Segmentation Environment, which supports haptically-enabled segmentation and visualization of large volumetric data sets, such as the Visible Human data sets. The system is server-based and supports several simultaneous remote clients collaborating in a shared visual space. Several distinct data streams are exchanged between the server and client, with differing requirements in terms of bandwidth and reliability. Because of these characteristics, UW-L was very interested in the development of a new transport protocol such as VFER. UW-L has been happy to participate in the working group that defined the specifications for



VFER and has begun to integrate it into the Immersive Segmentation Environment.”

-- Steven Senger, University of Wisconsin-Lacrosse

History

In late 2004, Internet2's Bulk Transport Working Group (transport.internet2.edu/) began identifying specifications for a protocol that would enable high-speed bulk data transfers more effectively than TCP or UDP, would not have a negative impact on TCP traffic, and would be user-accessible and not requiring kernel modifications (see inset, next page).

The Working Group developed a list of design space requirements, published in <http://transport.internet2.edu/transport-design-space.pdf>. Steven Senger, a member of the Working Group, integrated the library and developed a draft API, which is available online, for the VHP community. The API closely parallels the Berkeley Sockets API, making application migration easy. As previously mentioned, VFER can be used as a stand-alone application for file transfer but the underlying library can be linked against by multimedia applications, amongst others.

As part of their early discussions, they examined several protocols that were under development to see if they matched all or some of the group's requirements. When none matched, completely, an unsuccessful effort was made to modify one that was close to meeting the specifications. At this point, the group realized that original code was needed.

vfer.internet2.edu



www.internet2.edu

*Internet2's **Bulk Transport Working Group** wanted no kernel modifications be required to enable fast experimentation with various congestion control frameworks. When there is no need to recompile the kernel or reboot the machine for changes to take effect, there is a faster development cycle and faster progress.*

One of the Working Group Chairs, Dr. Stanislav Shalunov, submitted the project to the Google Summer-of-Code (SoC) project list in 2005; several students worked on projects that addressed aspects of the specifications but one, Ivan Beschastnikh, was working on a project that closely met the specifications of the Working Group. Through the SoC 2005 and 2006, an Internship with Internet2 during the academic year between, and the efforts, in 2006, of three other SoC interns mentored by Internet2, VFER 0.98 has been released and is in constant use by VHP and regular use by e-VLBI.

Specifications

VFER is a connection-oriented protocol. As with TCP, a VFER connection is established with a three-way handshake that lets the endpoints advertise and exchange flow control settings. Like UDP, VFER accepts application data in discrete datagrams, which it then transmits in smaller fragments called data packets. On the receiver end, the data packets are composed and once a datagram is received in full, it may be collected by the receiving application.

VFER uses a novel path MTU discovery probing mechanism¹ to select the largest possible data packet size. This mechanism is adaptable to changing network conditions and is rerun periodically without disturbing the VFER connection, as the MTU probes payload VFER data that would have been sent anyway.

Acknowledgement packets (acks) provide for reliability in VFER and carry congestion control information used by the sender. These acks are sent by the data receiver

and each ack carries a DSACK-like set of data ranges that the receiver is missing for a datagram.

VFER uses delay-based congestion control. The receiver maintains two moving history windows of the observed relative delays of the data-packet stream: the base delay history and the current delay history. The two windows differ only in their respective lengths, which are dependent on the RTT. The receiver also maintains statistics about the relative delays these histories contain. A unique feature of VFER is that ack packets also carry a delay delta, which is the difference of the maximum observed current delay and the minimum observed base delay. This value is always greater or equal to zero and it strongly correlates with queuing delay in the network. By using this information, the sender corrects its congestion window whenever the delay delta diverges from the expected target value.

The data-sender responds to loss events just like a TCP client would – by halving the congestion window, however, for most network conditions, this is unnecessary because the delay delta stream observed by the data sender allow the sender to proactively shrink the window before losses occur.

VFER's congestion control ensures that 1) the sender does not overrun the receiver with its sending rate and 2) the protocol is TCP-friendly towards other co-existing connections in the network.

Connection termination can be initialized by any end-point at any time and is a simple closing handshake. For more information, see the current specifications on the VFER website.

¹ Internet Draft, "Packetization Layer Path MTU Discovery", Mathis, and Zekauskas.

vfer.internet2.edu