API for a Bulk Transport Tool

Internet2 Bulk Transport Working Group

Draft: transport-api-lastcall, Feb. 12, 2007

Contributors: Larry Dunn (Cisco), Yunhong Gu (UIC), Injong Rhee (NCSU), Stanislav Shalunov (Internet2),Lisong Xu (UNL), Matt Zekauskas (Internet2)
Editor: Steven Senger (UW-L)
Notes: This is the last call for comment draft.

---

vfer_fd
vfer_socket(int socktype)

Return a new socket identifier of type *socktype,* which is SOCK_STREAM or SOCK_DGRAM. If the socket creation fails the error status can be obtained through vfer_sockerror().

Returns

>= 0	success
VFER_BADTYPE	invalid socktype
VFER_IMPL	underlying socket descriptor failure

---

int
vfer_close((vfer_fd vfd)

Closes the socket *vfd*. If the socket is currently marked with an error this call returns immediately without changing the socket error status.

Returns

0	success
VFER_BADSOCK	bad vfd argument
VFER_IMPL	failure closing underlying socket descriptor

---

int
vfer_connect(vfer_fd vfd, const struct sockaddr *addr, int len)

For a socket *vfd* created as either SOCK_STREAM and SOCK_DGRAM this call attempts a connection operation with the peer specified by *addr* and *len*. This call returns immediately without changing the error status of *vfd* if *vfd* is marked with an error on entry. When *vfd* identifies a nonblocking socket this call return VFER_INPROGRESS and returns immediately with the connection operation performed in the background.

Returns

0	success

VFER_BADSOCK	bad vfd argument
VFER_TIMEOUT	connection attempt timedout
VFER_REFUSED	connection attempt refused
VFER_NOCONN	socket is not able to perform a connect
VFER_INPROGRESS	success return for nonblocking socket

---

int
vfer_bind(vfer_fd vfd, const struct sockaddr *addr, int len)

Associate the socket *vfd* with the address specified *addr* and *len*. This call returns immediately without changing the error status of *vfd* if *vfd* is marked with an error on entry.

Returns

0	success
VFER_BADSOCK	bad vfd argument
VFER_NOBIND	socket is not able to perform a bind because it is bound
VFER_BADADDR	address is unavailable or in use

---

int
vfer_listen(vfer_fd vfd, int backlog)

Configures socket vfd to listen for incoming connection requests. This call returns immediately without changing the error status of vfd if vfd is marked with an error on entry.

Returns

0	success
VFER_BADSOCK	bad vfd argument
VFER_NOLISTEN	socket is unable to listen

---

vfer_fd
vfer_accept(vfer_fd vfd)

Wait for a connection on *vfd*. On success eturns a socket identifier for use with the connection. This call returns immediately without changing the error status of *vfd* if *vfd* is marked with an error on entry.
Errors

>= 0	success
VFER_BADSOCK	bad X_SOCKET
VFER_TIMEOUT	accept operation timedout
VFER_NOACCEPT	socket is not listening
VFER_NOTSTREAM	socket is not of type SOCK_STREAM
VFER_WOULDBLOCK	socket is non-blocking and there are no waiting connections
VFER_IMPL	underlying socket error

---

int
vfer_setsockopt(vfer_fd vfd, int optname, void *optval, int optlen)

Uses the data described by *optval* and *optlen* to set *optname* on socket *vfd*. If, on entry, *skt* is marked with an error, this call immediately returns without changing the error status. The possible option names are

<u>SOCK_STREAM</u>

<u>SOCK_DGRAM</u>

QTTL	int	The duration (in usec) that unacknowledged messages will remain available for retransmission. A value of 0 gives unreliable service. An int maximum value gives reliable service. The default is zero.

<u>Both</u>

SNDSIZE	int	The packet size used for the underlying transport. This defaults to the MTU.
NOPMTUD	int	Disable PMTUD.
NONBLOCK	int	Set socket to use non-blocking mode for send, recv, accept and connect calls.

Returns

0	Success
VFER_BADSOCK	bad vfd argument
VFER_BADOPT	unknown or illegal option/length

---

int
vfer_getsockopt(vfer_fd vfd, int optname, void *optval, int *optlen)

Returns, in *optval* and *optlen*, the data describing *optname* on socket *vfd*. On entry *optlen* is the size of space pointed to by *optval*. On exit, *optlen* is changed to reflect the actual size of the returned data. If, on entry, *vfd* is marked with an error, this call immediately returns without changing the error status. In addition to the option names valid for the vfer_setsockopt() call the following additional options are available.

<u>SOCK_STREAM</u>

<u>SOCK_DGRAM</u>

MTU	int	The current MTU.
MSGSIZE	int	The maximum size of an individual message.

Returns

0	success
VFER_BADSOCK	bad vfd arguemtn
VFER_BADOPT	unknown or illegal option/length

---

vfer_stats

vfer_sockstats(vfer_fd vfd)

Compute and return current performance data for socket *vfd*. If, on entry, *vfd* is invalid or marked with an error this call immediately returns vfer_stats structure filled with zeros.

This structure contains average performance statistics for the socket and incremental statistics since the last call to vfer_sockstats(). Performance statistics include the following.

- send rate in Mbps
- recv rate in Mbps
- retransmit traffic as a fraction of total traffic
- round trip time

For a SOCK_DGRAM socket, configured for partial reliability, the performance statistics also include the number of messages that expire after exceeding the QTTL limit.

---

int
vfer_sockerror(vfer_fd vfd)

Returns the error condition of socket *vfd*.

---

char *
vfer_errortext(int err)

Returns a text description of the error code *err*.

---

int
vfer_sendfile(vfer_fd vfd, int fd, off_t offset, size_t size)

Mmap the file described by *fd* and send *size* bytes of data starting at *offset*. The send occurs with the currently reliability setting. Consequently, the call is intended for use with full reliability.

Returns

>= 0	number of bytes sent on success
VFER_BADSOCK	bad vfd argument
VFER_BADFD	error using mmap on fd
VFER_INVAL	invalid offset or size

---

size_t
vfer_send(vfer_fd vfd, const void * buf, size_t len)

Send *len* bytes from *buf* on socket *vfd*.

If *vfd* is of type SOCK_DGRAM then the data is treated as a single message and *len* must be less then the maximum message length. Messages greater than SNDSIZE are fragmented. On a fully reliable socket messages will be delivered intact and in the original order. On an unreliable socket messages are either delivered intact in the original order or discarded. If the socket is configured for partial reliability, messages will be retransmitted until no longer available in which case the message will be discarded.

Returns

>= 0	number of bytes sent on success
VFER_BADSOCK	bad vfer argument
VFER_TIMEOUT	timed out while sending data
VFER_UNCONN	socket is not connected
VFER_INVAL	invalid buf or len
VFER_WOULDBLOCK	socket is non-blocking and requested operation would block

---

size_t
vfer_recvfile(vfer_fd vfd, int fd, off_t offset, size_t size)

Reads *size* bytes from socket *vfd* writing them into the file described by *fd* at *offset*.

Returns

>= 0	number of bytes received on success
VFER_TIMEOUT	timed out while waiting to receive file
VFER_BADSOCK	bad vfd argument
VFER_BADFD	error using mmap on fd
VFER_INVAL	invalid offset or size

---

int
vfer_recv(vfer_fd vfd, void *buf, size_t len)

Read up to *len* bytes from socket *skt* into *buf*. If *skt* is of type SOCK_DGRAM then an entire message is read into *buf* with bytes in excess of *len* being discarded.

Returns

>= 0	number of bytes received on success
VFER_TIMEOUT	timed out while waiting to receive data
VFER_BADSOCK	bad X_SOCKET
VFER_UNCONN	skt is not connected
VFER_INVAL	invalid buf or len
VFER_WOULDBLOCK	socket is non-blocking and requested operation would block

---

int
vfer_selectmark(vfer_fd vfd, int mark)

Called prior to using vfer_select(), this function marks *vfd* with the conditions to be tested by the select call. *Mark* is a bitwise OR of VFER_READABLE, VFER_WRITABLE and VFER_EXCEPTION.

Returns

0	success
VFER_BADSOCK	bad vfd argument
VFER_INVAL	invalid mark value

---

int
vfer_selecttest(vfer_fd vfd)

Returns the result of the vfer_select() call for *skt*. The return value will be a bitwise OR of VFER_READABLE, VFER_WRITABLE and VFER_EXCEPTION depending on how *skt* was marked and the result of the vfer_select() call.

Returns

Bitwise OR	success
VFER_BADSOCK	bad vfd argument

---

int
vfer_select(int len, vfer_fd *vfds, struct timeval *timeout)

Returns the number of sockets in the array *vfds* of length *len* that satisfy the marked conditions. *Timeout* is the maximum time to wait before the call returns. If *timeout* is null the call blocks indefinitely. A *timeout* value of zero can be used to effect a poll operation. *Timeout* is not changed by the call.

Returns

>= 0	number of sockets satisfying marked conditions
VFER_BADSOCK	one of the socket identifiers in vfds is bad
VFER_IMPL	underlying select error
VFER_INVAL	timeout value is invalid
