

## Re: detect bytes written on abort

---

*Source:* <http://coding.derkeiler.com/Archive/PHP/comp.lang.php/2006-01/msg00351.html>

---

- *From:* Shailesh Humbad <[humbads@xxxxxxxxxxxxxxxxx](mailto:humbads@xxxxxxxxxxxxxxxxx)>
  - *Date:* Fri, 06 Jan 2006 00:26:49 -0500
- 

Gordon Burditt wrote:

TCP is a reliable transport, meaning that at the application layer, one always know exactly how much data the client received, and this is always equal to how much was successfully sent.

The above is *\*NOT\** a conventional definition of "reliable transport". And it's not what TCP tries to implement.

Stdio buffering put on a "reliable transport" as you define it above makes it unreliable, as a successful `fwrite()` on a socket may simply mean that the data has been placed in a buffer on the sender, not even passed to the OS yet. You also don't know how much data is buffered by Apache or web proxies. You don't know that the other end of the TCP connection is on the user's browser.

In a scenario where the communication channel is going to be cut at some point in time (corresponding to, say, a modem dropping carrier or network connectivity otherwise going down and staying down), and no further message traffic is possible, it is impossible to implement a protocol where the sender and receiver always agree exactly on the number of bytes received. If you send a packet and get no answer, you don't know whether the sent packet got lost or the acknowledgement got lost. You can get the uncertainty down to one byte by sending single-byte packets all the time. Slow. Wasteful of bandwidth. Even the Theory of Relativity is relevant here. The Speed of Data, as well as the Speed of Light, is finite and does not permit instantaneous communication of information.

I don't care how many bytes were transferred by TCP in the data link layer.

I don't want to restart sending the file. I also do

## Re: detect bytes written on abort

not care why the script aborted.

You DO care if the \*client\* aborted. Just because the browser got the data from TCP doesn't mean it was safely saved to disk before someone tripped over the power cord.

Of course, it must be from a network/client abort, not a server reboot or such, because the script must finish executing. I only want to be able to track how many bytes were sent to the client, which equals the value that is eventually written to the server log file.

The reason I need it is because in this system, I want to be able to show the user how many bytes the server sent them. This will tell them how much data transfer they have used.

Why would the user care? Unless you're billing them against a quota or something, which is quite a different problem from being able to restart a file transfer.

I need the status of bytes sent as soon as possible after the script completes or aborts. Thanks.

It won't happen reliably. You might get something accurate enough for \*quotas\*, but not for restarting file transfers. The way things like FTP do this is get the size of the partially-transferred file on the client side and start from there.

Gordon L. Burditt

I don't care how much data the client actually saved, only how much was transferred. Yes, my eventual aim is to bill against a quota. To solve the file restart problem, I can implement HTTP range handling later.

"Reliable Delivery - Once a connection has been established, TCP guarantees that data is delivered in exactly the same order it was sent, with no loss, and no duplication. If a failure prevents reliable delivery, the sender is informed.", Internetworking with TCP/IP Vol. III, p. 103

.