

Who Is at Fault for Poor Application Performance?



Lori MacVittie, 2008-30-04

I didn't say it was your fault, I said I was going to blame you.



When the issue of application performance rears its ugly head like some kind of ancient dragon hell-bent on destruction (yours) it is often the application developer that ends up shouldering the blame.

It's also often the case that neither the network admin or the developer can do anything to banish the evil dragon of poor performance. That's because sometimes the fault lies somewhere between the network and the application, in the murky middle layers of the OSI stack - *above* the network layers.

And that's exactly why the developer is often left standing alone to deal with the problem, as if he's the poor village sot who drew the shortest straw in the quest to stop the dragon.

But because the problems are often below the application layer, there isn't really much a developer can do. It isn't the code, it isn't the web or application server, it isn't even necessarily the operating system. In fact, it's probably the case that everything is working *exactly as it should* be despite the angry cries coming from users who believe they could write out orders by hand and mail them in faster than the application they're forced to use.

TCP, as it's defined, is not a very efficient protocol. It was designed to provide **reliable** delivery of messages and is therefore quite insistent that it do so, regardless of the potential impact on performance. Using a variety of techniques including window sizing and a (some would say excessively) thorough acknowledge protocol, TCP ensures every packet is received. Therefore it can be performing exactly to specifications and still impede application performance.

And there's nothing the developer can really do to change that. It can't be changed by the code, or by web/application server settings, and there are only a few operating system level modifications that can be made to improve the behavior of the TCP/IP stack. And there's also not really anything in the routing and switching infrastructure that can improve the situation. Somehow TCP has to be made to be more efficient, and that's not something that either group can accomplish without some assistance.

This is where application delivery solutions can help. Because an intelligent application delivery controller sits in the middle, a proxy between the client and the server, it can "muck" with TCP and make it more efficient. For example, [F5 BIG-IP](#) implements a highly optimized TCP/IP stack called [TCPEXpress](#). TCP Express implements (and improves in some cases) many of the performance enhancing RFCs (Request For Comments) related to TCP such as:

- Delayed Acknowledgements, Nagle Algorithm – (RFC [896](#), [1122](#))
- Selective Acknowledgements – (RFC [2018](#), RFC [2883](#))
- Explicit Congestion Notification ECN – (RFC [3168](#), [2481](#))
- Limited and Fast Retransmits – (RFC [3042](#), RFC [2582](#))
- Adaptive Initial Congestion Windows – (RFC [3390](#))
- TimeStamps – (RFC [1323](#))
- Improve TCP TIME-WAIT Assassination Hazards – (RFC [1337](#))
- Improve TCP Congestion Management – (RFC [3168](#))
- Improve TCP Slow-Start for TCP with Large Congestion Windows – (RFC [3742](#))
- Appropriate Byte Counting – (RFC [3465](#))
- Improve TCP Fast Recovery Algorithm (NewReno) – (RFC [3782](#))

Each of these implementations is designed to improve the performance of TCP/IP and thus improves the performance of TCP/IP based applications (99% of all web applications fall into this category). These performance-related enhancements improve the situation on both the client and server side of the equation, because a full-proxy solution mediates absolutely between the two, giving it more control over the behavior of TCP/IP.

There's no code to change in your application, no modifications to router and switch configurations, nothing. A full-proxy application delivery solution can transparently improve the efficiency of TCP/IP and immediately provide performance benefits to all managed TCP/IP based applications.

So the next time someone is blaming you - whether it's your fault or not - for poor application performance, consider whether the problem might really lie within the realm of "things not under our control". And if that's the case, remember that you *can* get it under control by adding an application delivery solution to your infrastructure.

Imbibing: Coffee

Technorati tags: [MacVittie](#), [F5](#), [TCP](#), [application performance](#), [application acceleration](#), [developers](#)

F5 Networks, Inc. | 401 Elliot Avenue West, Seattle, WA 98119 | 888-882-4447 | [f5.com](#)

F5 Networks, Inc.
Corporate Headquarters
info@f5.com

F5 Networks
Asia-Pacific
apacinfo@f5.com

F5 Networks Ltd.
Europe/Middle-East/Africa
emeainfo@f5.com

F5 Networks
Japan K.K.
f5j-info@f5.com