What's the Difference Between TCP and UDP?

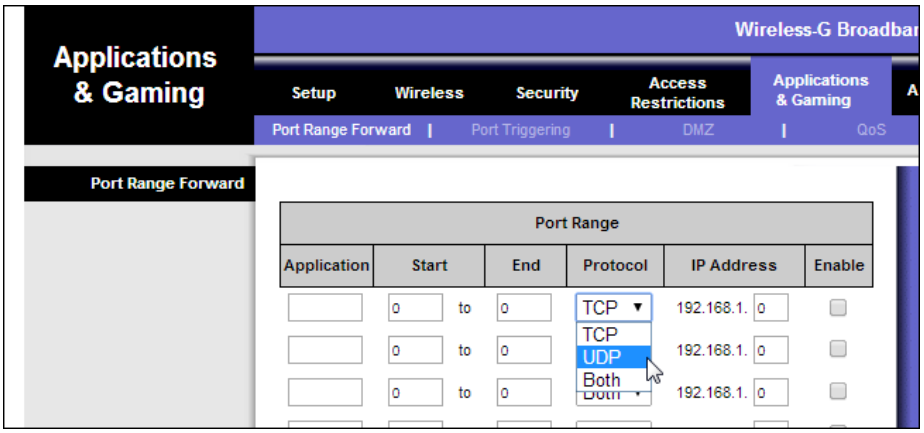
You've probably seen references to TCP and UDP when setting up port-forwarding on a router or when configuring firewall software. These two protocols are used for different types of data.

Both of these protocols are used to send data over the Internet or a local network. When configuring some network hardware or software, you may need to know the difference.

What They Have In Common

Both TCP and UDP are protocols used for sending bits of data — known as packets — over the Internet. They both build on top of the Internet protocol. In other words, whether you're sending a packet via TCP or UDP, that packet is sent to an IP address. These packets are treated similarly, as they're forwarded from your computer to intermediary routers and on to the destination.

TCP and UDP aren't the only protocols that work on top of IP. However, they are the most widely used. The widely used term "TCP/IP" refers to TCP over IP. UDP over IP could just as well be referred to as "UDP/IP", although this isn't a common term.



How TCP Works

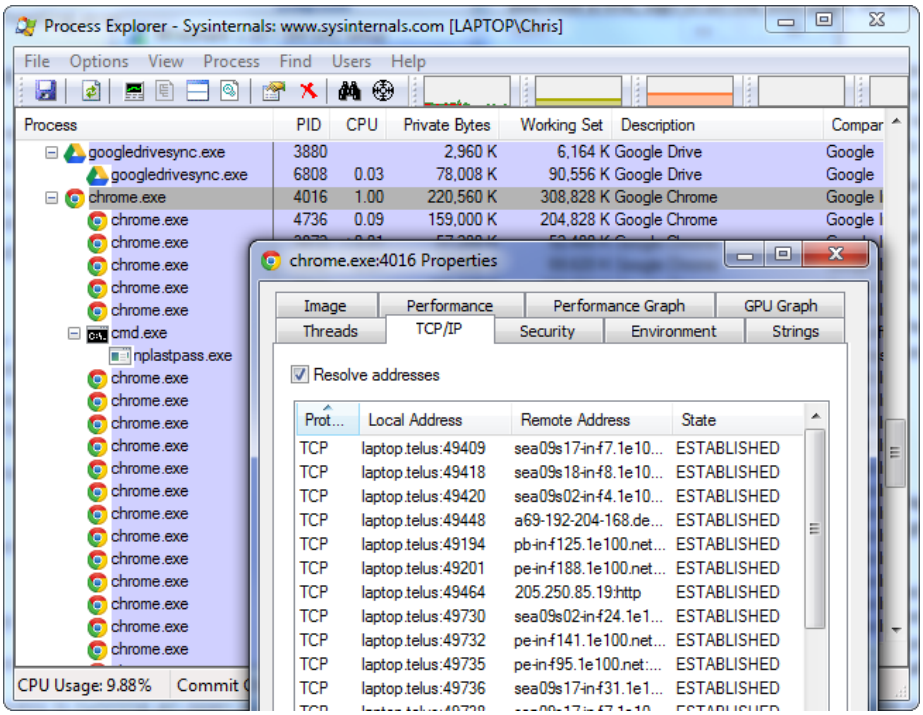
TCP stands for Transmission Control Protocol. It's the most commonly used protocol on the Internet.

When you load a web page, your computer sends TCP packets to the web server's address, asking it to send the web page to you. The web server responds by sending a stream of TCP packets, which your web browser stitches together to form the web page and display it to you. When you click a link, sign in, post a comment, or do anything else, your web browser sends TCP packets to the server and the server sends TCP packets back. TCP isn't just one way communication — the remote system sends packets back to acknowledge it's received your packets.

TCP guarantees the recipient will receive the packets in order by numbering them. The recipient sends messages back to the sender saying it received the messages. If the sender doesn't get a correct response, it will resend the

packets to ensure the recipient received them. Packets are also checked for errors. TCP is all about this reliability — packets sent with TCP are tracked so no data is lost or corrupted in transit. This is why file downloads don't become corrupted even if there are network hiccups. Of course, if the recipient is completely offline, your computer will give up and you'll see an error message saying it can't communicate with the remote host.

Process Explorer and other system utilities can show the type of connections a process makes — here we can see the Chrome browser with open TCP connections to a variety of web servers.



How UDP Works

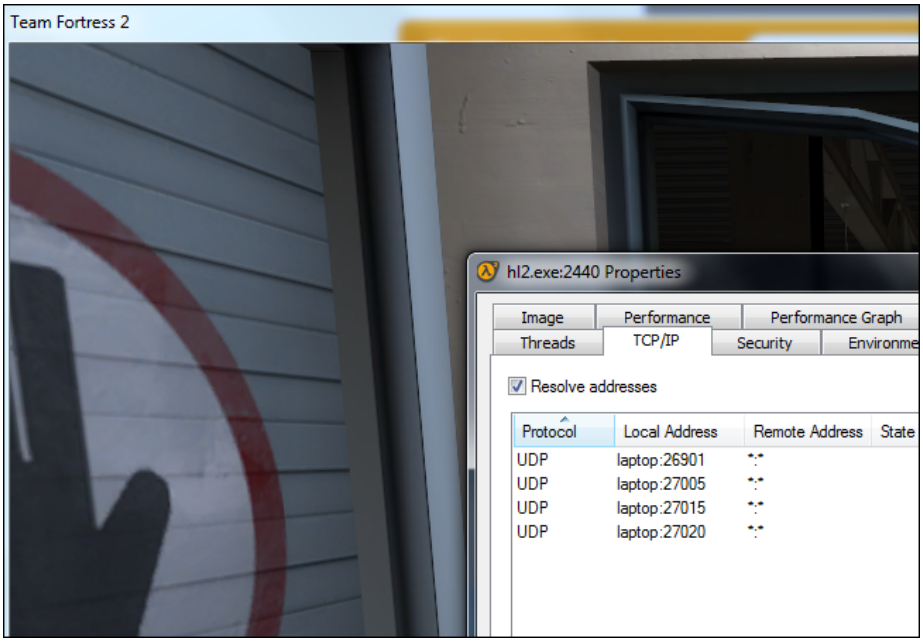
UDP stands for User Datagram Protocol — a datagram is the same thing as a packet of information. The UDP protocol works similarly to TCP, but it throws all the error-checking stuff out. All the back-and-forth communication and deliverability guarantees slow things down.

When using UDP, packets are just sent to the recipient. The sender won't wait to make sure the recipient received the packet — it will just continue sending the next packets. If you're the recipient and you miss some UDP packets, too bad — you can't ask for those packets again. There's no guarantee you're getting all the packets and there's no way to ask for a packet again if you miss it, but losing all this overhead means the computers can communicate more quickly.

UDP is used when speed is desirable and error correction isn't necessary. For example, UDP is frequently used for live broadcasts and online games.

For example, let's say you're watching a live video stream. Live streams are often broadcast using UDP instead of TCP. The server just sends a constant stream of UDP packets to computers watching. If you lose your connection for a few seconds, the video will freeze for a moment and then jump to the current bit of the broadcast, skipping the bits you missed. If you experience minor packet-loss, the video or audio may be distorted for a moment as the video continues to play without the missing data.

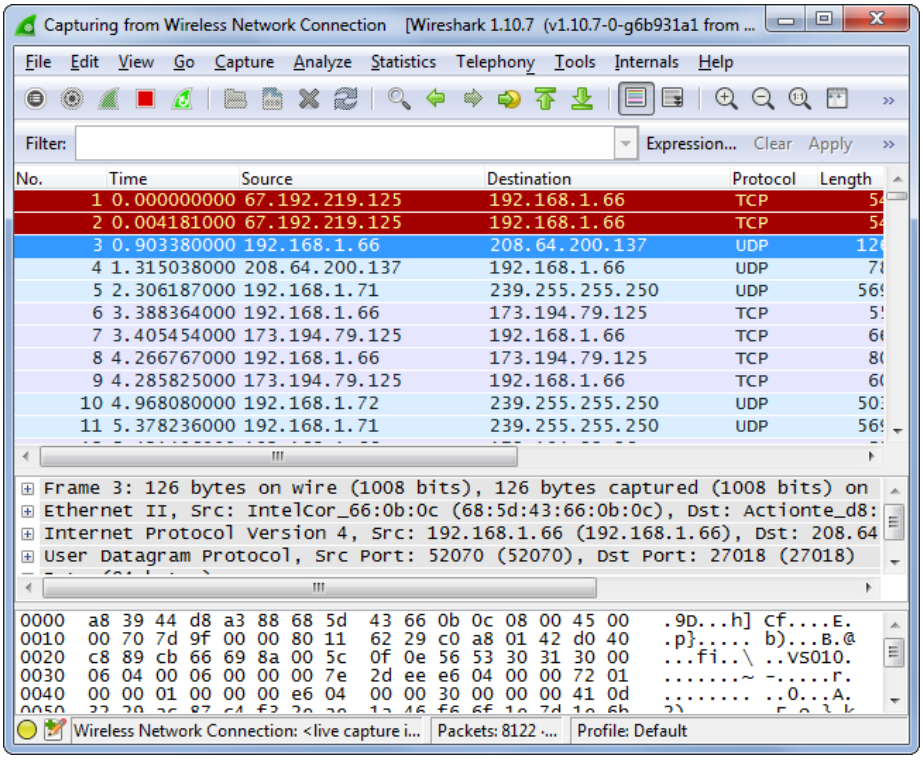
This works similarly in online games — if you miss some UDP packets, player characters may appear to teleport across the map as you receive the newer UDP packets. There's no point in requesting the old packets if you missed them, as the game is continuing without you. All that matters is what's happening right now on the game server — not what happened a few seconds ago. Ditching TCP's error correction helps speed up the game connection and reduce latency.



So What?

Whether an application uses TCP or UDP is up to its developer. This really depends on what an application needs — most applications want the error-correction and robustness of TCP, but some applications need the speed and reduced overhead of UDP. Use a network analysis tool like WIreshark and you'll see the different types of packets travelling back and forth.

Unless you're a network administrator or software developer, this shouldn't doesn't affect you too much. If you're configuring your router or firewall software and you're not sure whether an application uses TCP or UDP, you can generally select the "Both" option to have your router or firewall apply the same rule to both TCP and UDP traffic.



ICMP, or Internet Control Message Protocol, is another common protocol used on the Internet. It's generally used for query and error messages between routers and other network devices. Applications you'll use generally don't use ICMP, although ping and traceroute use ICMP for their diagnostics.

JOIN THE DISCUSSION (2 REPLIES)

Chris Hoffman is a technology writer and all-around computer geek. He's as at home using the Linux terminal as he is digging into the Windows registry. Connect with him on Google+.

- **Published 06/1/14**

MORE ARTICLES YOU MIGHT LIKE