



# Path MTU Discovery

Peter J. Welcher

## Introduction

The last three articles have been the quick visual tour of Call Manager. There are a bunch of related topics we didn't get into: Features, Services, Serviceability, the Help system. But we did cover the fundamentals. It seems like time to move on, since there might be a more than a few folks who aren't all that interested in Call Manager, despite its charms. There are some final IP Telephony [thoughts](#) at the end of this article.

For this month's article, I thought we'd take a look at one of those issues that seems to keep coming up, in different way and for different folks. That topic is Path MTU Discovery. This article should have something for everybody, since this topic affects not only networking staff, but server and firewall / security staff as well! And if you wear all three hats in your shop, you now have three good reasons to read this article!

We'll start by taking a look at how Path MTU Discovery (P-MTU-D) is supposed to work, and why it's generally considered desirable. We'll then look at how well-meaning security administrators break the protocol, why it breaks, and what the symptoms are. We'll finish by talking about the various ways to fix the problem.

## What Is Path MTU Discovery?

Path MTU Discovery is the recommended mechanism for servers to send optimally-sized segments to clients.

As you may know, if IP is asked to deliver a packet on an interface with MTU (Max Transmission Unit) smaller than the packet size, IP fragments the packet. The ultimate receiver of the fragments then has to re-assemble the fragments. This ultimate receiver is probably not the next-hop router. That's good, because there's no point in re-assembly in the middle of the network. Some subsequent link might also have a small MTU, and then the network device transmitting onto that link would have to fragment the packet all over again.

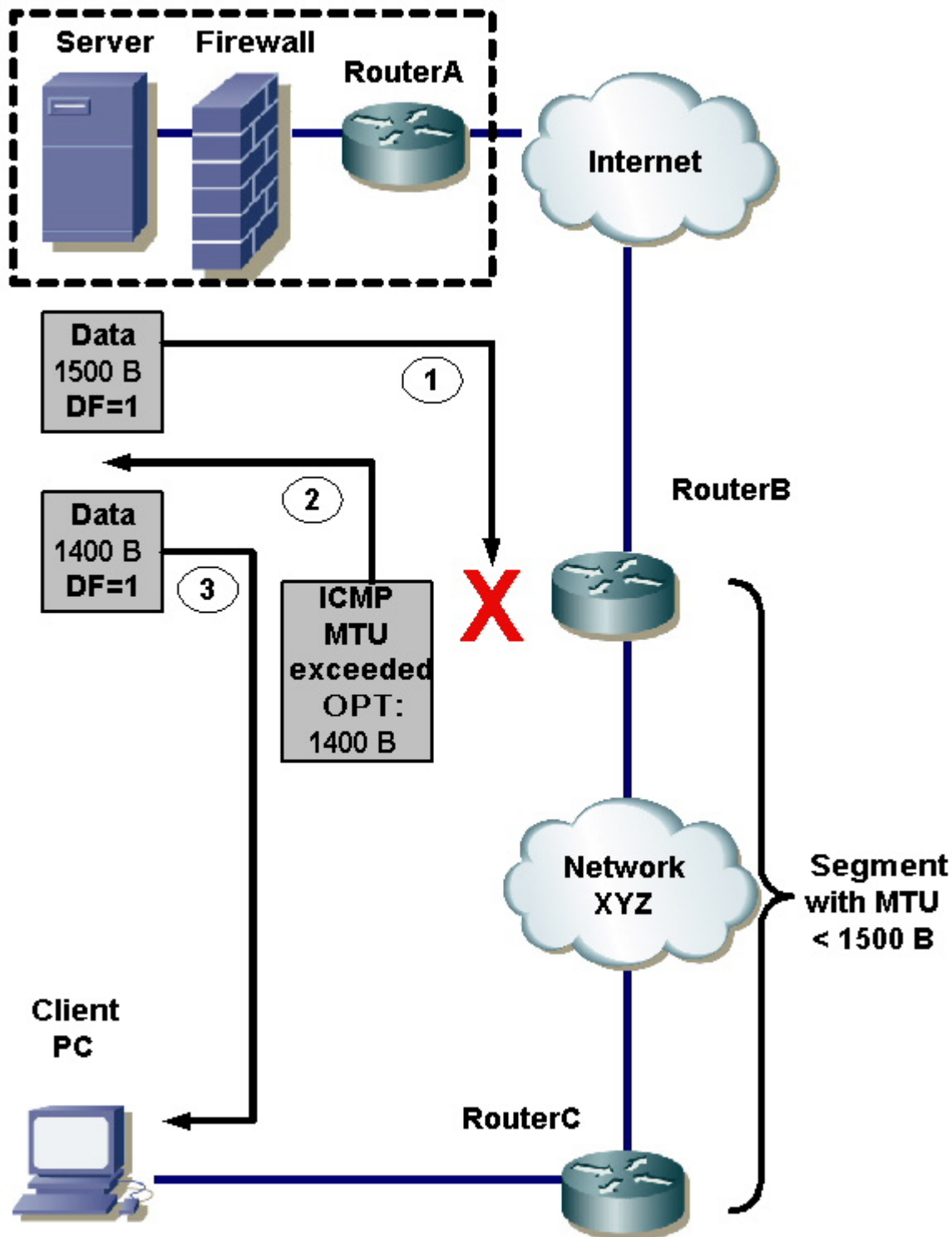
IP Fragmentation is considered evil (well, undesirable), partly because it imposes a buffer and CPU burden on the receiver. The receiver has to collect the fragments in a buffer, which may persist and tie up memory for minutes if some fragments are missing. But worse, if even one fragment is lost, the sender has to retransmit the packet, which in effect probably causes retransmission of all the fragments coming from that packet. This means one lost fragment results in several being retransmitted, which may well make congestion and loss worse along the path.

The intent of Path MTU Discovery (P-MTU-D) is to greatly reduce the use of IP Fragmentation in the Internet. Fragmentation is now perhaps in the status of an experiment that didn't quite work out, with a better answer now known (Path MTU Discovery). Let's see how it can do that, by examining its underlying mechanism.

## Details of How it Works

If you follow some of the links below, you'll see that it is possible to say quite a bit about P-MTU-D. I'll try not to do so, however.

The following figure will be used to discuss how P-MTU-D works.



In Step 1, we see the server sending a 1500 Byte packet. It reaches the middle router, RouterB, which has been configured to know that the Network XYZ link has MTU smaller than 1500 B. Normally, RouterB would fragment the packet, and life would be good. But since the server has set the Don't Fragment flag (DF=1) in the IP header, RouterB is not allowed to fragment the packet, and must instead drop it.

When RouterB drops the packet, it sends back an ICMP error message, indicating the MTU "Can't Fragment" error. The newer form of this error message (since 1993) requires that a router include 16 bits indicating the MTU of the next hop link. In effect, the router tells the server "Not that big please, here's the appropriate size to use."

The server then drops its transmit segment size down to the indicated MTU, retransmits, and continues using that segment size for subsequent transmissions. Presto, no fragments on the network!

I hope you're wondering, why the server would set DF=1. Good question!

This is the one easy indicator that the server is doing P-MTU-D. Most modern servers {UNIX, Linux, Windows} do P-MTU-D. They set DF=1 to, in effect, force a router with an MTU issue to send back information about the preferred smaller MTU. This means they can subsequently transmit segments of the ideal size to the client PC.

## Where Would I See P-MTU-D?

This perhaps leads to the question, well, where do I really need P-MTU-D? After all, don't most modern networking media support 1500 B MTU sizes?

And that's a good thought. When it was invented, P-MTU-D was very useful if you had servers on Token Ring or FDDI. That's because the media supported larger frame sizes, so servers would transmit large packets of 4000 B or larger, and these would need fragmentation for Ethernet or serial media.

Well, Token Ring is now Officially Dead, since IBM has announced conversion to (Fast) Ethernet as part of an internal IP Telephony initiative. And FDDI is End of Life, they don't even make the chips anymore.

But there are two new settings where MTU can still be an issue. The first is with home broadband. If you're doing DSL, you might well be doing PPP Over Ethernet (PPPOE). The extra header bits shrink your effect MTU size on the link from your home to the DSL provider (Network XYZ in the above diagram). The second place MTU issues show up is if you're doing IPsec VPN, usually if you are **not** doing split tunneling. That is, usually when you force Internet traffic through a central site, then across the IPsec VPN to remote sites. This is becoming more common even though it is somewhat wasteful of bandwidth, because you can use central firewalling and IDS and so on to audit all Internet traffic, without buying and managing IDS' for each site. In this second case, the internal IPsec VPN plays the role of Network XYZ in the above diagram. Physically, the bits are probably traversing your ISP, but with extra IPsec and perhaps GRE headers imposed. Those extra headers reduce the effectively usable MTU.

So I hope the above diagram isn't confusing, I was just trying to use one figure for both scenarios. In both cases, the smaller MTU link is the one near the home or small office client.

By the way, if you're doing GRE tunnels, or your provider is doing L2TP, then you may have similar issues.

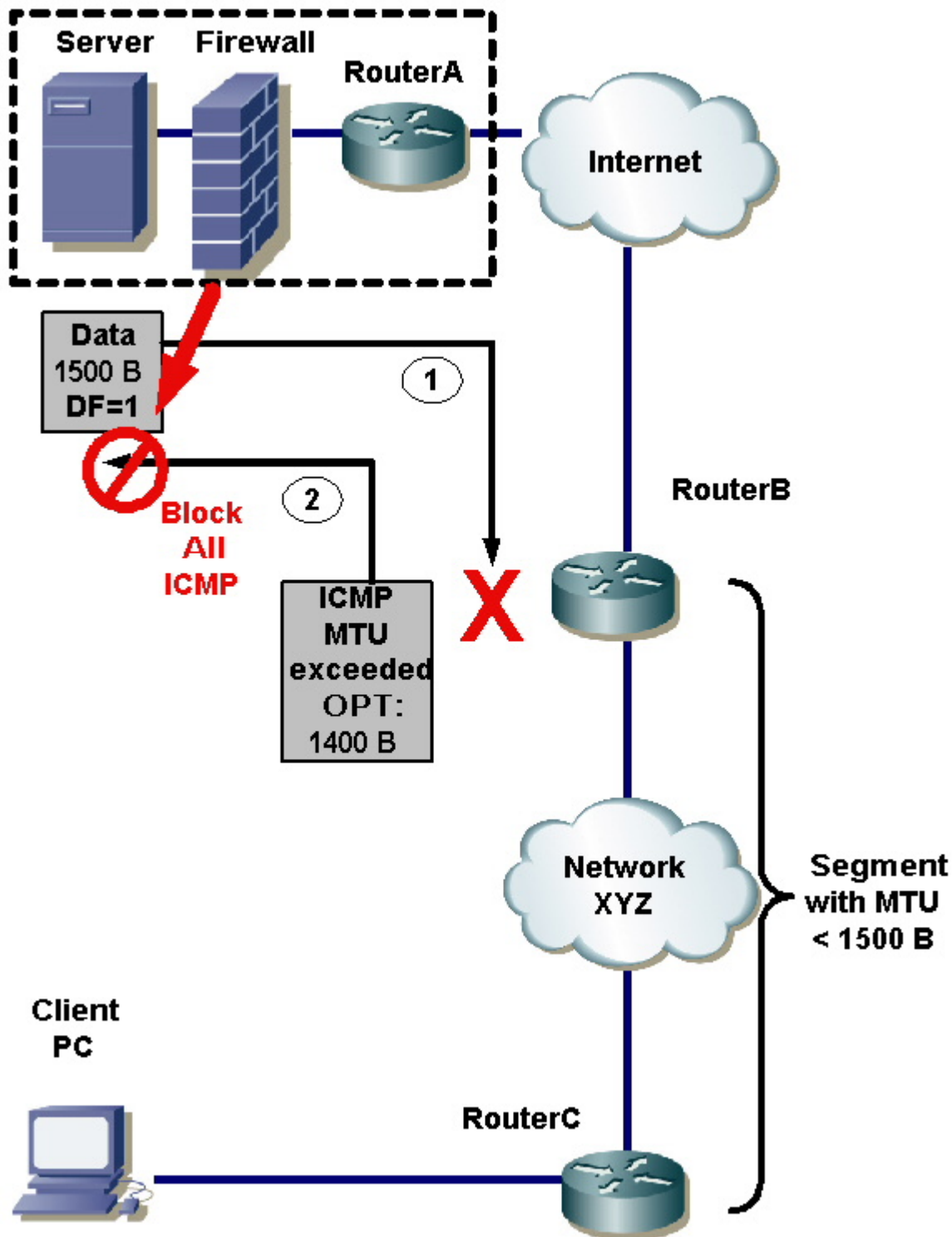
By the way, the Stevens book (see [below](#)) makes the point that the Solaris implementation of P-MTU-D used to back off to the full MTU (1500 B) every 2 minutes. The purpose of this would be to re-initiate P-MTU-D, just in case the route had changed to allow a path that did not require a smaller MTU size. After all, you'd hate to be sending small segments and experiencing lower throughput and higher CPU when there was no good reason to do so.

Cisco does support P-MTU-D for IPsec and GRE tunnels. In fact, some of the links below recommend you turn it on. In principle, that's great advice. In the real world, there's one more factor to consider.

## How Well-Meaning People Break P-MTU-D

Some folks are using a very simple firewall ruleset which blocks all ICMP. This is intended to prevent Denial of Service (DoS) attacks. I personally think it might be better to use QoS to rate limit ICMP traffic, if one is worried about that.

The problem comes about because when the server site blocks all ICMP, those helpful ICMP messages saying what segment size to use aren't getting through to the server. Consequently, the server just keeps sending out 1500 B segments (or whatever its MTU is for its local media). And the server never gets the word that this size won't work. Worse, because modern servers now default to doing P-MTU-D, they are setting DF=1 so the server's packets are just getting discarded at RouterB.



I'm told Windows 2000 does eventually react to lack of acknowledgements by dropping the sending segment size, perhaps to 1/2 the former size. This does solve the problem, at the price of substantial delay before packets start reaching the client. An alternative is to set the server MTU size appropriately, if your site security folks are filtering all inbound ICMP.

There's one nasty aspect to this I haven't mentioned yet. If your users are doing file transfers or viewing web images off an Internet server for another company, and that other company is blocking all inbound ICMP packets, then your users will start complaining. The file transfers or web pages don't work! It must be your fault! I'm not going to name names in print, but I'm told some major sites (news and financial information sites) that ought to know better have this problem.

The part that's nasty is that you the administrator can't do much to solve your user's problem (see below). Hmm, let's see, you could email the webmaster at the affected site. Now all you have to do is clean up the Internet ... just a few sites ...

shouldn't take long... NOT! Besides that, what are the chances your email to webmaster@WhateverTheServerSite.com is actually getting to a human? One who is inclined to take your word concerning his or her security?

## Working Around the Problem

To explain the workarounds, we need to get a small amount more technical.

TCP normally looks at the MSS (Maximum Segment Size) sent by the other end in setting up a connection. And it looks at the local computer's MTU for the outgoing interface. The smaller of these two numbers is used as the starting segment size for transmissions.

So the first workaround is to change the MTU. If you're a server administrator and know your firewall people are filtering all inbound ICMP messages, you can reduce your server MTU sizes to say 1400 B. If you're a corporate network person and your users are complaining because some file downloads and some web sites don't work, you can run around and change the MTU size on all the PC's to 1400 B.

The only drawback to that is that it takes work.

Cisco has a new feature in the smaller home routers that intercepts the opening of the TCP connection, and over-rides the MSS size transmitted by your users. Typically, you'd tell the router to change the MSS to 1400 B or so. So whatever server your users are connecting to through that router, that server will send segments smaller than what it thinks your users' MSS is, namely smaller than 1400 B. That's pretty clean! And a lot less work. The only drawback: very recent Cisco code, risk of bugs, and possible load on the CPU of the router. The feature does appear to be present in 12.2(4) T, you need to check under the WAN part of the manual. (I'm noting this since I couldn't find the feature in the IP section, and had begun to doubt it was broadly implemented in Cisco IOS.)

See for instance the URL

[http://www.cisco.com/en/US/products/sw/iosswrel/ps1839/products\\_command\\_reference\\_chapter09186a008010a3c4.html#1064471](http://www.cisco.com/en/US/products/sw/iosswrel/ps1839/products_command_reference_chapter09186a008010a3c4.html#1064471)

This seems to be a clean solution. If it is available to you, use it!

For DSL environments, the Service Provider should be setting the MTU on their RouterB to something around 1490 (or smaller). And you can easily set the MTU on your home PC(s) similarly, assuming your PPPoE driver install doesn't do this for you.

For IPsec or IPsec + GRE environments, there are several good Cisco documents, but none quite solves this problem. Part of the issue here is that your alternatives have expanded with the more recent Cisco IOS code.

Some of the Cisco materials I've seen make the very good point that fragmentation and IPsec don't mix well. The thing that had me scratching my head was that the recommendation seemed be to fragment anyway.

The relevant point is that if you fragment at the interface or GRE level, you are fragmenting before IPsec encryption. These fragments can travel to the ultimate endpoint before re-assembly. Yes, you had to fragment, but you can't do much else. By way of contrast, if the router encrypts into IPsec, and if you're doing tunneling, and after all this the router then has something too big to transmit, it has to fragment the encrypted packet at the IP level. The consequence is that the receiving IPsec endpoint has to process switch while re-assembling the fragments, before it can hand something off to IPsec hardware to decrypt. That's very slow and also a healthy CPU performance impact. By the way, in the Cisco routers, you can over-ride DF=1 for both IPsec and GRE tunnels. But until recently, all that let you do was fragment and be inefficient. Coupled with pre-fragmentation, you can now fragment and stay efficient. **Do note that you have to be using IPsec tunnel mode, not transport mode, for this to work.**

**The option here for those using IPsec tunnel mode and GRE tunneling on older code seems to be to use a route-map to clear DF. (This has to be applied inbound to the Ethernet side of the router, using policy based routing commands.) In tandem, set a smaller MTU on the GRE tunnel interface. Then (if I'm correct about this), fragmentation occurs before the GRE header is applied. (After would force reassembly before removal of the GRE header, which would be process switched and slow).**

See the following links for details. The first feature allows you the option of ignoring DF=1, the second is a new feature that makes IPsec smarter about fragmentation.

DF Bit Override Functionality with IPsec	<a href="http://www.cisco.com/en/US/products/sw/iosswrel/ps1839/products_feature_guide_09186a0080087ae1.html">http://www.cisco.com/en/US/products/sw/iosswrel/ps1839/products_feature_guide_09186a0080087ae1.html</a>
---	---

Tunnels	
Pre-fragmentation for IPsec VPNs	<a href="http://www.cisco.com/en/US/products/sw/iosswrel/ps1839/products_feature_guide_09186a0080115533.html">http://www.cisco.com/en/US/products/sw/iosswrel/ps1839/products_feature_guide_09186a0080115533.html</a>

So now you have your choice of good workarounds, if you don't mind using recent Cisco IOS code. Before that, yeah, you're kind of stuck altering client and/or server MTU's.

## Some Final IP Telephony Thoughts

To wrap up the IP Telephony theme from the last three articles, I'd like to mention a couple of things in passing, in the hope that they help.

First, we at Chesapeake Netcraftsmen do now have our Cisco IP Telephony (Revised) Partner Certification. That doesn't mean we're new to IP Telephony, it just means we finished taking the tests and getting formal recognition for our skills.

Second, I've really been enjoying reading through the Troubleshooting Cisco IP Telephony book from Cisco Press. **Troubleshooting Cisco IP Telephony**, by Paul Giral, Addis Hallmark, Anne Smith. See the Amazon URL <http://www.amazon.com/exec/obidos/ASIN/1587050757/> for details. The part I've read has been well-written, clear and well-organized. And it contains all sorts of good things you can do to help yourself out when troubleshooting Cisco IP Telephone problems.

Third, if you're a CallManager administrator, you perhaps ought to look at the documentation for ART and BAT. For those who don't recognize those acronyms: Administrative Reporting Tool, and Bulk Administration Tool. It seems like there's a good article lurking there! See the links:

- 1 [http://www.cisco.com/univercd/cc/td/doc/product/voice/sw\\_ap\\_to/admin/admin\\_rp/1\\_1\\_1/user\\_gd/index.htm](http://www.cisco.com/univercd/cc/td/doc/product/voice/sw_ap_to/admin/admin_rp/1_1_1/user_gd/index.htm)
- 1 [http://www.cisco.com/univercd/cc/td/doc/product/voice/c\\_callmg/3\\_3/bulk\\_adm/4\\_4\\_2/index.htm](http://www.cisco.com/univercd/cc/td/doc/product/voice/c_callmg/3_3/bulk_adm/4_4_2/index.htm)

## Conclusion

I highly recommend the book W. Richard Stevens *TCP/illustrated (volume 1)* book, <http://www.amazon.com/exec/obidos/tg/detail/-/0201633469>. See especially Section 11.5: ICMP Unreachable Error, Fragmentation Required, and Section 24.2: Path MTU Discovery.

Some other Cisco links follow. These can help you determine what MTU or MSS size to use, among other things.

MTU Tuning for L2TP	<a href="http://www.cisco.com/en/US/tech/tk801/tk703/technologies_tech_note_09186a0080094c4f.shtml">http://www.cisco.com/en/US/tech/tk801/tk703/technologies_tech_note_09186a0080094c4f.shtml</a>
IP Fragmentation and PMTUD	<a href="http://www.cisco.com/en/US/tech/tk827/tk369/technologies_white_paper_09186a00800d6979.shtml">http://www.cisco.com/en/US/tech/tk827/tk369/technologies_white_paper_09186a00800d6979.shtml</a>
Adjusting IP MTU, TCP MSS, and PMTUD on Windows and Sun Systems	<a href="http://www.cisco.com/en/US/tech/tk472/tk473/technologies_tech_note_09186a008011a218.shtml">http://www.cisco.com/en/US/tech/tk472/tk473/technologies_tech_note_09186a008011a218.shtml</a>
Why Can't I Browse the Internet when Using a GRE Tunnel?	<a href="http://www.cisco.com/en/US/tech/tk827/tk369/technologies_tech_note09186a0080093f1f.shtml">http://www.cisco.com/en/US/tech/tk827/tk369/technologies_tech_note09186a0080093f1f.shtml</a>

If you've found any other alternatives to the P-MTU-D issue, please email me about them, and I'll mention them in a future

article.

At this point, I have no idea what next month will bring. If you have ideas or suggestions for articles, please let me know! If you have an interesting network design or troubleshooting case study that you don't mind exposing in public to some degree, by all means, please get in touch!

---

Dr. Peter J. Welcher (CCIE #1773, CCSI #94014) is a Senior Consultant with Chesapeake NetCraftsmen. NetCraftsmen is a high-end consulting firm and Cisco Premier Partner dedicated to quality consulting and knowledge transfer. NetCraftsmen has nine CCIE's, with expertise including large network high-availability routing/switching and design, VoIP, QoS, MPLS, network management, security, IP multicast, and other areas. See <http://www.netcraftsmen.net> for more information about NetCraftsmen. Pete's links start at <http://www.netcraftsmen.net/welcher> . New articles will be posted under the Articles link. Questions, suggestions for articles, etc. can be sent to [pjw@netcraftsmen.net](mailto:pjw@netcraftsmen.net) .

5/5/2003, updated 3/9/2004

Copyright (C) 2003, 2004 Peter J. Welcher