SECURITY NOW!

Transcript of Episode #139

Net Congestion

Description: Steve and Leo discuss an aspect of the "cost" of using the Internet - a packetized global network which (only) offers "best effort" packet delivery service. Since "capacity" is the cost, not per-packet usage, the cost is the same whether the network is used or not. But once it becomes "overused" the economics change since "congestion" results in a sudden loss of network performance.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-139.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-139-lq.mp3

INTRO: Netcasts you love, from people you trust. This is TWiT.

Leo Laporte: Bandwidth for Security Now! is provided by AOL Radio at AOL.com/podcasting.

This is Security Now! with Steve Gibson, Episode 139 for April 10, 2008: Network Congestion. This show and the entire TWiT broadcast network is brought to you by donations from listeners like you. Thanks.

Time for Security Now!, the podcast that helps you stay safe online, with our security guru, Mr. Steve Gibson of GRC.com and SpinRite fame. Hi, Steve.

Steve Gibson: Yo, Leo, it's great to be back with you. We're approaching Episode 140. This is 139, so...

Leo: And TWiT is stalled in the water because I'm in Australia. And so you...

Steve: Oh, darn.

Leo: Oh, darn. Pulling ahead now.

Steve: Yup.

Leo: Well, good. I'm glad we could do that. Now, last week we promised we'd talk a little bit about, I guess, Net Neutrality and...

Steve: Well, it's sort - maybe, I mean, that's the only common jargon that I know of to refer to this. But it's sort of really not that. As I understand the Net Neutrality argument, you know, I don't have any real interest in the politics. I'm Mr. Technology and Security more than filtering out or giving preferential bandwidth to one party or another. But I know that the topic has come up. We've talked about how some ISPs are, like, dropping people's connections. They're, like, doing sort of nefarious, behind-the-scenes - and initially not admitting to doing so - sending spoofed packets at the connection endpoints of people, for example, who are using BitTorrent or downloading massive blobs of data, to sort or curtail them from doing that. So I started doing some analysis and ran across much more that was going on, like in the IETF, in the Internet Engineering Task Force, about basically they're working on dealing with this huge transformation which has come about only really in the last few years as we've moved from using the Internet primarily for email and web browsing, and now for massive content delivery. So I want to talk about it.

Leo: And, you know, when they say "Net Neutrality," I think the biggest issues of Net Neutrality are the political issues involved, where an ISP, let's say Comcast, might block Skype traffic, not to protect its network, but for anticompetitive reasons, to protect their Voice Over Internet product. We're not going to talk about that today.

Steve: No, and in fact one of the things I saw, maybe an even more annoying example, is this idea that an ISP could give preferential treatment to YouTube as a content supplier to their customers, versus some other content supplier that is not paying a premium to, like, get premium traffic treatment. And that's what really has people upset is because, I mean, really then it really does bias this notion of all the content being treated equally. And I think that underlies this issue of Net Neutrality. What we're going to talk about today of course is not the politics. I want to really discuss something we've never talked about before, which is what happens when you try to push too much data through a narrow pipe. How do the fundamental Internet systems, routers and protocols deal with that? Because it turns out it has a real, I mean, that's ultimately what's causing the problem is there isn't enough bandwidth. So people are scrambling around trying to figure out, uh-oh, how do we react to that?

Leo: Well, good. That's good. So we're going to talk about it from an engineering point of view in particular.

Steve: Right.

Leo: Before we - we don't have - we are prerecording this before I go to Australia. So I don't imagine you have any addenda from the three shows we've already recorded this week.

Steve: Well, I don't have, unfortunately, any security updates because for the last two weeks we're prerecorded. And so we've been blind to whatever might have happened, and we're now into the beginning of August - I'm sorry, August - the beginning of April.

Leo: In your dreams.

Steve: We'll be catching up next week with any important security news that has occurred since. I had one interesting note that I found, actually this was email forwarded through Sue, my operations gal. Someone named Dan Linder wrote. It was something that I thought was really just kind of cute. He said, on Episode 136 you mentioned some listener feedback pointing you to a Google search for detachable key ring. Remember this was some guy who was sort of being just a little bit of a smart aleck and saying, hey, Gibson, you know, rather than having to TrueCrypt your thumb drive on your key chain, have you ever heard of a detachable key ring? Anyway, so Dan, being a little bit of a punster, says, if you end up splitting your keys up like this, doesn't this become a physical implementation of your public key and private key?

Leo: There you go. It's true. It's true. That's very funny.

Steve: So I thought that was great. I wanted to give him some credit for that. And I did have - a question came up when I was running through all of our mailbag from last week in Q&A, a question that a number of our listeners have asked, so I wanted to share it and an answer. And as it happens, this was bundled with his SpinRite story. So I thought I'd incorporate that. This is just - he identifies himself as Dave. Oh, no, he says his last name in his Gmail, David Crisman. And the subject was "A SpinRite story and a question."

He says, "Hey, Steve, I wanted to write in to thank you for all the work you've done with both Security Now! and SpinRite. I've been listening from the start. And in addition to being very interesting, many of the topics you've covered have greatly helped my understanding in the classes I've taken for my computer science degree, particularly the series you did early on about basic network technologies" - well, he's going to like today's episode, then, too - "as well as the discussions on various types of encryption. I just had one question for you. But first I thought I'd share the story of my personal experience with SpinRite.

"Last year, given my part-time job for the campus IT department, I came to be known as the guy to go to in my building when someone was having trouble with their computer. So I wasn't too surprised when one evening someone came to my room asking for help with his girlfriend's machine. I followed him down the hall, and a bunch of their friends were gathered around the room. They explained that the computer was bluescreening on boot, and the girl who owned it was in tears because she was afraid that she had lost all the things she'd been working on for her classes. I played around with it for a few minutes, but it didn't take long for me to realize that SpinRite could almost definitely do the job. I told them I'd be back in a few minutes. Unfortunately, being on the budget of a college student, I didn't yet own a copy. After quickly checking to make sure I could afford it, I hopped onto GRC.com. In less than 15 minutes I had purchased my copy of SpinRite, burned it onto a CD, and popped the disk into the drive of the offending machine."

Leo: It's a small - what is it, 70K? It's a very small download.

Steve: Yeah, it pissed me off when it went above 64 because it used to be...

Leo: 64K, folks, not 64MB.

Steve: It used to be a COM file. Remember the old DOS COM, you know...

Leo: They had to be under 64K?

Steve: Exactly, because they fit into a single segment of memory that you were able to address with 16 bits. So it was 16-bit code that fit into 64K. And it was a major revamp when it's like, oh, I can't...

Leo: It's an EXE, oh.

Steve: I had to switch to an EXE and have multiple segments. It's like, okay, fine.

Leo: Boy, that's a long time ago. Wow.

Steve: So he says, "It ran for about two hours. Then all of us gathered around the desk to see what would happen when I turned it on. Of course, it booted up, good as new, and all..."

Leo: I bet he got a kiss for that.

Steve: I would hope, "...and all the files were recovered. She greatly appreciated the help, and many of the people there asked about this great program I had used. I pointed them all in your direction. So hopefully you sold a few more copies as a result." Or if not then, then perhaps when any of them have a problem. So he says, "So thanks a lot, Steve, for creating such a great, compact product, along with a quick and easy system for acquiring it. I know that my story isn't nearly as impressive as many of the others you've gotten. No files worth millions of dollars, and it didn't run for months to do its job. But I just thought I'd share it with you to express my appreciation." Well, you know, his story was as great as any that we've received. So thank you for sharing that, David.

He says, "Anyway, here's my question. Do you have any advice or book or article recommendations for a newly graduated programmer entering the workforce, like me, to prevent the code that I write from becoming the cause of the next major PR nightmare security vulnerability?" He says, "I realize that it's a bit of a broad question, and there may not be much I can do if I end up on a team where my work is only a tiny part of the overall project. But I'd just be interested to hear your thoughts. Again, Steve and Leo, keep up the great work, and thanks."

Leo: Well, don't use strcpy. There's number one. Right?

Steve: Yeah, exactly. Many people have asked, sort of at the similar stage in their career to where Dave is.

Leo: It's got to be scary.

Steve: Oh, I'm not surprised we've scared people. I mean, I'm scared when, literally, I mean truly, it's something that everyone who's writing code that's going to be exposed to the 'Net - and almost by definition these days, any code you write is Internet exposed. It's got some Internet-facing surface. The only thing I could suggest, I don't have any specific books, but I do know just sort of from general browsing that this topic has now been around enough and has been getting enough attention that there are a bunch of books that have been written about writing secure software, or security issues. So I would just say go to Amazon and put "secure programming techniques" or something into Amazon, and I'll bet you will find a whole bunch of really interesting and essentially topical books that address the same sorts of things we're talking about all the time.

Leo: I saw a good one the other day. I think they sent me a copy of it, and I just can't remember the name of it. Yeah, there absolutely are a number of these books. And I think that this is - now that this is such a high priority, there's got to be more and more thinking about this, more and more than just use "strncpy" instead of "strcpy." But yeah, I think you're right, go online and look because there's going to be a ton of stuff out there.

Steve: Yeah. I think, I mean, I don't have any specific recommendations, I mean, I'm programming in Assembly language, so I'm not working with any higher level libraries. I'm being extremely careful. And that's the number one thing, I think, is just be thinking about this. We've talked about how - the general role of programming, especially in a corporate environment or a team environment, where typically you're being pressed to be done. Software always takes longer than we expect, so there's deadline pressure. People are saying, is it done yet, is it done yet? Please check your code in so we can do a build and we can start testing, blah blah blah. I mean, all of that works against taking care. I mean, it works against caution. The focus is getting it done, making it work, rather than, okay, yeah, it works, but what if someone wants it not to?

And that's really the strange thing about software which I think for many people, for many programmers who really love programming, it's the thing they enjoy is that there's a mindset you get which is a belief that it's correct. And it takes a debugger to rub your face around in the fact that you're wrong, that it's not correct. And sometimes you can be staring at your own code, and it looks perfect until you step through it with a debugger that says, look, dummy, this is a zero. And it's like, oh, you know what I mean, and you have to be shown so clearly where there's a problem. And my point is that security is an even more subtle kind of bug because it's something you're just not used to seeing. We're not expecting it. And it makes writing really bulletproof code really difficult. You'd have to look at your own code skeptically and really keep in mind what the challenges are from a security standpoint.

Leo: It's one of the reasons modern programming languages have built-in testing in a lot of them, so that as you write a module, you test a module. And it's just kind of an automatic process. Anyway, good subject. And boy, I have to say, if I were a kid in college studying this stuff, that would be probably job one. You don't want to be that person that is the one in the headlines. Shall we get to the topic at hand, network congestion?

Steve: Yeah. I touched on it at the end of our episode last week, sort of teasing this episode a little bit. So, okay. Here's the issue, essentially. When you think about a network infrastructure which is in place, meaning we've got the last mile connections by cable modem or DSL. Then those all at some level get aggregated to what ISPs call their "aggregation router," which aggregates many of their clients' traffic into a single connection. So then there will be an interface that has much larger bandwidth, so it's able to carry many of the tributary bandwidth feeds and aggregate it into a larger one. And that'll then go typically to an even bigger router that is aggregating the traffic from many of the aggregation routers. And so it sort of is a classic tree structure with many branches per node feeding together into a larger and larger chunk, which ultimately then gets routed to the ISP's peering partners.

And we've talked about peering in the past. Some of our listeners, as many listeners write, they've been listening since Episode 1. They'll remember when my T1 provider, Cogent, got into an argument with Level 3. And GRC's networking technology, all of our servers and things are now being hosted in a Level 3 facility, in a Level 3 datacenter. And I was unable, over the T1s that served me personally, I was suddenly one morning unable to reach my servers because Cogent and Level 3 that had this peering agreement got into a dispute, and Level 3 stopped peering with Cogent, meaning that literally there were chunks of the Internet that I as a Cogent subscriber could not get to because they were over on Level 3 side, and Level 3 said we're not going to permit Cogent traffic into our borders any longer. So that lasted, I guess, a few days, and they settled their contract dispute. But it was sort of an interesting wakeup call of sort of the infrastructure that we all take for granted.

Well, okay. So now there's all this equipment in place. There's bandwidth and pipes and routers and all that. Now, when you think about data, data sort of being ephemeral, there isn't a difference in infrastructure cost as a function of how much that infrastructure is being used. It's all in place. If no one's using it, it costs just as much as if all the links are saturated because the bandwidth has to be there in order to carry the maximum traffic that the bandwidth allows. The hardware has to be there in case it's needed to be used. But not running it at 100 percent doesn't cost less than running it 100 percent. So my point is that there is a fixed cost for the whole capability. And until you start overusing it, there is no increased cost as you approach that saturation point.

For example, these, I mean, even at the higher level, these peering agreements are agreements between, for example in the case I was just citing, Level 3 and Cogent, where they don't charge each other for traffic transit. They figure, hey, we're getting a reciprocal benefit. We're getting a benefit that is balanced in our agreement. So we're each benefiting equally. We're just going to agree to carry each other's traffic. And that's the way at these Tier 1 providers that they operate. So it's only down at the retail level where you're negotiating with your provider about what kind of maximum bandwidth you're going to have and what you're going to pay for that.

Well, as I was researching sort of this general area of network neutrality, trying to sort of get a sense for what was going on, what I discovered was that there is a huge amount of work which is happening in academia and among the gurus that design our protocols that ultimately bubble up into new protocol support in hardware, like in Internet routers and in our own PCs that are Internet hosts, able to connect to the Internet. There's a huge amount of work being done which we're currently unaware of, that is, very little of it surfaces. And it's being driven by the recognition that we are really seeing a dramatic transformation in the way the Internet is being used. I think it's pretty clear now that the Internet is not a fad.

Leo: We can say that, yeah.

Steve: For a long people were, oh, that's just a fad, that's not going to happen. I think Bill Gates was for a while thinking, oh, you know, that's just not going to happen. He was doing MSN, it was going to be his - Microsoft was going to compete with CompuServe in the dialup modem pool to see who can have more dialup modems. And then of course Netscape happened and caught Microsoft off guard. And I think it's, as you say, it's pretty much established that the Internet is not a fad.

Leo: Even Bill Gates agrees now.

Steve: I'm sure he does. So what's happened is it's gone from sort of a messaging medium and a bursting medium, in other words, for example, when you're downloading a web page, you grab the page. As our listeners know, the browser looks at the page, which calls out its need for additional resources, images of all kinds and maybe assets from other sites, and then the browser turns around and gets those. So there's sort of an event of loading the page, and then it comes in, and we stare at it for a while and decide where we want to go from there.

So this transformation is in many different stages. For example, you and I, Leo, right now are communicating through a VoIP application, Skype, that we use over the Internet, and able to achieve, since it's being recorded at your end, the people listening to the podcast are listening to me at the other end of an Internet connection. And so now there's a persistent connection. We are very sensitive to the amount of bandwidth that we have. We are very sensitive to the percentage of dropped packets that we have and to packet delay, which is known as "jitter," in the arrival time of the packets because, if jitter were too great, bandwidth were too low, too much packet loss was occurring, my voice wouldn't sound nearly as good as it does.

Leo: And we've heard that, actually.

Steve: Exactly. I mean, it happens.

Leo: You'll hear that on the shows from time to time.

Steve: Exactly. Similarly, I was asking my sister a couple months ago, I guess I was up in Northern California during Christmas. And I was sort of saying - so I have a niece and nephew who are in high school and college. And I said, Nan, what TV do they watch? And my sister said, oh, TV? They don't watch TV.

Leo: Oh, that's great.

Steve: They watch everything - no no. They watch everything on their laptops.

Leo: Yes, yeah. Same with my kids, yeah.

Steve: And I guess that's the new model is you're using BitTorrent to download shows, or you're just hanging out in YouTube. And we've talked about how there are companies that are deliberately blocking their employees' access to YouTube during the day after doing studies showing how large a productivity drain YouTube is because people are just wanting to sit there and click on these amazing videos which are popping up all the time. So essentially what's happened is the newer content applications have put a huge drain, a huge load on the existing infrastructure. Now, ISPs are happy to charge what they can for this additional bandwidth. I mean, it used to be that a modem, you could actually use the original Internet over a modem. The pages came in a little slowly, but they loaded, and they weren't nearly as big as they are now. You certainly didn't have all this Flash and animation stuff jumping around, burning up bandwidth on every single page. So there was much less bandwidth demand. But certainly email was entirely practical over a modem.

Well, nowadays you can't take advantage of any of this next-generation technology without having substantially more bandwidth. And of course people are noticing that they're having to pay for that. Well, so we sort of laid the foundation that as long as the network is not overused, that is, it does not become congested, there isn't any problem with using it all the way up to its limit. What happens, though, when we go beyond that, when we push beyond that? The whole Internet, the genius of the design of the Internet is the fact that the designers realized you could have a workable system which simply operated on a best effort delivery principle. That is to say, my computer puts a packet of data onto the Internet. And so everything is packetized. It puts a packet of data on the Internet that's got a destination IP. And that goes to my router that probably translates the IP from my public IP to my - I mean from my private IP to my public IP, and then puts that out on my ISP's wire that goes to the closest router, which picks it up and sends it on.

And we've talked about how routing works, how at every one of these hops routers have routing tables and multiple interfaces, and all these routers are richly connected in a large network to each other so that there isn't just one route. There's an optimal route, but you may have backup routes and alternative means for getting the data from one point to another, which provides a great deal of resilience to the whole network. So individual packets are coming into routers' interfaces, and then the router examines the packet and then sends it out another interface.

Well, we also talked about how routers aggregate inherently. Routers have lots of interfaces and are accepting packets on any of them and routing them out of any. Well, it's possible for a router at any stage in this process to be receiving too much incoming bandwidth for it to send the totality of the incoming bandwidth - for it to fit out the bandwidth that it's trying to get out of. So it's because you've got multiple interfaces, you might have just an overabundance of packets that are arriving through different incoming interfaces that coincidentally at this point in time, you know, a busy time in the evening or lunchtime, for example, the router just gets overloaded. Just because of the fact that it's got multiple connections, it might be that the sum of the bandwidth of the packets that are trying to all go out of one particular interface won't fit.

Well, the first thing, the first solution is there are buffers. So all routers have buffers on their interfaces so that the packets are put on the front end of the buffer, and then the buffer empties out of the interface. Well, that helps little bits of burstiness. That is, you would certainly want some buffer so that if, like, three packets all arrived on different interfaces at the same time, bound for a fourth one, they wouldn't all just instantly collide. They'd be able to line up in the order that they were received and get their way out. But you could still, over a larger, slightly larger period of time, you could have a situation where the persistent bandwidth that is being demanded on the outgoing interface is just not sufficient to carry all the packets that are arriving.

So the router, all Internet routers have the right to simply discard packets. They just drop them. And there are various strategies that have been devised for looking at the packets that are already in the buffer and trying to drop packets that are part of recognized flows, as they're called, a flow being a connection between one source IP and one destination IP. So there are various strategies the router uses. But ultimately a router whose outbound buffer is full has no choice but to discard a packet. Well, and this is why we call it "best effort delivery." Because essentially some host computer somewhere has been putting packets onto the Internet and trusting that they're going to get to their destination.

Well, we've talked about how TCP is a reliable delivery protocol. What that means is that one way or another the TCP protocol will accept responsibility for getting all the packets through that the sending computer wants to. They may be a little delayed. They may be slowed down. But TCP, the TCP protocol is responsible for getting that message through. What happens is that when the packets finally arrive at their endpoint IP, at their final destination, the receiving TCP/IP protocol sends back acknowledgments. And what it does is it looks at all the packets that it has received so far. And the bytes in the TCP packets are numbered sequentially in order to solve the problem of packets arriving out of order, which can also happen on the Internet.

We talked about how there are different routes that packets could take. And routers will sometimes send packets that are bound normally for a congested link, they may send it out an alternative route which could be faster, which meant that a packet arriving later at a given router ends up arriving sooner at its final destination. So packets have sequence numbers that they carry that allow the final recipient to put them back into order. Once it's done that, it looks at the highest numbered byte that it has received so far and periodically sends back an acknowledgment to the sender saying, I've received every single packet you've ever sent me in this connection up until this packet carrying this byte number. So basically that acknowledgment says everything up until now I've received.

So what this simple solution does is it means that if somewhere along the way, anywhere between the sender and the recipient, a router is overloaded and unable to deliver a packet, it'll simply drop it. All routers have permission. And when you think about it, they have no choice. I mean, they've only got a finite amount of buffer space. So they end up just saying, fine, there's no way I can store one more packet in this buffer. Everybody's trying to get out of this one popular interface right now. I have no choice but to just say, sorry. And so it drops it. The router has no obligation to notify anybody.

And this is, again, part of the brilliance of the designers because they recognized that relying on routers that were congested to send some sort of help, I'm buried underwater message, well, that would just increase the congestion, even though it might be in a different direction. I mean, it's essentially the goal is to minimize congestion. So generating packets that are warning of congestion doesn't make a lot of sense. So routers simply discard packets whenever they need to. What happens on the sending end is the sender is receiving these acknowledgments from the far end which is saying I've received everything up until here. Oh, now I've received everything up until here.

Well, what happens is, if acknowledgments don't come in in a timely fashion, the sender, using some timeouts which are cleverly designed to be adaptive and to just do the right thing in the vast majority of cases, the sender says, okay, there must have been a loss at some point because remember that the recipient might have received packets after the one that was lost because the sender could be sending packets ahead of the receipt of acknowledgment. In fact, that is the way TCP operates. In order for it to operate efficiently, it has permission to send packets ahead, assuming that the acknowledgments will be delayed. So when an acknowledgment that it's expecting doesn't arrive, it starts resending - it backs up and starts resending packets from the point of its last acknowledgment, assuming that nothing since then has been received by the recipient.

Well, the other thing it does, because it assumes if a packet was dropped it was due to congestion at some part along the way, the other thing it does is it drops its speed in half.

Leo: Ah, boy. And therein lies a tale.

Steve: Well, exactly. What happens is, so it drops its speed in half in response to what it assumes is congestion, and then over time slowly ramps it back up again. So all TCP connections do this. This is the way TCP works. So everybody on the Internet is slowly increasing their transmission speed until they start seeing packets dropped, which they sense because the far end has stopped acknowledging the continual flow, the receipt of a continual flow of packets. So as soon as they lose an acknowledgment, they cut their speed in half and start speeding up again. Then cut in half and start speeding up again, and cut in half and start speeding up again, and cut in half and start speeding up. So it's sort of the waveform is a sawtooth because it's slowly increasing, then drops in half, slowly increasing and drops in half.

What this does is it is generally - what it means is, the TCP is seeking the optimal transmission rate for the prevailing conditions from endpoint to endpoint so that any point among all these routers, any number of routers, a completely nondeterministic path, we don't have any idea what path the packets are taking to get there, we just drop them on the 'Net and cross our fingers. And when we get an acknowledgment back we know that, oh, everything that I've sent up until this point has been received.

So what this means is it means a couple things. It means that all TCP connections are treated individually. So my connection from my browser to a remote server is treated in exactly the same way as somebody else's connection. Because routers drop packets randomly, they basically just say, well, I would love to hold onto this packet, but my outgoing buffer is full on this interface. I have nowhere to put it. I've got to drop it. So they drop them randomly. What that means is that just statistically all of the packet flows that are moving through a router, for example towards a destination server, they're going to generally get equal treatment because the router doesn't love any one flow more than another. It would happily route all the packets it was receiving if it could. But, gee, there's just not enough bandwidth on the link that the incoming packets are all trying to use, so it's got to throw some away.

Well, a number of things have resulted as a consequence of this sort of equal treatment of flows. What it means is that all of the connections that are trying to use a given pinch point, a given congested interface, because of the random dropping of their packets, all of the connections share equally in the available bandwidth at that point. But that means that if somebody were to open more than one connection through the same two points, they would get a larger share of that total congested bandwidth. Thus parallel file downloaders, which open multiple connections - I actually downloaded an updated copy of Corel Draw the other day. And I was sort of curious. It was downloading, and I'd been doing all this research, so I fired up netstat in a DOS box. And sure enough, there were four established connections between my computer and Corel's server. And it was using Corel's own file downloading system. So what this meant was that if there was no congestion anywhere, these parallel connections don't really help because...

Leo: Oh, that's interesting.

Steve: ...because I'm not getting any more bandwidth. I'm not - it's their server...

Leo: Well, it doesn't help in your case because your bandwidth probably exceeds any server you're going to.

Steve: Well, I guess my point is that...

Leo: Oh, actually that's when it should help, when you have more bandwidth than they do.

Steve: Well, one connection, because it's what TCP is doing, where TCP is ramping up, trying to - running as fast as it possibly can, it would be, for example, if I were using a cable modem, it would be my cable modem's upstream bandwidth, or rather downstream bandwidth as I'm downloading it. That would be the limiting factor. So Corel's server would end up finding my maximum bandwidth point because the router trying to squeeze packets to me over my cable modem connection, it would have to stop dropping them - it would have to start dropping packets coming from Corel when Corel's TCP endpoint was sending them too fast. So it would be slowing Corel down. But one TCP connection would end up maximizing our connectivity so long as there is no competition at a congestion point.

Leo: Well, but maybe it's designed to get around downloading sites that have a limit per connection. I think a number of sites will do that. They'll say no connection may have more than 500 kilobits. So by opening multiple connections you could be getting around that.

Steve: Yes. You would certainly be doing that.

Leo: And I think that's the intent.

Steve: Well, and the other thing happening is, exactly as we were saying, given that routers that are congested themselves as opposed to the transmitting site - well, okay. A perfect example is any sort of peer-to-peer program where you've got clients which are sending as fast as they can and recipients that are receiving as fast as they can. There's no throttling going on there. Everyone is trying to move these large movie files and TV program files or music files, whatever they are, they're trying to move them around as fast as they can. So in the event of multiple connections going through a single congestion point, and that might well be your own - a router very close to you, your own ISP router. So, for example, I'm using a parallel downloader, and my neighbor that's on the same network segment as me is just using his browser. Well, because I've got all these multiple connections open, if the congestion point is my ISP's router, which it probably is in that situation, then I'm getting an unfair share - well, I'm getting, I don't know, I guess fairness is a value judgment. But I am getting a larger proportion of the bandwidth through the router because all of my connections are being treated individually rather than in any kind of aggregate. And so all of mine are sharing with the total number of connections that are running through the router because packets are just being dropped at random, and TCP protocol that is running across this is having to do the best job it can of maximizing its rate of flow.

Leo: Okay.

Steve: Okay. So what this means is, this means that there is no cost to the network until we start having congestion, and that it is congestion at routers which begins to create some cost as we are trying to push the network beyond its capacity. And so the position that is being taken by the guys that are designing these next-level protocols, they're recognizing that people who are opening lots of connections and who have them open for a long period of time, in this model of thinking about the use of the Internet, they're recognizing that it's almost as if you were to count the number of dropped packets that result from someone's use of the 'Net. Not your own dropped packets. But your use is causing congestion, which is causing everybody else to have packet loss. And it's not just instantaneous loss, it's the sum of lost packets over time.

And so they're looking at ways to create some means for accountability. And, I mean, it's a hard problem because there's no way for routers that are just sort of these autonomous packet-moving boxes, there's no way for routers to know anything about their users, the ultimate sources and destinations. There's no way for them to maintain any kind of history. And an individual end-user, for example, who's using a peer-to-peer filesharing system, they've got connections branching out from them in all kinds of different directions, going to different people, meaning that they're going over different routing paths. So the only solution that people have come up with is some sort of system which looks at the current use and the history of use of individual users and begins to hold people accountable for their aggregate use of bandwidth over time.

And one interesting thing is that people have noted that, for example, say that my ISP's routers are super congested because there's a whole bunch of people using the system who are downloading large files for a long period of time. Now I come along, and I want to look at a web page. Well, it's aggravating to me if it takes a long time for me to bring the page up and to download the page's images because my ISP is so busy. And the ISP is so busy because you've got all these other customers of the ISP downloading huge files over the course of many hours during this window of time. And I just want to look at a web page. The point is that I'm suffering because I want to bring up a page. But my bandwidth requirement for the page is the same if it takes me a minute to finally get the page loaded or if I can bring the page up much faster because largely I'm going to now sit there and look at the page and then decide what I want to do next.

So there's this notion of looking at end-users' usage of bandwidth over some period of time and changing the priority of their packet-handling dynamically so that a user like myself, that isn't downloading big files, but just wants snappy use of the Internet to do a little Wikipedia research or to go find something on Google, looks at a few pages, it would be nice if I could click the links and have the pages snap up quickly. And I'm going to be using the same amount of bandwidth, but the profile of use is different. That is, I'd like the amount of bandwidth that I'm using to come in in three seconds rather than 30 seconds because that means that my experience is much better.

And there's one other aspect of this that affects this conversation, and that is, as packets are buffered more, their delay increases because then you have packet delay caused by buffers which are full, and that begins to affect realtime services like Voice over IP. So again we want to minimize congestion. We want to hopefully not overflow buffers. But what's even better is if we can keep the buffers from becoming too deep because that way we're getting timely transmission of packets across the Internet and not having lots of jitter in addition to lots of lost packets.

Leo: Well, it's all very exciting if it works.

Steve: Well, and the problem is, if you think the surface politics of this are hairy, you wouldn't believe the fighting that is going on in the technical committees because there are people who talk about flow rates. And then there are people that talk about fair use. And there people saying, well, wait a minute, what do we mean by "fair"? And what is it that people are buying? And end-users don't want variable pricing. They want fixed pricing. They want to be able to say, look, I want to know what it's going to cost me per month, and I want to know what I'm going to get in return. Because of course lots of users are wondering whether they're really obtaining the bandwidth that they're purchasing.

And it's clear that ISPs - essentially we're going to have to have an evolution, one way or another, into a different sort of model where there is some sort of accountability so that ISPs are able to prevent their networks from being overused, at the same time allow them to be fully used. Because certainly from an economic standpoint a fully used network benefits everybody. The ISP's costs are the same if it's fully used or half used. The end-users are happier if they're part of an efficient network which is being fully used because that means that, again from a theoretical economic standpoint, their cost is minimized because the ISP is not having to charge them for a network which is not fully used.

So what's going to end up happening is a change in the ISP's customer-facing contracts and relationship where what you're buying is a best effort delivery and availability, where people who are moving huge files may have their moving of huge files take longer, but there'll be a protocol in place so that people who are paying the same fixed price but not moving huge amounts of bandwidth over time will find that the network is always extremely sappy and responsive because this notion of congestion of the network from point to point will have been worked out so that the users of huge payload large bandwidth will end up having themselves throttled so that the network ends up getting used, but not overused.

Leo: Is it possible to do some sort of just-in-time solution for bandwidth? I mean, I know that streaming providers have this kind of setup where, when you need it, the bandwidth can kick in, and you don't pay for it until you need it. Why don't ISPs do something like that?

Steve: Well, at some level that's going on. For example, my relationship with Level 3. I've got what's called a 95-5 billing. I actually have a 100MB connection between the Level 3 aggregation router and my equipment. I'm paying for 15MB, what they call "15MB commit." I'm burstable up to 100MB. So, for example, when I'm transferring a file from your server to mine, Leo, when I'm grabbing the Security Now! podcast, it's 25MB that I see, and it's a short burst, I've got the file. It's like, hey, that's really cool. I'm not sitting around watching the little bar, the meter bar slowly crawl along. I'm able to go get the next one. So it's a little bit of a burst. But I'm paying Level 3 at the rate as if I were using 15MB 24/7, although frankly GRC's usage is more like 3 or 4. But 15 is the minimum that Level 3 will sell because they're not wanting to have lots of relationships with small guys. They're wanting to keep their contracts at a higher level. There are Level 3 resellers who, again, I could have gone with a reseller and been able to purchase a much smaller chunk of bandwidth. But I just didn't want a middleman in between me and Level 3. And I've been glad for that.

Leo: Yeah. I'm trying to think what we have. I think our TWiT servers are 100MB, yeah, 100MB uplink speed. But I don't know what that means in terms of day to day. They pace it figuring you're not going to use it all the time. I mean, that's what's always happened. It happened with modem pools. You'd have five users per modem, figuring they're not all going to use it at the same time.

Steve: Well, and that's the classic example. Remember, and I think I mentioned this last week, we know of our phone system that if everyone goes off-hook at the same time, the system collapses. It's designed for typical use. But no way does it have the switching capacity to allow all of its subscribers to be talking at the same time. You just don't et a dial tone when you go off-hook.

Leo: So you would overbill, but you just don't want to overbill too much.

Steve: Exactly.

Leo: I imagine there's quite a bit of theory in how much you overbilled and how, you know, I mean, and I guess that's the point is a lot of that theory's out the window now that we're doing so much more online.

Steve: Right. And the real problem is, none of the technology we have today solves this problem.

Leo: Even fiber?

Steve: No no, I mean the...

Leo: Switching technology.

Steve: Well, the hardware and the protocol. We've got, I mean, I'm now running XP. And...

Leo: Aren't you modern.

Steve: Oh, baby. And with the state-of-the-art TCP/IP stack. And there is no technology in here today in our machines to begin to deal with this next-generation problem of how people who want to use a little bit of bandwidth but would like it to be snappy, how we coexist with people who are saying, hey, I bought a cable modem, and I have a $49 a month contract that says...

Leo: Unlimited, baby.

Steve: Unlimited baby, and it's a megabit. I want to use it all.

Leo: We just bought a T1 from Covad for $379 a month. It's only 1.5MB up and down. But I presume that the reason you're paying all that extra money is because, first of all, you have a lot of upstream, but also it's kind of guaranteed. It's low latency; right?

Steve: Oh, actually I'm glad you've done that, yes. You've bought a full T1, which is 1.54MB, symmetric, both ways. And assuming that Covad is a good supplier, and they are good, their infrastructure will allow you to move all of that 1.54MB 24 hours a day. Well, in fact, T1s used to be voice links. They carried 24 64KB voice channels on a single T1. And many people used them for a long time. Corporations wouldn't have 24 pairs of copper coming in, they'd have one T1, and then they would have a multiplexer that would turn that into individual voice channels inside their corporate facility.

Leo: I'm just buying it on faith, although I have to say 379 bucks is a lot cheaper than they used to be. They used to be 1,500 bucks.

Steve: Oh, I had a pair when they were 1,500 bucks. And I'm glad now that I'm at Level 3 and paying less than that for my whole lash up. So, yes. Certainly prices have come down. The problem is that what's happened in the last couple of years with this explosion in bandwidth, it has caught, well, essentially it's caught our protocols and our hardware off guard. And unfortunately we've seen some first reactions from ISPs saying, well, we're just going to kick these connections which are using too much bandwidth. And the reason they're doing it, Leo, is they don't have any other technology. They would love to somehow throttle these flows if they had the capability to do it. But there isn't the technology in the system I've described, there isn't the technology to do that. So what they end up doing is spoofing packets and just shutting down users, which really upsets the people who are saying, wait a minute, I'm paying...

Leo: I bought this.

Steve: Yeah, you said $49.95 and unlimited bandwidth. Well, I'm trying to use my unlimited bandwidth, and you're saying no, we're going to send out dummy packets in order to shut your links down.

Leo: Yeah, the specific technology that Comcast uses is called Sandvine, which is a Canadian company. I think most ISPs use this Sandvine thing. And that's exactly what it does. It just turns off the peers. It sends a message saying, yeah, we're done, you're done, we don't have anything more to offer. Just cuts them off. And I guess I understand that. Aren't there - I guess you've just explained why there aren't - systems like Squid and so forth that would allow you to kind of throttle stuff down?

Steve: Yes. Certainly there are approaches where you could put some technology near the end-user. You need to put it, essentially, at the other end of the connection because if you go any...

Leo: Don't want to throttle everybody, yeah.

Steve: Well, but more than that, you want to see all of that end-user's traffic.

Leo: Right.

Steve: If you put it too far away or, like, at your border, it'd be much more difficult in order to - you'd have to aggregate all of the traffic information from a single user into one point in order to know what they were doing. But it's certainly the case that, for example, if you could aggregate all of the traffic that an individual user was transiting, then you could drop packets simulating router congestion before that congestion occurred and essentially throttle that user's traffic in a way that's not causing your network trouble and is not just simply dropping the connection and causing that problem. But that's expensive technology. And the ISPs are trying to avoid deploying it. It's easy for them just to see the connection and have a router or a firewall or a traffic filter pattern match and go, oh, let's send this packet out, that'll take care of the problem. And sure enough, that traffic is gone now.

Leo: I have a good friend who runs a local ISP, Sonic.net here, which is very good. He's been past-president of the California Independent ISPs Association, really smart guy. We should get him on sometime and talk about what they do as an independent ISP. He's very forward-thinking and I'm sure is very, I mean, you've got to be a little sympathetic if you understand this for companies like Comcast that are trying to ensure that the majority of users get what they expect, which is as you said, that snappy occasional service.

Steve: Right.

Leo: And it's a difficult thing. I wish they'd spend a little more - maybe if they spent a little more on infrastructure they wouldn't have to be so draconian in shutting people down. I don't know.

Steve: Well, and I guess also I think that the stigma of peer-to-peer systems, the fact that people are downloading television shows and movies, I mean, it doesn't help the effort any that the ISP is able to say, well, this is copyright violating traffic anyway. It's like, well, yes, but Net Neutrality says...

Leo: It's not necessarily, though. That's the point.

Steve: That's very true. I mean, for example, someone downloading who wants to download all past 138 of our podcasts.

Leo: That's legal.

Steve: They have every right to do so. Just click a bunch of links, baby, and start sucking that down, 100 percent legal. Yet it's going to use up a lot of bandwidth.

Leo: A lot of Linux is distributed over BitTorrent.

Steve: Right.

Leo: You can't make the assumption that it's an illegal - I hope they're not doing it for that reason. Although remember, companies - unfortunately these are big media companies as well as Internet service providers, so they do in fact have a dog in that hunt. They have some interest in shutting you down. That's what we talked about at the very beginning. Some of this may be political. Some of it may be anticompetitive. Only some of it is technical.

Steve: Right. Well, and the fact is, as I mentioned, I think it was two weeks ago when I turned on my Mac, we were talking about security problems, I downloaded a 50MB blob OS X replacement and a 39MB blob new Safari.

Leo: Right. That's a lot of bandwidth.

Steve: Yeah, two big chunks of code. But it's not because I'm doing anything wrong, it's because Apple wants to send me a new copy of the OS. Every week or two.

Leo: Hey, Microsoft just pushed out SP1 for Vista. That means there are 150 million people downloading 150MB each. Do the math. There's a lot of traffic. That's a lot.

And pretty much all at the same time.

Steve: And important because, if they don't get that, they've got security nightmares.

Leo: That's right. That's right. So one last question before we wrap this up. I find this fascinating because we're all dealing with it. I mean, there's times I'll get online, and it's slow, and it's a beast. And I don't know if I'm mad at my neighbor who's downloading all the versions of the "Terminator" movie, or if it's just that there's too many of us. How does something like all that dark fiber that's supposed to be out there, how would that help? Wouldn't that make bandwidth kind of free and plentiful?

Steve: Well, okay, a couple things. There's a dark fiber which is linking Internet, essentially Internet backbone, which was overbuilt during the whole dotcom boom. And then there's the notion of FIOS, where fiber will be coming to the so-called last mile, yeah, I mean into people's houses. And the idea of end-users being fiber connected, I mean, it just makes my eyes cross. I mean, in terms of the impact on the Internet backbone and infrastructure. I'm sure as - our listeners have been around since pre-Internet likely, many of them. And I remember that there is a change of behavior when your own last-mile bandwidth changes. That is, I was using, as you were, Leo, for a long time a modem in order to connect to the 'Net. And then when I got my T1s, or certainly when most people switched from a modem to a broadband connection, to either DSL, high-speed DSL or cable modem, it's like, whoa, just think of all the stuff I can get now. I mean, so there...

Leo: You do, you start downloading stuff.

Steve: You know what I mean? Yes. There is behavior elasticity in the type of connection you have and how feasible it is for you to do certain things on the 'Net. And so what that means to me is that people who have fiber are going to be much more inclined to grab big chunks of stuff because now they can so much more easily. I don't know. I mean, we need some solution to this problem of the individual users getting a disproportionate use of limited resources is what it really comes down to. So either you make the resources unlimited, or you come up with a means for explaining to people that, look, you can download movies, but your traffic, which we're now going to be able to recognize as such, is going to have a lower priority on our system than everybody else that just wants their web pages to come up snappy. And we're not there yet, but ultimately that's what it's going to take.

Leo: Well, I'm very interested to see how this T1 changes our experience as Skype users. I've been - you have a T1 obviously you're using. You always sound the best of all of the Skype participants here. I have been using a shared DSL, I mean, it's business-class DSL, and it's 384 up. But it's sharing it with my Internet access and everything. And every once in a while it will degrade as my email client starts to download stuff. Be very interesting. I'm going to dedicate that T1 ultimately to streaming video. But next time we do the show, next week I will have a T1 in here. And we will see if it sounds any better. I think we're at max anyway, aren't we? I don't think it's going to sound any better.

Steve: It probably won't. But I would imagine, well, first of all, DSL is going to give you a good connection so long as it's being fed with sufficient bandwidth and lack of congestion at the other end. I would guess that having a T1 connection, you're just at a higher class of service.

Leo: I'm high-class, yeah.

Steve: You're higher class, Leo. You're going into a bigger router that is probably serving T1s.

Leo: It's an Edgemark router for T1s, yeah.

Steve: So it's going to have enough upstream bandwidth to deal with all the T1s' total aggregate bandwidth, which means you're not going to have any problems there. And then I would imagine it'll be a short hop, so to speak, till you're on the Internet backbone and over to me. It'll be interesting to see, you know, right now - I remember when we've done packet traces, my traffic goes right up to San Francisco and then over to you. It's funny, due to the Cogent/Level 3 connection, my traffic goes up to San Francisco and then back down to Southern California.

Leo: No. Well, I have DSLExtreme, and they are a Southern California company. I don't know if they're peering with Cogent in L.A. or what's going on there, but...

Steve: Yeah. It'll be interesting to see. But you are switching over to Covad, that is a different outfit. And I imagine we'll have just absolutely pristine connections.

Leo: Won't that be nice.

Steve: And you'll be able to do other stuff at the same time.

Leo: Well, I'll have a different connection for that, exactly.

Steve: Exactly, and so it won't interfere with our audio connection.

Leo: I shut down everything so I don't interfere with it. But I still surf. I'm still looking at the web, and I'm still taking notes and so forth. And so there is some usage. You know, for people who use Skype, just to fill you in, Steve and I run the diagnostics as we go. And you've never seen more than six, under the bandwidth monitor, 6,250 for the audio out; right?

Steve: Right. That's as high as it will go. It'll use 6,250.

Leo: 6,250 bits per second, is that what that is, or kilobit? I don't even know what it is.

Steve: I would think that's probably 6,250 bits per second. So, no, bits per second. So 6.25KB.

Leo: Which isn't that much, really.

Steve: No, because it's doing compression, and it's doing a good job.

Leo: We also look at jitter, which is 20 right now, as low as I've ever seen it. Roundtrip time is 30 milliseconds. So this is probably as good as you're going to get on a Skype connection.

Steve: It's as good as it goes.

Leo: Yeah, yeah.

Steve: I mean, Leo, you couldn't ask for better connection, better sound.

Leo: It's pretty amazing.

Steve: It's way better than a telephone, yeah.

Leo: It's pretty amazing, yeah. Steve, always a pleasure to talk to you. This has been very interesting. And it's, I think, very important to understand how this works. You know, sometimes when we talk about streaming Internet radio or streaming video - as we're about to do, that's really why I bought the T1 is for streaming video - I have to explain to broadcasters, it's a very different model, just as you were talking about. It's virtually free as a broadcaster. Once you put up the tower and the transmitter, doesn't matter how many people listen. A million people costs you no more than 100,000. But it's the exact opposite business model for data. That's data, is the more people listen, the more data costs. But as you point out, the cost doesn't get great until it's congested.

Steve: Exactly. If you've got the network there, the routers don't care whether they're limping along doing a few packets, or whether they're saturated. And neither do your network links. All of those, it's capacity. And once you've established it, whether you're using it fully or not, the cost is the same until you start overflowing. Then you're essentially pushing cost back onto your customers because they're not getting what they're paying for.

Leo: Right. It's a hybrid model because it's like the broadcast model until it breaks down, and then it becomes more like a magazine model or some other pay-per-download model.

Steve: Right.

Leo: Great to talk to you. Thank you so much for your time, Steve. I'm in Australia as this airs, but I will be back next week. We will be on the T1. They say they're going to install it on April 14th, so...

Steve: Cool, cool.

Leo: We should - it'll be very interesting to hear if it sounds any better at all. I bet it doesn't.

Steve: I think it'll just sound - it'll be consistently better. We won't have any of those little occasional blurches and dropouts and things. And we'll be doing our 39th Q&A for Episode 140, which our listeners will hear one week from now. And we'll be going from there.

Leo: Visit Steve's site, GRC.com. He's got 16KB versions of this for those of you suffering from congestion, or just dialup. He also has transcripts, so you can read along as you listen. I think that's often very helpful on this show. There's a lot of information packed in there. And notes, other great programs, software like ShieldsUP!, Shoot The Messenger, DCOMbobulator, Unplug n' Pray, Wizmo, and of course the most famous, SpinRite, the ultimate disk maintenance and recovery utility, a must-have for anybody. If you've got disk drives, you need SpinRite. GRC.com.

Steve: It's probably worth mentioning, too, that listeners can send their feedback and questions...

Leo: Oh, yeah, for next week.

Steve: ...and thoughts and even show suggestions to me at GRC.com/feedback. And also one of the other benefits of the transcripts that Elaine does every episode is, once Google has found them, which it tends to find pretty quickly because the site's indexed by Google, you can then search all of the textual transcripts for keywords in order to find the podcasts that are topical or you remember us talking about something but can't remember which one it was. So we've got sitewide search also in the GRC menu now.

Leo: Very cool. I love that. All right, Steve. Thanks a lot. Have a great week. I'll see you next week when I'm back from Australia for Security Now!.