SECURITY NOW!

Transcript of Episode #345

Buffer Bloat

Description: After catching up with the week's news, Steve and Leo examine the growing concern over, and performance problems created by, the Internet's "Buffer Bloat," which has been silently creeping into our networks as the cost of RAM memory used for buffers has been dropping. It's easy to assume that more buffering is good, but that's not true for the Internet.

High quality (64 kbps) mp3 audio file URL: http://media.GRC.com/sn/SN-345.mp3
Quarter size (16 kbps) mp3 audio file URL: http://media.GRC.com/sn/sn-345-lq.mp3

SHOW TEASE: It's time for Security Now!. Steve is going to blow the lid off the biggest scandal in router configuration ever. It's called "buffer bloat," and you probably have it. We'll find out how you can tell and what buffer bloat is, next on Security Now!.

Leo Laporte: This is Security Now! with Steve Gibson, Episode 345, recorded March 21st, 2012: Buffer Bloat.

It's time for Security Now!. Get ready. Fasten your seatbelts. It's time to protect yourself and your privacy online with this guy right here, our Explainer in Chief, Mr. Steve Gibson. For 345 episodes he's been protecting you. Yes, sir.

Steve Gibson: And we have an explainer episode this week that I think everyone is going to find very interesting. This is something which has actually been at a low simmer for about a year and a half, when some of the serious guru designers of the Internet began to wonder why their home connections, for which they were paying a useful amount of money for "x" number of megabits, didn't seem to be performing as well as they expected.

Leo: Hmmm.

Steve: And these are the guys who did all of this, and something seemed to be wrong. It turned out that, over time, almost as you could expect, in the same way that we've had hard drives getting inexorably bigger and processors getting inexorably faster, RAM prices have been inexorably falling. And manufacturers of routers just began putting more RAM in them. In the same way that you cannot buy a small hard drive anymore, like hundreds of megs - you used to be able to, that used to be big - you can't even buy a few gig now. Similarly, you can't buy a little bit of RAM. You always get, well, a bloat load.

Leo: A bloat load.

Steve: A bloat load. So what's happened is our routers have large buffers. And it turns out that's not good. You would think, oh, that's good because then you won't drop packets. But we're going to look in detail at why the Internet needs short buffers, how its entire design has been based on small buffers and load latency which small buffers deliver, and how this sort of silent bloating of buffering throughout the Internet is already causing problems. We've got a cool way for people to determine whether they're bloated or not and to what degree. And even some things that adventurous people can do. So I think a great podcast.

Leo: Wow. Yeah, actually I first heard the phrase "buffer bloat" not so long ago on a triangulation. We had Bram Cohen, the creator of BitTorrent, on. And he, because he's doing BitTorrent Live, which we are on, we're part of their launch partnership, he learned a lot about streaming media and the problem of buffer bloat. And then he pointed to an article by Vint Cerf which he said, well, they kind of got it wrong. But he had a lot of empirical information about buffer bloat, and it was shocking.

Steve: Well, and one of the things that the torrent clients have recently started trying to do is to be better citizens on the Internet because their nature was to cause this bloat, which would, for example, collapse VoIP or regular web surfing, just there was no - this notion of fair treatment of different flows, where flows are like we've discussed connections before, unless you're very careful with that, it's difficult to guarantee that. And what's really interesting is, even reading the most recent dialogues among these super bright people, they don't really have the answer. So there's also an aspect of this which I find fascinating, which is there isn't a good answer to this problem. It's really interesting. I mean, the best minds have been scratching their heads, thinking, okay, what's a universal answer?

One of the problems is that we have such a incredibly heterogeneous environment, made up of all kinds of different stuff, many different link speeds, many different architectures. The nature of the Internet is to just be glued together with these autonomous routers which bounce packets from one place to another. So there's no way of knowing, for example, what the roundtrip delay will be between you and the place you're trying to connect to and back again. It doesn't even have to be the same in each direction because, as we know, packets can travel different routes in different directions. So this incredible freedom that we have, thanks to this cool design of the Internet, does create problems. So we're going to cover that in detail this week.

Leo: And I know you want to talk a little bit about your iPad. But before you do, I do want a little bit of a coffee thing because yesterday we had a very well-known computer programmer on, Rich Siegel, who's at Bare Bones Software. And he told me about the Black Blood of the Earth. And I have ordered some, and I will give you a report on this. You might be interested in this. The Black Blood of the Earth is a coffee that is, well, it's kind of a coffee extract. It's a cold extract using a vacuum.

Now, for $40 you get 750 milliliters of this stuff.

Steve: So it's concentrated.

Leo: Highly. He says that he recommends keeping it below 100 milliliters a day. You've got about...

Steve: You mean your own consumption.

Leo: Yes. He says you've got about a month and a half worth of caffeine by Starbucks ventis in a single bottle. So you could actually probably put yourself into fibrillation with this stuff. But I have ordered it. He says the reason he started doing it - the guy is a radiation specialist at Berkeley. And he also spends a lot of time at Amundsen Station at the South Pole. But he apparently needs coffee, but he's a diabetic, and he can't drink cream and sugar. So he wanted to create a coffee brew that didn't have the acidity of regular coffee. By cold extracting using a vacuum, apparently none of the acids are extracted, just the oils, the flavor, the caffeine. And it's slightly sweet, he says, because there is a little bit of sweetness in the bean. And when you don't have the acids...

Steve: Black Blood of the Earth. Is that what you're telling me?

Leo: It's at Funranium Labs. I've ordered a sampler. And a Stein of Science. Black Blood of the Earth.

Steve: Oh, goodness.

Leo: And this guy is pretty serious about it. We will, as soon as it arrives, I will give you - because he does single bean extracts. So he's really - there's a Guatemalan, a Colombian; there's a Rwandan. There's one called Death Wish I didn't really want to try. So stay tuned.

Steve: Wow. I ran across a crazy inventor decades ago who - and I don't remember what the project was. I was consulting for some company. And so they found this inventor guy. And he was, like, showing us his stuff. And he had designed a camera which you mounted on the underbelly of a plane and flew into a hurricane. And this thing somehow, it had spinning mirrors that could synchronize with the instantaneous wind velocity to take photographs of hailstones as they're being formed.

Leo: Wow.

Steve: And he says, oh, and by the way, I have the best coffee in the world. And so of course that caught my attention a little more even than hailstones.

Leo: Always looking, always looking, yeah.

Steve: Exactly. And his deal was the same thing. He took - I don't remember the details now. But it was cold, I remember, it was all about cold water, cold extraction. And something like he took a whole Yuban canister like you buy at the supermarket, like the large tin, and he did something with it, like maybe he just poured cold water in it or something. I don't remember now what the detail was. But then let it sit. And then he did, he definitely extracted this syrup from the result, and that was like his magic potion. And then he would mix a tablespoon or two of that in a cup, add hot water, and it made like this amazingly zero-bitter coffee that he loved

Leo: Right, right. It also doesn't stain your teeth because the acid etches your teeth.

Steve: And it could melt hailstones like no one's business.

Leo: There is a company, the chatroom's giving me a link to Toddy, a cold brew coffee system. So I'll order that, too, and I'll let you...

Steve: Of course you will.

Leo: Welcome, once again, to Coffee Talk, a subsidiary of the Gibson Research Corporation.

Steve: At least we're recording this time. We started to record.

Leo: So as long as we're off on a tangent, I guess I have to ask you, so how many iPads did you get?

Steve: Two. One 4G-LTE because I'm grandfathered in to the original AT&T unlimited bandwidth for $29.95. And so one question I had was whether they were going to honor that, whether they would still honor unlimited 4G-LTE bandwidth that everybody is saying is faster than WiFi, if you get, like, standing under the tower and in the right circumstances. When I went from the iPad 1 to the iPad 2, I moved the SIM card, and that's all that was necessary. And the iPad 2 then thought it was on the same account as the old one.

Well, I tried to play the same game this time. I did note that they were different colors and looked very different, the iPad 3 SIM card and the iPad 2. Sure enough, I swapped them, and neither of the iPads were happy with their new SIM card. So I put them back and then poked around. And essentially moving the account was as easy as logging into that AT&T account through the control panel on the iPad 3, and it showed me my different plans, and the one that was chosen is one no one else can choose anymore, but that's the unlimited for $29.95. And that one was selected, and it stayed selected, and it left me where I was. So migrating was easy.

I've been a big fan of antiglare, as you know, antiglare film. And so I've left one without the antiglare film and one with. And so my overall take is that the iPad 2 is fine. I mean, it does everything you want. The iPad 3 is a constant surprise with how clear it is. It's just, when I look at it, I'm like, wow, this is clear. Now, I now that will fade because I had that original same sort of feeling with the first eInk, where it looked like it was just printed on the screen. It didn't really look like it was real. It looked like a store demo where you have to peel that film off before you actually can use it. And that'll last for a couple months, where I look at it and just kind of marvel at this eInk technology. Like, wow, this is neat. It doesn't look like an LCD at all.

So now I have the same thing with the iPad 3. I mean, it is really clear. But people have said, well, I have an iPad 2, and I like it. Do I need to upgrade? And I would say, eh, no, probably not. There has been some interesting controversy that you've probably heard, Leo. We know that the batteries have been - somehow they squeezed substantially more watt hours into what is essentially the same size. If I put them both, lay them both flat on the table, I don't see that the iPad 3 is really any thicker than the iPad 2.

Leo: No, it's a millimeter.

Steve: It does feel heavier.

Leo: Yeah.

Steve: Yeah, it feels heavier. But the iPad 2 had 25 watt hours of total battery, whereas the iPad 3 has 43.8 watt hours. So from 25 to 43, that's a substantial change in terms of energy density. And the first thing that I thought when I saw the iFixit Teardown was, ooh, that's going to be slower to charge. And sure enough, that's one of the things that we're seeing being noticed on the Internet is people who do run their iPad all the way down, it takes a long time to charge it back up because those little chargers are 2.5 watts and 2 amps. And so, if you're charging a dead, an empty 43, to round it easily, watt hour battery with 2.5 watts, you do the math, and it's...

Leo: Actually the iPad adapter is 10 watts.

Steve: Okay, but still 2 amps, I guess.

Leo: Yeah.

Steve: I think it's only 2 amps. So it's going to take hours in order to charge it.

Leo: It's slow, yeah, yeah.

Steve: And that's what we are seeing. And then lastly, the last glitch that I just saw was the problem with some people's smart covers with the iPad 3 because the iPad 2 had magnet sensors that did not care about the polarity, the north-south polarity of the magnets. And the problem was, people who folded their original iPad 2 covers back sometimes had their iPad think that the cover was closed over the front of it rather than being all the way opened over the back of it. And so Apple fixed that little glitch by making the iPad 3's cover sensor magnets sensitive to north versus south so that it could tell whether the magnet was coming down from above or up from below.

But the problem is that some of the original covers and third-party covers that also take advantage of this, they had their magnets thrown in with no concern for their north-south orientation. So what that means is that some will work and some won't. It's just sort of luck of the draw. And apparently Apple is now taking back covers that don't work and exchanging them for ones that do because, when Apple fixed this problem, they had to start orienting the magnets in their covers correctly. So anyway, that's my iPad 3 trivia.

Leo: Yeah, yeah. I have noticed that on some of my covers. Although Apple smart covers, of course, are correct polarity.

Steve: Yup. In security news there's two things, both from one guy. I've got sort of a past friend, current friend, and hacker named Jeremy Collake, who I'm sure I've mentioned in podcasts before. He actually helped me with one part of the socket lock utility that I wrote. He's adept with drivers, and so he was able to quickly produce one component of that little gizmo when we were locking raw sockets back in those days.

Anyway, he discovered something - actually, this sort of follows nicely on last week's discussion about server configuration and security. He discovered that Apache servers by default have something called a "mod status module" installed and running. And so, for example, if you go to www.washingtonpost.com/server-status, that's a pseudo page generated by this status module which gives you a real-time snapshot into the server.

Now, the problem is that these Apache servers are all over the place. And you can see a list of the most recent URLs which have been served. And as we know, URLs often contain sensitive data. And these will be URLs, even if they were wrapped in SSL, this is once it gets to the server, after it's been decrypted, this server-status page will show you potentially confidential information.

And so anyway, Jeremy blogged about this. His blog is thepileof.blogspot.com, if anyone is curious. And, I mean, even Apache.org has their server-status page wide open. And I don't want to go into any other organizations that do, but some very sensitive organizations with running Apache server, you can see the IP addresses of the people who have been visiting and what URLs they clicked on. And just like, whoa. And as far as I know, Jeremy is the only person who has noticed this and talked about it. So he asked me if I would bring it to our listeners' attention. Anybody who is an Apache server admin, unless you're explicitly serving that server-status page to the outside world for some reason, you may want to lock that down. These things can happen, as we know.

And then the second point is many people have asked and been excited about the encrypted DNS service being offered by OpenDNS. And until actually I guess even now, the client side, in order to use encrypted DNS, you need to be making encrypted queries of the OpenDNS server. There wasn't a Windows client. But Jeremy found the one that was currently available, I think over on Google Code, and checked it out and tried it, and it looks like it works great. So his most recent blog entry, again at thepileof.blogspot.com, is how to use and configure the Windows client for issuing encrypted DNS.

And the reason this is of interest, I mean, I'm sure our listeners will understand this, we've all talked about the utility and growing need to encrypt our TCP connections. But there is no encryption of DNS. It's not available. It's not offered. It's not in the spec. There's no equivalent of making an encrypted DNS query. Which means anyone who's looking, for example, even at someone's encrypted traffic in an Open WiFi hotspot, will still see all their DNS queries, meaning that they know all the domains that they're looking up. And while this is not a huge problem, it's an information disclosure problem. And apparently there's enough interest that I've seen some people asking questions and tweeting me, asking me if I've seen and what I think about the OpenDNS encrypted DNS stuff. So it is available on Linux, Mac, and UNIX, and there is the Windows client that Jeremy says works just great that's now available also for download. So you can find out more about that at thepileof.blogspot.com.

And in just yesterday's news, actually I guess the last couple days, it came to light that the NSA - I don't know if you ran across this bit of news, Leo - is building a massive super-super-computer center in Utah with big cooling towers, huge water pumps to pump the water through the cooling towers to take the heat out of this place, its own energy generation system, all of the, I mean, this is going to be a top-secret NSA cyber-computation facility. And the concern was, naturally, among people who listen to the podcast, uh-oh, does that mean that the NSA will be developing the technology to crack state-of-the-art encryption?

And so I thought, well, okay, I'm still not worried because 256-bit AES is so already overkill stronger than we need. I mean, 64-bit is arguably strong. 128-bit, as we know, is not twice as strong with the bits being twice as long. It's, well, 64 bits is going to be twice 32 bits. So 32 bits we know is 4 billion. That means that 64 bits is going to be 16 billion billion. Which means that only 128 bits will be 256 billion billion billion billion. That's just 128 bits. And we keep going, multiplying for every one bit we add.

So I'm not worried about 256-bit AES encryption, which is now available. I would say that's what I would recommend. But what was really interesting is that the NSA has - apparently all kinds of different satellites, literally and figuratively, of the NSA are feeding into this center, and they have an amazing amount of storage. And one of the things that one of the articles that I saw about this mentioned was that they're very interested in decrypting the encrypted foreign communications from years past which was still using less strong encryption. So it's not so much that we have a concern today.

And what I loved about this was it brings up a really important lesson for us. And that is that, while our current encrypted data may not be crackable today, it may be crackable a hundred years from now. And the NSA has been storing encrypted communications globally for a long time and archiving it. Even though they couldn't crack it, they figured one of these days maybe we will be able to. So this facility that they're building is more designed at cracking the past than it is cracking now and the future because, frankly, now and the future is really, really strong. We've got seriously strong technology. But even a decade ago there was stuff they would dearly love to crack. And a decade ago our encryption was not nearly as strong as it is today.

So that's an interesting lesson, that there are just these massive stockpilers and archivers of global communications that they don't know what they say yet. But once this facility is in place and humming, probably literally, from miles away, and glowing at night, that's the project they're going to be on is bringing to bear insane computing power in order to peer back into encrypted data that is still opaque. Which I think is kind of cool.

There was a story that I didn't have a chance to plow into, another one of these Wi-Fi

Alliance things where somehow our cell phones are going to be authenticating to wireless hotspots. And so it's not clear. There's no technical information yet. I did dig as deep as I could, and all it was was a press release saying that, due to the proliferation of wireless hotspots, and of course the cellular carrier preference for moving their customers to ground-based WiFi rather than competing for limited cellular bandwidth, you can sort of see that the cellular providers would like to somehow get their customers moved over.

For example, the iPad, mine is AT&T and WiFi. And when I'm somewhere where there's a WiFi access point that I have acknowledged and logged into, the iPad will preferentially use that bandwidth over using cellular. So there's some movement afoot to make this official. And it's still premature. I'll keep my eye out for it. And if any of my Twitter followers see any more details, by all means give me a heads-up so that I can kind of look into it. But one hopes that whatever it is they do, they do correctly. And it would be neat, for example, if there were some sort of encrypted negotiation so that you were able to seamlessly encrypt your connection to a WiFi hotspot rather than, I don't know, join it in some fashion without the advantage of encryption. So that would be nice because your cellular connection is always encrypted; whereas, as we know, any use of WiFi hotspots is just not at all.

And then there was another story that I was unable to track any more details down. Maybe you ran across this, Leo. The RIAA and the MPAA were at a conference two days ago and saying that by June, like June 1st was the date that was quoted, so just a few months from now, ISPs are going to be watching their users' behavior on behalf of copyright holders and, for the first time ever, are implementing infrastructure, which is what it takes on the ISP's side, to start sending notifications to their customers if they see them downloading copyrighted material. First they get - they get a couple notices first, and a few strikes. And then your bandwidth gets throttled. And then ultimately you get disconnected.

So that's news, to have ISPs which to this point have just been blind bandwidth carriers of ours. I mean, they've been doing some shaping. We know that they've been caught trying to throttle things that they thought were using too much bandwidth. But they weren't doing any looking at our traffic. So this is a concern. Now, the good news is SSL is our friend because they can't filter and look into any of our SSL connections. And they can't proxy them unless they start making us accept a browser certificate, which will be the end of life as we know it. So none of that is apparently happening. But for people who are not using SSL, a few months from now the word is - and it's a whole lineup of, like, a bunch of the major ISPs are saying they're going to start being proactive.

Leo: This is that "six strikes" rule.

Steve: That's the one, yes.

Leo: AT&T, Verizon, Comcast, Cablevision, and Time Warner. It's a voluntary agreement. It's not a legal - it's not a law. But...

Steve: But why? I mean, like, we're their customer. Why are they serving the interests of the RIAA and the MPAA, who are doing everything they can to mess things up?

Leo: The article that I found is eight months old, so I don't know if this has changed. But at this point the ISPs say, "We will forward copyright notices to subscribers, but we won't turn over information about subscribers without a court order." It's a one-way street.

Steve: Yes. As far as I know, that is still the case, based on what I saw that talked about what was just being said two days ago. This would be ISP to their customer, not - and I didn't mean in any way to imply that data is going from the ISP back to the copyright holder.

Leo: What happens after five or six alerts, which is quite a few, is the ISPs have agreed to institute mitigation based on the copyright holder's request, which could include temporary reduction of Internet speeds (throttling), redirection to a landing page until the subscriber contacts the ISP to discuss the matter or responds to some educational information about copyright or other measures the ISP may deem necessary to help resolve the matter. This does not involve a disconnect at any point from Internet service. But throttling can be a pretty serious penalty.

Steve: Yeah. So now we come to the question, what constitutes copyrighted material? How is that determination made?

Leo: Well, it's a letter from the copyright holder. And I have to say, this system is very broken in a number of ways. For instance, YouTube, which has this DMCA takedown. I am now getting a notice on almost every show we post that we contain content from - and it's people we definitely don't contain content from, often Brazilian broadcasters, just strange. And I think what's happened is these people are gaming YouTube now so that they are saying, well, they're giving them something, and it sets up a - it's all automatic at YouTube's end. And what it does is it allows these guys to put ads into my content because YouTube gives you a choice. YouTube says, well, we'll take it down, or you could put an ad in, or you can offer to sell the content if it's a song or whatever. And so what these Brazilian television stations are doing, I think they're doing this as a gaming thing, but I haven't really done much digging, is putting ads in our content. So you're getting a false - I think they're getting a false copyright notice.

Steve: False, yeah.

Leo: And then getting the right to put an ad in our content. It's just appalling. And so you could see how this kind of copyright notification system can be absolutely abused.

Steve: Well, exactly, and that was my point, was that the ISPs are adding automated systems. So, for example, in order to send out notices and to count how many they've sent, and including to actually actively filter the queries that their customers are sending out to the Internet in order to retrieve what is believed to be copyrighted content, this is all new infrastructure. This is some serious equipment that the ISPs are for some reason installing and sticking in the circuit of their customers in order to offer this service, such as it is. But the question is, then, where is the list of what is copyrighted coming from? Their filters have to be driven by a blacklist of...

Leo: I don't think the ISPs are doing this. Isn't it the copyright holders who are notifying the ISPs?

Steve: But, like, what, on every file on the Internet? You see what I mean?

Leo: Here's what it says: "Copyright holders will scan the 'Net for infringement, grabbing suspect IP addresses from peer-to-peer filesharing networks." BitTorrent, if you're not smart, for instance, you can see who's sharing it. If you're smart, you just encrypt and that doesn't happen. So if they see your IP address participating, they'll then contact that ISP. By the way, this is better than the old procedure, which was they would ask the ISP for information, and the ISP might actually give it to them. Now the ISPs say, no, we're not going to give you any information. We'll notify them, and we'll do the six strikes thing. So I think it's just a formalized agreement about what will happen. But copyright holders have been doing this for ages. This is the only way they can do that.

Steve: No, see, but this is different. What you're talking about is what has been happening, where IPs were identified as being infringing, and then the ISPs could notify their customers. Now the ISP has some sort of blacklist of content, not of customers, but of content. So if one of their customers downloads that content, the ISP says, oh, that's copyrighted content. And so my question is, where does that list come from? That seems to be the real problem because the ISP has to be able to identify customer queries that are attempting to fetch copyrighted material from the Internet. Not by customer, but by the name of the material in the URL. And so there's got to be a list of that somehow that the ISP's filters trip on.

Leo: I don't see that on the original article. Here's an article from the Law & Disorder column a couple of days ago in Ars Technica, the "Copyright Alert System." Everything I've seen says that it is still incumbent upon, not the ISPs, but the copyright holders to notify the ISPs. So I'm trying to find that information. So you're saying that the ISPs are now running some sort of filtering.

Steve: My understanding was of that. Maybe that's not the case, though. Maybe it's just that they're maintaining the six strike counter and being more proactive in what they do with their customers. And there just isn't enough information yet about what this thing is. I did want to bring it up because it popped onto my radar, and I knew that our listeners...

Leo: Yeah, I don't blame you. Here's probably the one you saw, which was from the panel.

Steve: Yes.

Leo: "Each ISP has developed their infrastructure for automating the system. Start date for traffic.... Major labels monitor BitTorrent and peer-to-peer networks for copyright infringement, then report that infringement to ISPs." Oh, so it is the labels doing it. I really think that this agreement is the ISPs trying to get out from under this. They've set up a system, but it's so that they don't have to do anything more.

Steve: Well, and it does sound like it's a benefit to customers because the ISPs will not be turning over the customers' identities to then be brought up in these ridiculous lawsuits that we've covered.

Leo: Unless there's a subpoena. So these ridiculous lawsuits are all John Doe suits, which then the point is to get the court to go to the ISP, saying, okay, hand over that information.

Steve: And compel them.

Leo: And compel them. And I think that this is the ISPs saying, look, let's just do it this way, and let's not go to court. And frankly, I think the record industry and the motion picture industry are looking for a way to save face and to back down on all these John Doe court cases.

Steve: You know, I think that's right, too, Leo, because I do remember reading something about, like, a pro forma letter that said individuals at the account associated with the IP address of your account have downloaded copyrighted materials. So this probably represents an interface between the copyright holders who are still responsible for generating IP lists of misbehavior, and then those they turn over to the ISPs managing those IPs, and then the ISPs now are taking a new role in notifying the customers who have those IPs.

Leo: Right, exactly. And there is an appeal process. This is, I think, a way to avoid a three strikes law.

Steve: Yeah, good.

Leo: Which nobody wants except the copyright holders.

Steve: Right. I did want to just mention that I finished Book 13 of the Honor Harrington series. Actually there's probably one or two more coming. I'm glad that I read them. I have this - this whole new world that exists in me now, after 13 novels.

Leo: The Honorverse; right? Isn't that what they call it?

Steve: Yeah, the Honorverse. And it wrapped up in a nice place. I'm not chomping at the bit for 14 and 15. David can - I mean, 13 just came out, like this month, earlier this month. So it just happened. So I imagine I'll wait a year or two for 14, and then I'll read that; and 15, and I'll read that. There are other ancillary books, but I'm not going off and down those rat holes because I've had all I can handle with 13. And as I mentioned last week, I'm excited now to start reading more about nutrition, which is my current reading focus.

Leo: I got a very nice email from somebody who pointed me to Gary Taubes' blog post about that Harvard meat study, in which he blows it out of the water. And I had forgotten - I had read "Good Calories, Bad Calories," which is the book I know that you're taking as gospel. And he blows it out of the water there, too, these epidemiological studies...

Steve: Yes, that's the problem.

Leo: ...that our entire nutrition system is based on now.

Steve: Yes.

Leo: They've gained such currency. And it's really very simple. It's funny, I posed this question to my daughter, who has studied statistics. And she's actually taking a course at her university called "Citizen Science," which is about educating non-scientists in scientific thought so that they can more intelligently judge things like this.

Steve: Nice, nice.

Leo: And I said, "Abby, what's wrong with this study? They took 100,000 physicians, and they followed them for 18 years, giving them the questionnaire about what they ate, and then checked their mortality. And according to this study, they had a 20 percent higher chance of dying prematurely if they ate meat. What's wrong with that study?" And she cut through it right away. I was really impressed. She said, "Well, the problem is, for those past 18 years, we've been told that eating meat is bad for you. So the core cohort of physicians who are eating less or no meat are probably also people who take better care of themselves in other ways. So it's a self-fulfilling prophecy. Since we've been told eating meat is bad..."

Steve: Yeah, very good, Abby.

Leo: "...people who don't eat meat are going to be more likely to take good care of themselves. Correlation does not prove causality."

Steve: Exactly.

Leo: A-plus, Abby Laporte, you get a checkmark on your quiz.

Steve: Yep. In fact, you and I may remember - you may remember that you and I, a long time ago, we were talking about the problem of tracking down causality. And I think we used the analogy of an alien landing in New York and noticing that, when the rain came down, everyone opened their umbrellas. And that unless you under...

Leo: Umbrellas cause rain.

Steve: Exactly. Unless you understood what the actual mechanism was, you could easily draw the wrong conclusion.

Leo: And that's why we have scientific process. That's why the scientific method exists. As Taubes points this out, okay, now you have a theory. That's all you have. Now you actually have to test, and this is what's so difficult, well, does meat actually cause mortality? We maybe have a theory based on this epidemiological study, but you have no information. So now you create a scientific double-blind study, and then we'll know. But of course nobody's doing double-blind studies with meat eaters.

Steve: Well, actually the problem is that dietary studies are notoriously difficult because you don't know about the level of compliance that the people actually have to the diet, and we're talking about problems that manifest over decades. And that's, I mean, that's the real problem is...

Leo: It's very difficult, yeah.

Steve: ...because we're really not good about anything that takes that period of time.

Leo: He says, and he does quote in the book, both books, "Good Calories, Bad Calories" and his kind of more popularized...

Steve: "Why We Get Fat."

Leo: ..."Why We Get Fat," he talks about a study that was done on all the popular diets and those who followed closely each diet and the prognosis for each. And oddly enough, counterintuitively, Atkins won, against the Zone Diet, against the Pritikin Diet, Dean Ornish's diet. And he says that's the only evidence we have. So anyway, you were right. I brought it up unthinking. Fortunately my daughter has a better head than I do, and I read the blog post from Gary Taubes, which was just well done.

Steve: Well, and in fact at the end of the first part of "Good Calories, Bad Calories" - and I'm going to start that from the beginning because I took a break from Honor Harrington just because this sort of came on my radar, and it immediately captured my attention. The book I'm reading currently is one on nutritional anthropology, which is really interesting. But the point is, at the end of the first section where Gary talks about how it is that our society came to believe that fat is bad for us, that a low-fat, higher carb diet is heart smart or heart healthy, it turns out that there's one guy that's responsible for this, and he doctored his data. Ancel Keys is his name, since no longer with us. And at the end of that, in a beautiful, short paragraph, Gary explains the fundamental impossibility of using epidemiological processes to determine dietary outcomes. And when I encounter that again I'll share it because it was very short, and it was just spot-on.

Leo: Thank you.

Steve: So, interesting stuff. My news is that I've moved to Firefox v11.

Leo: Wow, Steven.

Steve: From 3 to 11.

Leo: That's quite a jump.

Steve: They have solved the memory problems. So I wanted to let everyone know that my big complaint was that 8 and 9 and 10 kept saying, oh, we're better about memory, we're better about memory. It's solved now. You close pages, and you get back the memory that those pages were taking, which is the first version of Firefox since they broke it a long time ago that that's been true for. And really what was my final motivation was that I wanted to be able to turn on SPDY in Firefox, which is available in v11. You need to go to, using v11, because it's not on by default, you go to about:config. You put about:config in the URL. That brings up a pseudo config page. Then, in the search term up at the top, the search field, just type in "SPDY," and that'll find you that subset of configuration settings involving the SPDY, the so-called SPDY protocol, and it'll be turned off by default. Just double-click it and flip it back on, or flip it on. And you then have SPDY support in Firefox. And there is an add-on which will show when you are using SPDY connections as you surf around the web.

But also, while you're in that about:config, if you type in "cookie," you'll be taken to a bunch of lines involving cookies. And there's an interesting setting that I did some research on after I found it. It's network.cookie.thirdparty.sessionOnly. And you can turn that on. And people who are concerned about tracking and would like more prevention of that, sort of just generically, this makes third-party cookies session-only, so that when you close the page for a website, the third-party cookies that may have been transacted and in some cases have to be in order to use things - like some Facebook apps require third-party cookies. Someone told me that some Google services now won't function because Google is using their own third-party domains to glue their things together, so you have to have third-party cookies enabled. So you can make them just session-only. You close the page, and they're never written to disk, and they just go away. So that's another nice little feature that's probably been in Firefox for a while. But I just happened to put in "cookie." I searched for that when I was searching for SPDY because I wanted to turn that on.

And lastly, just complete randomness, I wanted to find good wallpaper for my super high-density, high-definition screen on my iPad 3. And I was reminded of one of my favorite websites, I became a lifelong member years ago, called DigitalBlasphemy.com. And this is a guy who for a decade has been using a huge array of digital artwork creation tools to generate really beautiful scenes, scenes of nature, castles in the background with lakes and trees in front of them and snow in the distance, complete abstracts, neurons firing, and all of this is available at very high resolution, I mean, like full desktop, large monitor resolution. He also has dual and triple monitor versions so that you can have your wallpaper stretch out over a three-monitor setup, rather than have it repeat, have it all be coherent.

So anyway, I just wanted to tip our listeners off to DigitalBlasphemy.com. It's a great site. There's stuff that's available for free. You can join for a year. I joined for life because I always wanted access to it, and it's been worthwhile. He keeps generating new stuff every year. So that lifetime subscription ended up being useful. And I like to support someone who's doing that stuff. I mean really beautiful, beautiful artwork.

Leo: Yeah, he does great stuff, yeah.

Steve: And I heard from a listener, Ken Harthun, who wrote to me on the 19th of February: "SpinRite saves a student's laptop." He said, "Steve, I'm a loyal listener of Security Now!, having listened to every single episode. That first episode was only 18 minutes and left me wanting more." Well, we've taken care of that.

Leo: Was it that short? Wow.

Steve: Wow. And that was your original concept, Leo, was just to do sort of a check-in on the week. It's like, okay, well, that didn't last long. And it's funny, too, because I remember Elaine quoting me for transcription, didn't sound like it was going to be very expensive, either.

Leo: No, sorry about that. Whoops.

Steve: Oh, it's been worthwhile, and I haven't looked back.

Leo: Thank you.

Steve: So he said, "Today's episode was a little over two hours and still left me wanting more. You are often the source and inspiration for my Security Corner blog posts over at IT Knowledge Exchange. So a big geek thank you to you and Leo. Please continue." He says, "I first used SpinRite in 1999 - it was v5.0 - to recover a floppy disk that had been corrupted. Since that day I've insisted that wherever I worked, the IT department agreed to make SpinRite available to me should the need arise, and too often it has. In my private service world, I always insist that, if SpinRite recovers the drive for my client, that my client purchase a copy. Needless to say, there have been a few sales as a result."

Leo: That's good idea. That's a good way to do it.

Steve: I have no problem with that, yeah. He says, "I have my own copy, of course, and last summer I insisted that my new employer, Antonelli College, where I am the network administrator, purchase a site license. Well, that's a good thing because last week it saved one of their students' laptops and all of her interior design coursework. Windows was throwing all kinds of errors. The wireless wouldn't connect. She gave me a list of seemingly random errors that didn't seem to make a whole lot of sense. But they pointed toward a hard drive failure. I was about to attempt to backup the data and restore the system when it just completely locked up, and I had to force a shutdown with the power button. On restart it just hung at the starting Windows screen and would go no further. I could hear the drive thrashing about. Not good.

"Enter SpinRite. I booted up from my thumb drive and ran it at Level 2. After a couple of hours SpinRite reported that it was finished, though no errors or bad sectors were found," which of course is a story we've heard many times. And I've explained why that doesn't mean SpinRite didn't do anything. He says, "On reboot, the system came right up, faster than ever, connected to the wireless, and immediately began downloading updates. I completed the updates, ran a few tests, and pronounced the patient healthy. Needless to say, the student was ecstatic. And thanks to SpinRite, I did my part to provide a 'superior student experience.'" He says, "Part of our vision statement for the campus." He said, "Steve, SpinRite is absolutely the best hard drive maintenance and recovery utility on the planet, and maybe in the universe. It's worth 10 times the price you charge for it. Thanks for all you do. Ken Harthun."

And he said, "P.S.: I've never had a hard drive failure, and I attribute that to my using SpinRite on my own systems on a regular basis." And of course we understand also why it is a good preventive maintenance utility. Running it on a drive, even a quick Level 1, shows the drive where it's got problems developing that it's able to correct before they get critical.

Leo: All right, Steve. What is buffer bloat?

Steve: Okay. Our listeners who are live can start something up in the background while we're talking, so that when we get down to where we're talking about what this is, they may have some results.

Leo: I've done it already, and I have my results.

Steve: Cool. This is Episode 345 of Security Now!, so I have done what I've done before, which is create a bit.ly link with the episode number, bit.ly/sn345.

Leo: Bit.ly/sn345.

Steve: Yup. And I made it both uppercase "SN" and lowercase "sn" so that it didn't matter which people used this time. That will take you to what's called the "Netalyzr," which has been put together by the ICSI at Berkeley.edu, at UC Berkeley, at the

International Computer Science Institute. It is a Java applet. And I tweeted this link in preparation for the podcast earlier this morning and got some of our listeners who sent back, "Steve, that's Java. What's happened to you? You're running Java?" It's like, yes, yes, we have no choice. And a couple of people said, well, I'll install it just for this because it sounds really interesting, but then I'm uninstalling it." It's like, okay, fine. I mean, we're losing this battle against scripting. So I'm accepting that, that scripting is the future.

The beauty of this application in Java, this is a stunning piece of work. And when I'm looking at what they can do in Java, I'm thinking, ooh, I could do some amazing stuff, which has the benefit of being platform agnostic, which is really important, being able to do low-level, packet-level work, and writing it once, and being able to run it across platforms. Of course it won't run on an iPad. You need to have something that'll run Java, and the iPad won't, in the same way that it won't run Flash. But you do get both Mac and PC. So, yes. This is a Java applet. Let it run. It takes a few minutes, maybe five minutes. And there is one of the things it does is measure the size of your buffers, that is, under load, the latency that the buffering between you and them has.

Leo: By the way, we have killed the site. So people, go later, don't go - everybody went all at the same time.

Steve: Oh. Oh, you mean we killed Berkeley?

Leo: Oh, yeah.

Steve: Oh.

Leo: Berkeley.edu is down, my friends.

Steve: Actually, when I tweeted it that happened. And I even tweeted about OneID, like a week or two ago, and they went down.

Leo: People don't build sites, you know this, they don't build sites for the peak. They build them for the average. It's too expensive.

Steve: You cannot afford to build them for the peak, yes.

Leo: So when we send 1,000 or 2,000 or 5,000 people to a site, of course it's going to - most sites will go down.

Steve: Bye-bye. Okay. Let's step back. We've in the past created a perfect foundation of knowledge about the way the Internet works for understanding the problem with buffering. We understand that, instead of a modem connection, where you actually have the equivalent of wires from the sender to the receiver, and you know exactly what the bandwidth is because that's the baud rate of the modem or the bandwidth of your point-to-point hardwired connection, that's all gone now. And I remember describing how, I mean, what a conceptual leap it was in the minds of the original designers, the concept that you could create virtual connections, not actual physical connections, but the equivalent of a virtual connection, with the agreement between endpoints that they were connected by maintaining some state information, some knowledge at each end about the condition and the history of their connection, and then having them just launch packets of data towards each other which the intervening Internet of routers would arrange to get to the other end.

And I've talked about router buffers before in this context, where you think of a router as like a star, like a hub with a bunch of connections coming out of it, going north, south, east, west, and other compass directions, and packets are coming in on various of those connections to interfaces on the router. Then they go into the core, routing core of the router, which examines the IP address at the front of the packet and decides, using its routing table, which is the best interface to send the packet back out on.

So the timing of all of this is uncertain. These packets are arriving on all these different wires coming into the router whenever they want to, asynchronously. And it's having to sort of shuffle them around, look at them, and then send them back out. But if by chance a bunch of packets came in on three lines that all wanted to go out on a fourth line, and assuming for a second that these lines were all the same speed, well, if three came in, or if a bunch of packets came in from three lines, they can't all go out at once. They have to be lined up.

So interfaces in routers have buffers. They have a staging area where packets can be placed to sort of deal with these little brief events, sort of just the need to deal with the fact, as a consequence of this autonomous routing, where we've just got packets flying all over in every direction, the consequence of that is we need some buffer. We need to deal with the possibility that there'll be moments where a connection is saturated, and where it's not so saturated that we have to throw things away because, if we had no buffer, then we would be over-discarding packets.

But one of the other weird consequences of this multilink links between routers, we've got links, our WiFi from our laptop on the couch to our WiFi router. Then there's a link to maybe our DSL modem or our cable modem. Then there's the cable link to the ISP's network. Then there's multiple routes through the ISP. Then there's their link to the Internet. Now we're on, finally, after all those links, we're on the core of the Internet. So then we've got multiple major Tier 1 providers, like Level 3, for example, and others that are major carriers. And they get the traffic over to another ISP where, through many links, it gets back to its destination.

So all of these links, probably almost without exception, they're running at different speeds. There's big hefty multi-gig fiber links. There's GigE Ethernet links. There's 100Base-T links. There's who knows what between the cable modem or what the cable modem is actually doing in terms of its connection. Then there's a link between it and your WiFi router, and then there's your WiFi link. And we know that WiFi, actual WiFi performance varies greatly, depending upon the signal strength and multipath interference and various things that the WiFi system is doing in order to make it work. So, I mean, it's just when you stand back and you look at this, it's amazing, frankly, that any data gets anywhere at all.

But what's significant is that there is no way to know what the bandwidth is because of all of these links and routers. That is, it's one thing to talk about buffering. Think about buffering as sort of short-term overage handling, where just in an instant three packets can all want to go out one wire, and they've just got to line up a little bit. But if over time many input feeds were saturated that all wanted to go to one output feed that was no faster than the inputs, then more data would be coming into that router than it was possible for it to send out. And so it would have no choice but to drop packets. Its buffer would fill, and then no more could get in. And so it just wouldn't happen. They keep coming in on the wire, but there's no where for it to put them. It can't get rid of them fast enough, so it's got to just discard them. And we're discussed this. That's the way - that's the genius of the original designers was that they said, okay.

Leo: Throw it out.

Steve: Not all packets have to get there.

Leo: Right.

Steve: They made peace with that, which had to have given them some sleepless nights. But they said, okay, we're just going to make, I mean, that's like - that's the tradeoff of a packet-switching network is we can't guarantee that you're going to be able to send as much as you want because other people could be competing with you on the same thoroughfare, and just you can't get there. So you'll have to back off. You'll have to throttle back.

And that is the genius of the TCP protocol. Remember that we've talked about the way TCP works. It starts off slowly, the so-called TCP slow start. When you initiate a connection to a remote server, your computer doesn't know how much bandwidth you've got. It doesn't know how fast it can send the data. So it starts sending it and hopes for the best. And as time goes by, it sends the data faster and faster and faster. What it's trying to do, it's trying to sense when the link gets saturated. And because of all these hops and all these links that may be running at different speeds and may have differing levels of congestion, we don't know where we're going to have a problem. But at some point along the way, there will be a situation where packets are lost. They may be lost due to a momentary surge, where there's competing traffic, or they could be lost at any point.

And this is a key concept. At any point where the bandwidth drops, any point where you go from a high-bandwidth flow to a reduced bandwidth, you're going to have a problem because, up until then, you've been able to send packets at high speed. As soon as you drop to lower bandwidth, as we've seen, there will probably be a buffer of some sort there. It'll be some device which is doing the best job it can. But you're just giving it more than it can send, so it has no choice but to discard something. So the brilliance of TCP is that it senses the loss of packets when the packets it's sending are not being acknowledged. When it fails to get acknowledgment, it assumes packet loss, so it backs off. It slows down in order to sort of adjust to having hit the ceiling.

Now, it doesn't know whether that was a fixed bandwidth limit that it hit, and so that it should just stay where it is, or that it could have equally been a burst of congestion somewhere along the way that's gone now. So it always creeps back up. And so what TCP is always doing is sort of riding just under the ceiling of what it's able to establish. It's always trying to push a little harder. And when it gets the news back that, whoops, packets apparently have been dropped because it's not getting acknowledgments of their receipt from the far end, then it takes that as, oops, okay, I found the ceiling again, back off a little bit, and then it begins to creep forward.

So that's been the solution. But what that depends upon is low latency, that is, that depends upon, by its nature, that soon after it's going too quickly, it receives notice that it's not. And it was trying to think, Leo, of the best analogy, and I got a great one that everyone can relate to. And we've all seen on newscasts how painful it is for two people to talk over a satellite delay which is substantial.

Leo: Yeah. You see it on CNN all the time, this long lag.

Steve: Yes. Or, bless his heart, Chris Matthews cannot shut up. And it's so...

Leo: On MSNBC, yeah. You have to just stop talking. That's the only way to handle it.

Steve: Yes. Well, and what he does is worse because - and I've watched this over and over and over. He'll be talking to somebody in Iraq, and the person is sitting there, patiently waiting, because Chris loves the sound of his own voice. And so Chris finally ends with a question. And then, just as the other person hears the end of it, Chris thinks of a better question.

Leo: Yeah.

Steve: And so asks him, he sort of, like, amends that question and goes on. And so you see the other guy beginning to respond, and then he hears Chris change his mind about the question. And it's just like, oh, my god.

Leo: Painful, yeah, painful.

Steve: Anyway, so the point is, what is that? That is two people, two entities, trying to interact in a real-time fashion in the face of delay, that latency. And you just can't do it. It creates a problem. And so what has happened is, over time, all of the specifications have been laid out beautifully. If you look at the RFCs, everything I've talked about is spelled out in detail. Nowhere, nowhere does it talk about the size of the buffers.

Leo: Yeah. Unfortunately.

Steve: Never comes up. Never did come up. Now, it used to be that RAM was expensive. So buffers were small because even Cisco making BigIron routers, they were trying to get as much profit margin as they could, and hardware was expensive 15 years ago, 20 years ago. So the buffers were small. And also these were the engineers who designed the Internet. They knew that packets were supposed to be dropped. And I've said it in our original tutorial series on How the Internet Works, that's the genius of packet routing is, oh, well, packet got dropped. We couldn't get it there. And all the protocols have been designed with that in mind. TCP sets its speed assuming that, immediately after the bandwidth is hit, it will get a notification that lets it back off.

But now what happened? Everything got cheap. Chips get big. RAM, huge amounts of RAM is built into the ARM processors, just because why not? We keep making the die sizes larger, and the details we're able to imprint on the chips are becoming ever smaller. We're able to increase the transistor count dramatically. So being able to say, oh, this thing's got 256MB of RAM, well, it doesn't cost anything. And so the router vendors, really, who are not the Internet engineers that founded Cisco and Jupiter and the major backbone providers of the Internet, they're thinking, hey, we can have larger buffers. And then packets won't get dropped. Won't that be wonderful?

Well, turns out the answer is no, that's really bad because what happens is, it is often at the client router, at the end-user router, that we have the biggest drop in bandwidth, that is, where we go from large bandwidth pipes to a restricted bandwidth. And if this router has large buffers, and now we're talking, I mean, they could be, and in fact in some they are, megabytes because the RAM is there, it's free, it's on the chip. And it would take a certain amount of self-control for the designer to say I'm only going to use 10K when he's got a meg, and when he thinks, oh, not understanding this, thinking that dropping packets is bad. Turns out, no, dropping packets is really important. That's the only way that TCP knows how fast it can go. And if you allow TCP to keep going faster, it will fill this huge buffer. And only when this huge buffer is full will packets start getting dropped.

The problem is that, remember that it's the acknowledgment of unreceived packets which is - I'm sorry. It's the non-acknowledgment of packets that were not received that tells TCP to stop. So what happens? TCP keeps filling this buffer, which then goes into a - it drops in bandwidth, there's a constraint, before it gets to the endpoint. Well, the endpoint is happy because it's still getting data from the buffer. So it's continuing to acknowledge the correct receipt of the packets incoming, which continues to encourage the sender to, not only keep sending them, but to send them faster. So this buffer continues to fill. It starts even filling up faster now because, due to its depth, the recipient is still getting data from this big buffer and acknowledging, so the sender keeps cranking it forward. So what ends up happening is we end up, as a consequence of this delay, we end up delaying the news to the sender that a long time ago we hit our bandwidth limit.

Well, what TCP does is it backs off by a percentage. It assumes timely notification. So if you put a big buffer there, it backs off by a percentage of what it was sending. But the buffer has been so big that it has gone way beyond the recipient's actual ability to receive. So even backing off a bit doesn't solve the problem. It's still going too fast. And then it backs off again, and it's still going too fast; and again, and it's still going too fast. So you end up with this big problem caused by a buffer which is too deep.

Now, the other thing that happens is the phenomenon that people see at home. And this is what got Jim Gettys, who is the person a year and a half ago who got onto this problem, he noticed that interactive gaming came to a standstill when he was downloading a file. Or, no, I think he was uploading a file, actually. So the idea was that he was uploading a file, and even the reverse direction had a problem. Well, the reason that happens is notice that we've got data for TCP. Acknowledgments have to go back also. So if you've got a buffer, a large meg buffer, which is full to the brim, carrying one flow of traffic, then what's happened is you've introduced a delay. You may not care if the movie you're downloading to watch later isn't - you don't care about the real-time performance of that.

But you care about the real-time performance of web surfing, where inherently you're getting a page. That page comes in, and you're wanting to set up a whole bunch of new connections to all the other places that this page needs to be built from. And so the web page is highly interactive. But if you've got your router buffer, an overly large router which has been allowed to fill up with traffic from a download going on, suddenly now other traffic is stuck at the end of it, if it can even get in. It may be discarded prematurely because this buffer is full with someone else's work.

And even if it wasn't, we require - the assumption on the Internet is a roundtrip time on the order of 100ms. 100ms is, out and back, is sort of - that was the target that these protocols were designed around. And maybe it's 200, maybe it's 150, 156. I'm looking at my own roundtrip, and I'm seeing, oh, it looks like about 60ms between here and GRC. And I'm actually going up to Northern California before I jump onto Level 3 and come back to GRC's servers. And I see a worse case of about 158ms. And that's what ping shows you when you ping something. So you could use the ping command to ping Google or ping Microsoft or ping Yahoo! and get a sense for what your roundtrip time is. But it's relatively fast.

And web surfing, interactive use depends upon that kind of performance, that kind of speed. So if we have a large buffer in our router, as long as it's empty, we're fine. We'll get good interactive performance. But as soon as it fills up, if it's ever allowed to fill up, that large buffer equals delay if it's full. And then all, I mean, then, quote, "The Internet is slow" is what everyone else in the family starts saying, even if there's, like, ample bandwidth. You could have a 100Mb down and 1Mb up, yet if you saturate that 1Mb up, sending a file out, for example, nothing can come back because the protocols have to have acknowledgments get back in a timely fashion, and even saturating your outbound buffer keeps your incoming data from being able to be acknowledged.

So this problem we got into unintentionally with router manufacturers just sort of thinking they were doing the right thing, turns out to be really more trouble than it's worth. Notice that that big file you're sending cannot get to you any faster than the slowest bandwidth link. It can't. Nothing can squeeze it through. So having that big buffer sitting there, trying to squeeze it through, doesn't get it there any faster. If the buffer were only 10K, so that it's only eight or nine packets, then that doesn't mean it's going to go slower. What that means is that little buffer will overflow immediately, and an acknowledgment will be sent back telling the sender back off a little bit. And then the buffer will clear up.

But also that little buffer will allow all the other things going on in the household to stay interactive, even when a big file is being downloaded. And that's the key. These large buffers allow large download traffic to block interactivity. And in many situations that's a deal killer. You don't want that. And it doesn't help the bigger traffic to go faster. Due to this weirdness of the way packets are moved around the Internet, it's not going to go faster than it needs to through the slowest link you've got. And you're much better served only buffering just enough to deal with transience. And it doesn't make any sense to buffer larger than that because it breaks our signaling.

Okay. So there is this Netalyzr. I'm sure everyone listening is now wondering, oh, my god, what's the situation with my network?

Leo: I've run it here and at home already. It's very curious. This is a neat thing. I guess this is something they're doing at Berkeley as part of a large-scale study. So not only is it a great service for us, but by using it you're helping them collect data about networks in general.

Steve: Yes, exactly. They do some aggregate recordkeeping by IP, so they're able to see that these ISPs are doing this and those ISPs are doing that. And they also collect, without IP, just overall general operation. So I ran it on myself. And remember that I've got a pair of T1 trunks. They each go at 1.54Mb, and they're bonded together, so I get the sum of their bandwidth. So sure enough, this thing said, for me, "Network Bandwidth: Upload 2.9Mb/sec; Download 2.9Mb/sec."

Leo: As you'd expect.

Steve: Exactly. I mean, I am very impressed that they just nailed that. And they said, "Your Uplink: We measured your uplink's sending bandwidth at 2.9Mb/sec." It says, "This level of bandwidth works well for many users. During this test, the applet observed 1,551 reordered packets." Now, that's actually a high number, but that's a consequence of the fact that I have two T1s. So normally you wouldn't see that high a reordering count, but it's because my two T1s are - packets are going across either one, and they might be coming out in a different sequence.

Then they said, "Your download link: We measured your download link's receiving bandwidth at 2.9Mb/sec. This level of bandwidth works well for many users. During this test the applet observed 696 reordered packets." Now, network...

Leo: That sounds like a lot.

Steve: Oh, it is, because of my two T1s. But again...

Leo: So this is why your Skype sucks, by the way. Well, it may not be the only reason because - keep going.

Steve: Okay. "Network buffer measurements: Uplink 940ms."

Leo: That's a lot.

Steve: Well, but remember, only if the buffer's full. So, yes, uplink is almost a second, 940ms. That is a large uplink buffer. And then downlink 370ms.

Leo: That's not so bad.

Steve: So they say, "We estimate your uplink as having 940ms of buffering. This level can in some situations prove somewhat high, and you may experience degraded performance when performing interactive tasks such as web surfing while simultaneously conducting large uploads. Real-time applications, such as games or audio chat, may also work poorly when conducting large uploads at the same time."

Leo: So it's okay as long as you're not doing other things?

Steve: Exactly, yes.

Leo: Okay.

Steve: It's only, see, the buffer only introduces a delay when it's full. If the buffer is just cruising along, packets come in, one or two, and they immediately leave, then it's not introducing any delay.

Leo: We do have some of the worst Skype from you, and it really should be the best. And I do suspect, now, is that your router buffer, that delay? Or is that a T1 artifact?

Steve: I don't know.

Leo: Let me look at mine. We have EFM, Ethernet to the First Mile, on this computer. And it's showing network buffer uplink 110ms, very low.

Steve: Nice.

Leo: And then it doesn't even - it says downlink is good. It's not giving me any details on that. I don't know why. "We were not able to produce enough traffic to load the downlink buffer, or the downlink buffer is particularly small," which was what we would like; right?

Steve: Yeah. Now, and that exactly speaks to my point. The way they were able to determine my uplink bandwidth was by specifically generating enough traffic to overflow it. And that's what it took was like figuring out how much traffic, generating enough traffic to overflow it. Because lesser traffic than that won't - the buffering is not a problem.

Leo: Now, at home I'm on Comcast. This is my home computer. And it's running an Apple router. And it's considerably worse. The speeds are better. The upload is 5Mb, download 20Mb, greater than 20Mb. But the uplink is 260ms, and the downlink is 98ms.

Steve: Well, it's still good compared to me.

Leo: Yeah. What router do you use?

Steve: I have a Cisco 3400. I mean, I've got a BigIron router. I mean, not a Cisco plastic box, a Cisco...

Leo: No, no, no, the high-end one, yeah.

Steve: Yeah.

Leo: Isn't that interesting. But your numbers are - wouldn't you say they're less than optimal?

Steve: I'll do some poking around. Well, I just ran this an hour ago.

Leo: I think this is fascinating.

Steve: Yeah.

Leo: So you look at the network access link properties; right? That's the section that you want to look at. Is that right?

Steve: Yes. And it's network buffer measurements is what it says.

Leo: And is that directly related to the router, or could there be other...

Steve: Well, see, yes, that's the problem. And that's why I'm saying I don't know yet. I haven't had a chance to research this. Because this is buffering somewhere between me and them.

Leo: Right.

Steve: So it doesn't - it's not necessarily my buffers. It could be Cogent, my T1 provider could have their system misconfigured. And, see, that's the other reason. I don't know why these are different, why uplink and downlink are different because my router is doing nothing. It's just sitting here directly sending stuff out my T1 lines. I see no reason that there should be anything asymmetric. Notice that my bandwidth was exactly 2.9 in each direction. So it may not be me that's doing the buffering. It could be Cogent that's doing the buffering. And so it's not within my control.

Now, you can imagine, buffering is important enough that a lot of thought has gone into this. The typical brain-dead buffer is a simple FIFO that we've talked about, a first-in, first-out. And so the overflow behavior of a FIFO buffer is called "tail drop," meaning that it just drops the packet from the tail of the buffer. It no more will fit in, so it discards it. But because a lot of work has gone into this, engineers have said, okay, how can we - yes, losing packets, dropping packets is the nature of the Internet. It's going to happen. How can we make it smarter? For example, how can we be more fair about competing traffic? One high-bandwidth user should not be allowed to fill and saturate the buffer because then an interactive user who would like to have - doesn't need that much bandwidth, and if we could just sneak him in a little bit, then he could be happy with his low-bandwidth interactivity while the big transfer goes on in the background.

The problem is that requires extreme knowledge of the nature of the flows, and routers don't have that. Routers just see packets coming in, and they go, okay, fine, and try and send it out the best direction it can. But a technology was developed called Random Early Detection, RED. What random early detection does is, as the buffer is beginning to fill, not once it's filled and we have to drop things off the tail, but as it fills, the router increases the statistical likelihood of discarding a packet, even that it has room for. It just says, you know, the buffers are beginning to fill up. Let's just toss this one out because tossing them out, as we know, is a healthy thing to do on the Internet. And as the buffer continues to fill, it increases the likelihood of tossing packets out.

And what this means is that, if somebody was greedy with their particular flow, the likelihood of their packets being tossed out statistically is greater than the likelihood of somebody who's not using that many packets having theirs tossed out. So it tends to throttle the people who have more packets in the buffer and not so much those that don't have that many packets in the buffer, the idea being that theoretically you never get to a point of actually saturating the entire buffer because you exponentially increase the probability of discarding packets as it continues to fill. And the beauty of that is, for example, that allows TCP to get an early notice.

Ooh, and I forgot about one other horrible thing that happens. If you've got - remember when I was talking about how a big buffer gets full, and then TCP keeps sending and increasing its speed because it doesn't know any better, because the receiver is still getting valid packets and acknowledging them from this big buffer, one other thing that happens is called TCP Global Synchronization. If multiple TCP flows are going through, then what happens is all of their traffic begins to stall at the same time, but none of them get the notice. They all get over-ramped. Then finally they all shut down.

And what can happen is, as a consequence of this, is all of the TCP connections can essentially synchronize. So you'd like them to be hitting their ceilings at different times so that they're backing off and sort or scaling and sort of cohabitating nicely. But if you end up with a big buffer, the phenomenon of the way TCP backs off is they can end up falling into synch. And this has been something that's been seen in routers on the Internet, where there's like this surge, and then stop, and surge, and then stop. And you sort of get into this oscillating positive feedback phenomenon, which is really bad. Because, I mean, it's just like - it starts to really break the Internet.

So the good news is that attention is being paid to this. There isn't - this hasn't really - I don't think it's reached critical mass yet, where router manufacturers are acknowledging it. But as you'd expect, the OpenWrt people are. There is a site called BufferBloat.net where there's been a concentration of work. There is a variant of the OpenWrt project called Cero, CeroWrt, which is now at Beta 2. And they're only supporting one of the more popular, Netgear, I think it is, I want to say N600, but I'm not sure. That's just from memory. I didn't write that down. But so we are beginning to see some firmware for OpenWrt-class routers, which they're using the latest Linux kernel, I think it's 3.3. Typically routers, I think, are back on 2.6.something for their firmware. This project is using the latest Linux kernel because Linux is experimenting with the buffer bloat problem. And in the 3.3 kernel they've got something called BQL, which is a Byte Queue Limit. They limit the number of bytes in the queue rather than the number of packets using a rather sophisticated strategy and something called SFQRED, which is Stochastic Fair...

Leo: Stochastic.

Steve: Yes, Stochastic Fair Queuing Random Early Drop.

Leo: Oh, that's an acronym. Acronym.

Steve: Yes. So anyway, that's where we stand. I expect that, in the future, in the advanced configuration pages of the better SOHO routers, we will see configuration settings that allow smart homeowners to introduce smarter buffer management and probably manually reduce the buffer size. I read an interesting dialogue among these engineers that were explaining that there's a problem, and that is that, if a router manufacturer deliberately used smaller buffers, then competitors with large buffers could claim that the better router was inferior because it dropped more packets and had smaller buffers. And unwitting users would go, oh, well, that sounds like I don't want that router, when in fact it would give you much better performance...

Leo: Right, right.

Steve: ...on your network.

Leo: Well, I'm glad to know that it's software configurable, I mean, in other words you could use a custom firmware and fix it. It's not that - even though the hardware has the RAM, you can reduce it.

Steve: Oh, yeah. You just tell it don't use so much. And in fact it turns out that there are even device drivers, this Linux 3.3 has improved device drivers because, again, RAM got so cheap, and there was so much RAM in our PCs, that our own network adapters have over - have too much buffers down in the kernel, down in the driver.

Leo: Wow. Geez Louise.

Steve: Because it's like, well, we got RAM, we don't want to throw these packets away, when in fact you'd like to. And I did see one interesting comment in a dialogue where someone said, well, there's a problem with sending out lots of little bursts, which is what you have to do if you have small buffers, because for power conservation, having a large transmit queue allows the processor, like in a smartphone, to put a whole bunch of data into the queue and then sleep itself, shut itself down so that a lower power portion of the chip is able to go and send off that data, and the CPU is not consuming so much power. Whereas, if you had much smaller buffers, it would have to be constantly waking up and shutting down much more rapidly. And so as a percentage you end up being alive longer than if you were able to shut down for a long period of time. So that's, I mean, there's a lot to be determined yet. But that's the story of buffer bloat. And...

Leo: What is the optimal buffer size? Can you say that?

Steve: And that's just it, there isn't one. It's so confounding because it's a function of roundtrip time and bandwidth and speed and usage characteristics. There just - there is no optimum. But what's happened is we know that too big is really bad, and too small - you want enough to deal with transience, yet you still want the total roundtrip time - see, this is it. If you just do a ping, a ping will give you your no-buffer-delay roundtrip time.

Leo: Right, because it's such a small amount of data.

Steve: Yes. So do this. Ping Google and see what that is. Then start downloading a podcast from TWiT.tv and ping again. Oh, and you also have to wait a bit. You have to wait for the buffer to fill. So wait a while. Or just go ahead and start pinging. And what you will probably...

Leo: Watch it go down, yeah, yeah.

Steve: Yes, exactly. And so you will begin to see the roundtrip time increasing, not because you're getting further away from Google, but because the buffers are filling, and your ping is having to wait in line, wade through that buffer to get to the front before it can finally leave, and the same thing happens in reverse. So it's not unfilled buffers that are the problem. The buffers themselves are not the problem. It's that they're allowed to get too deep, and that creates latency. And the Internet was not designed - it was designed for on the order of 100ms of latency. I mean, I've seen some reports of six-second buffers.

Leo: That's not good.

Steve: Buffers that are six seconds deep. You might as well just...

Leo: Forget it.

Steve: ...hang up and go home, yes.

Leo: By the way, this Netalyzr gives you a lot of other interesting information.

Steve: Oh, it's a fantastic application. It's worth...

Leo: I'm looking at certain TCP protocols are blocked in outbound traffic. Not all DNS types are correctly processed. I mean, there's a lot of diagnostic information in here.

Steve: Yes.

Leo: This is my home router, which of course has something going on, probably Comcast. It's blocking remote SMB servers. Well, that's probably right.

Steve: Yeah, I'm sure it would be, yes.

Leo: But I have to say Russell, our IT guy here, is so good that our router is well configured here at the studio for minimal - he's done a great job for minimal latency, great speed. That's why we get such good results. But Steve, I want you to look at your router. You're getting too many packet reorderings, too much of that. If you want to know more about this show - he's silent. But we do, it's funny, given that you have such a rock-solid setup, it always puzzles me that we have occasional audio breakups with you, sometimes weird results with you.

Steve: Okay. What I will do is, for our next podcast next week, I will shut down one of my T1s. Because what did you see, by the way, for packet reordering?

Leo: Zero.

Steve: Really.

Leo: Yeah.

Steve: Interesting. And you're right. I don't know how Skype handles reordering. If it sees them out of order at all, it might just drop anything that's not coming along. Anyway, I'll shut down a T1, and we'll try a podcast next week with only one T1. I can do it trivially, so it's not a problem at all.

Leo: Yeah. That might be - it's funny, but that's a perfect example. Less sometimes is more. More sometimes is less. Actually, I'm sorry, I had 10 reordered packets. Wait a minute. Which - okay. Now I have this...

Steve: Yeah, but I had 900.

Leo: That was at home, 10 was at home. Let me see on the...

Steve: I had 1551.

Leo: Yeah, we had no reordering here in the studio, and I had 10 reorders at home.

Steve: Wow.

Leo: Yeah. So I'm thinking that that's the dual T1s. Maybe just, yeah, let's disable it next time. Because you have plenty, 1.4Mb up is plenty for Skype.

Steve: Yup. I will shut one down, and we'll see how it does.

Leo: Steve is the master, and you can find out more by going to his website, GRC.com. That's where you'll get of course the fantastic SpinRite, world's finest hard drive maintenance and recovery utility. You'll also get all his free doohickeys, his security stuff, and copies of the podcast, 16Kb audio and transcriptions. Those are only available at GRC.com. Next week a feedback episode, so while you're there, if you've got a question, just leave a question at GRC.com/feedback. And otherwise, if you want the video or the high-quality audio or you just want to subscribe, you can do that at TWiT.tv.

We do this show live every Wednesday, 11:00 a.m. Pacific, 2:00 p.m. Eastern time, that's 1800 UTC. So stop by and watch live. Always fun. Then you can watch the chatroom as it does its tests and brings Netalyzr down. It's fun. It's fun. And it's been fun watching. There's such a wide variety. We have somebody in Britain who's got amazing bandwidth, 25 pounds for BT Infinity. And he's just got 45Mb up and down or something like that, it's incredible.

Steve: Wow.

Leo: It's incredible. But there is a wide variety of ping times. We've got people with 1900ms ping times.

Steve: Ooh, there's - okay, that's two seconds.

Leo: That's too much buffering.

Steve: Yup. Wow.

Leo: What a great subject. And more to come, I'm sure, on this one. Steve, thank you so much.

Steve: Thanks, Leo.

Leo: We'll see you next week on Security Now!.