If I have seen a little further, it is because I am on the shoulder of giants.

- Isaac Newton

This is a personal history - but I've provided only one piece of the puzzle, and put some pieces together, generating some insight on the picture it creates. It's not a pretty picture.

Many more pieces, and much more important pieces, come to me from many other people, including Dave Clark, Rich Woundy, Dave Reed, Van Jacobson, Nick Weaver, Vern Paxson, and many others whom I do not know who they are... My apologies if I have overlooked your contributions.
A few pieces assembled... What is the picture?
Agenda

1. History
2. TCP
3. History - the new millenium
4. Host and home router bufferbloat
5. Dark Buffers - where are they and how do we detect them?
6. Spikey Latency
7. Aggregate network behavior
8. Mitigation and solution
9. Smoke is in the Air
10. Next Steps
History

Those who cannot remember the past are condemned to repeat it. - *George Santayana*
Some History - with a couple of personal perspectives

- October, 1986 - Collapse of NSFNet; early Internet became unusable: could it happen again?
- 1987-1988 - Recovery with Van Jacobson's Slow Start, etc. Collateral damage included foundation of the X Consortium, to my long term personal regret
- 1990's - High year on year Internet growth rates triggered by the Web, with resulting congestion
- 1993 - Floyd & Jacobson - “Random Early Detection (RED) gateways for Congestion Avoidance”
- 1995/1996 - Formulation of “two connections” rule in HTTP - RFC2068 - avoid self-congestion on memory starved dial up servers
- 1997 - “RFC 2309 Recommendations on Queue Management and Congestion Avoidance in the Internet” - RED Manifesto
- 1997 - Emergence of 802.11 after several predecessors earlier in the decade
- 1999 - Problem solved(?)....
2
TCP
The touch stone for congestion avoiding protocols
TCP Congestion Control and Avoidance

Congestion Window

Slow-start

Congestion Avoidance

Fast Recovery

TCP’s design assumption is that a congested network will generate *timely* notification of congestion, by packet loss or ECN (later). To do otherwise destroys *TCP and other congestion avoiding protocol’s control loops*.

Some packet loss is not only normal, it is essential to their correct operation!

We judge other Internet protocols by whether they are “at least as good” as TCP at avoiding congestion.

What happens if TCP’s timeliness assumption is violated by alot?

What happens if packet loss is avoided by buffering?
Behavior of Internet routers in the face of congestion was “sub-optimal”, resulting in an explosion of research on congestion.

AQM schemes were developed - RED and related algorithms; fair queuing, etc.

A few were even deployed widely in routers: e.g. RED, GRED in Linux kernel, etc.

So most of us thought the problem was “solved”.

- Some knew better
- > 100 papers just on tuning RED in the last decade, Blue, etc.
What happens when a network is slow due to bufferbloat?

Once a network/link exhibits high latency and bad packet loss, other critical, statistically insignificant but mission critical packets can't do their jobs

- **DNS** - adding 100's ms of latency to lookups kills web browser feel and losses cause lookup failures
- **ARP** - relies on timely resolution to find other devices on your network
- **DHCP** - if these packets are lost or excessively delayed, machines can't get on the network
- **VOIP** - needs about a single packet per 10ms flow in order to be good, and less than 30ms jitter.
- **Gamers** = will get fragged a lot more often with latencies above their twitch factor
- **Responsiveness of all network applications, web or otherwise, suffers**
What happens when a network becomes congested? - Video

Streaming video has become an important application. It can amplify bufferbloat problems and itself suffer from bufferbloat

- Streaming provides a base bandwidth load, so other activity sharing a broadband pipe will start saturated, or much closer to saturating the link, with its bad effects
- The bursty nature of the TCP transfers of both the streaming itself, and other activity sharing the link will, force much larger jitter buffers in the player
- The player may have great trouble determining the correct resolution operating point without oscillating on annoying and large time scales
- When the link is saturated due to streaming and/or other traffic, scroll bars to determine where the video should be playing will work very poorly
History
in the new millennium
This millennium

2000 - Internet bubble “pops” - glut on fiber market, memory continues to get cheaper
   VJ unable to get RED fixes deployed or published

2000's - present 802.11 goes ballistic

2001 - Bram Cohen invents Bittorrent: does not immediately “go big”

2002 - faster broadband deploys - along with much faster processors and improved TCP's with
   window scaling, DOCSIS 2 approved; 802.11 deploys, variable bandwidth commonplace

2004 - Bittorrent searches enable finding bittorrent content and mainstream use; first
   instance of Bufferbloat detected by Dave Clark on his personal DSLAM; Dave DOSes his son's
   WoW addiction by exploiting bufferbloat

2007 - Comcast “throttles” bittorrent; JG endures increasing complaints from family and
   starts having equipment damaged by lightning surges

2009 - Dave Reed's 3G network observations - reports to end-to-end interest - shouted down
   and ignored

2009 - JG yet again chases home network problems

April 2010 - JG's “Blue Box” tests

June 2010 - ICSI data published at NANOG
“The Internet is slow today, Daddy!”

I'd heard this for years. I even tried to debug my network many times. Every time I would go look, the network would stop mis-behaving.

- I had reason to mistrust my cable due to “adventures” in its re-installation
- Due to lightning and other surges, I had to replace and swap equipment repeatedly, and it kept being the “latest and greatest” home equipment
- I now work from home on immersive teleconferencing: good home internet service went from “nice to have”, to “essential to do my job”
- I came close to nailing this once before, in the fall of 2009. I had generated numerous service calls; but swapping in a different router made the problem apparently vanish. But, I'll never know for sure: it too, was blown up.
- April, 2010 - “Bluebox” bandwidth/latency clearly demonstrated a problem: poor latency during continuous data transfers: 1-2 seconds latency, with very rapidly varying 1-2 seconds jitter
- Lightning struck yet again in early June, 2010...
I'm a bear of little brain...

I was proving the Bell Labs “BlueBox” tunnel was obsolete, and had to go:

- I was bandwidth testing the BlueBox, just by copying files over the link.
- I also ran a “ping” to monitor the latency.

Two results:

- Expected: Blue box is too slow, and has to be replaced
- Also, the latency observed is also bad, and highly variable, ergo I have a problem there too, 1-2 second latency and horrible jitter

Third, completely unexpected result:

- Same behavior on my Internet service on a very similar test *without the Bluebox! Tilt*…

*I know I have to investigate, when I get a chance…*
The Smoking Gun - bufferbloat is killing latency in the Internet

“We rushed into the captain's cabin . . . there he lay with his brains smeared over the chart of the Atlantic . . . while the chaplain stood with a smoking pistol in his hand at his elbow.”

Lightning strikes thrice in the same place: All new equipment, yet again.

New DOCSIS 3 cable modem, new/old router, new switches, new single point ground system and eventually new ground system. I start hunting again in early July.

I find April's “blue box” problem hard to see! What's changed?

I can still see “smoke” but no fire...

I suspect Comcast's “TurboBoost” feature, combined with the DOCSIS 3 modem may be why it's now so hard to catch.

Who did TurboBoost, anyway? Google is the answer, what as the question: author happens to live nearby

Morning of our lunch 7/15: start seeing strong signature again of the problem; but no time to be conclusive before lunch.
Lunch with Comcast... 7/15/2010 - Many puzzle pieces

Lunch provided many, many, puzzle pieces that fell into place later.

- Suggests the “big buffers” problem, which he been chasing on suggestion of Dave Clark for over two years, but had had no proof of the problem
  - I could drop back to DOCSIS 2, and the differences between DOCSIS 2 and DOCSIS 3 could be used to rule out TurboBoost
  - But I still don't trust the cable up my hill, anyway. Too long, to many problems.

- RED is often not enabled in major networks. It is distrusted due to the need to tune RED, which has made many network managers averse to it. Some network managers run RED or other AQM, some do not.

- ECN is blocked in some networks - Dave Oran later explained to me that ECN packets crashes many (old) home routers

- Pointer to ICSI netalyzr.
(Almost) trivial test

Here's my simple test; copy a 90Mbyte file from Carlisle, MA to MIT:

- `scp RealTimeX.wmv expo.x.org: & ping expo.x.org`

10ms path (unloaded)

The file copy could just as well be:

- Your kid's bittorrent; your wife's YouTube upload, or even a fast download of a movie to your hard disk.
RTT of this path is less than 10ms!

Scp of X Consortium archives from my house to expo.x.org.

The periods of “good behavior” are when I suspended the copy to get work done.
Bursts of really, really ill behavior of TCP: what the f*** is going on!
“Typical” tcptrace plot of a perfect TCP session

Yellow: Receive Window
Red: Instantaneous outstanding data
Blue: average outstanding data
Green: weighted average outstanding data
HUH? .... Half a megabyte in flight over a 10ms path??? Spikes? WTF?

Yellow: Receive Window
Red: Instantaneous outstanding data
Blue: average outstanding data
Green: weighted average outstanding data
HUH? ..... WTF?

Throughput graph
HUH? .....  WTF?

RTT graph
HUH? ...... WTF?

Red: Instantaneous outstanding data
Blue: average outstanding data
Green: weighted average outstanding data
Plots looks like no TCP behavior I've ever seen....

Bursts of duplicate acks; bursts of retransmits; lots of SACK's

What's the cause?
- TurboBoost?
- Buffer Bloat?
- Brokenness on JG's cable?

Dropped back to DOCSIS 2 modem
- Same bad behavior - Comcast says it can't be TurboBoost
- Cable move at my house due to lightning means that technician tests JG's cable at home end, and Comcast has technicians check my cable at CMTS end. Nominal cable (once my interior TV wiring was removed, anyway); as good as it ever gets outside of a lab

Others reproduce the behavior: Partha, at Georgia Tech, Nick Weaver at ICSI
The Plot Thickens....... My Inlaw's FIOS service.
The Plot Thickens....... My Inlaw's FIOS service.

My inlaws wired FIOS service in Summit, New Jersey, 25/25 service, 7/30/2010
And thickens.... Wireless side of the FIOS home router

Wireless side of inlaw's FIOS service, Verizon provided router, > 400ms, 9% loss
I called the fire chiefs/consulting detectives for help

Some of you are real TCP experts; I'm not...

- Dave Clark, Dave Reed, Vern Paxson and Dick Sites have all looked this over, and agree with the conclusions
- Van Jacobson says there are timestamps in my data which proves the case for bufferbloat, since both ends were recent enough Linux systems
- What is more, the bloated buffers are defeating congestion avoidance

I wish I were wrong about Bufferbloat...
What's happening here?

Buffers are filling with time.

While this happens, the latency, and rises, until the buffers are full.

The packets can't drain except through the link; that can easily take seconds for some of these network/devices.

*Traffic classification (QOS) can not help you. These are stupid devices.*

Eventually the buffers fill. Bad things happen.™ Rinse, wash, repeat

Exactly what might be determinable from the traces.

How bad?

Nick Weaver sees high packet loss rates on his home service, with my simple test; I've typically seen 1-3% loss; but note the early smokeeping with very high loss rates. *Needs further quantification…*

How much resources are consumed by the resulting mayhem? Is the Internet stable? *Dunno…*
Objections

“ICMP Ping may be processed on slow path”

- This may be true, but does not jibe with the reality of my personal testing
- Using a TCP based “ping”, I get the same result as with ICMP
  - I’m using a http based “httping” by Folkert Van Heusden, recently updated at my encouragement to support persistent connections; also available in the Android marketplace; so far, it matches ICMP ping behavior in my testing

Current httping can be used to test latency via TCP/HTTP to any website, with correct options: one packet there, one packet back, actual HTTP on top TCP

You go look at the packet traces…

http://people.freedesktop.org/~jg/Experiments/
Triggers - saturation of the path

Uploads and downloads - uploads may often be worse than downloads due to details of hardware and asymmetric provisioning, but some examples include...

- YouTube uploads, Crash dump uploads
- Email with large attachments
- Bittorrent
- File copies/backups
- Downloads of movies to disk
- Video teleconferencing
- Web browsing sites like YouTube, Google Images, etc.
  - Cross product of clients ignoring HTTP 2 connections spec, and congestion window opening.
  - Quick tests make me very worried, along with browser authors ignoring HTTP/1.1 spec, for reasons that become clear later in this presentation.
Key points

You must test latency and bandwidth together (to congestion) to see if “dark buffers” exist in the path. And you'll only see the buffers adjacent to the minimum “goodput” link in the path you test.

What you are doing to other people's latency and performance, or what they are doing to your latency and performance is insidious. Each time I chased the bufferbloat problem it would escape. Only when I happened to test both simultaneously (to a properly provisioned endpoint) could I see the problem.

The bufferbloat problem can easily be so severe today to make routine web browsing painful for everyone sharing a single broadband connection and trigger multiple service calls (as I believe I placed in 2009). And, if you were provisioned five years ago, with minimal uplink speeds of 384 or 768Kbps, you would have suffered 4 to 8 time worse. Us geeks have suffered much less; we tended to buy more than minimal broadband service....

Trying to avoid packet loss by increasing buffers has induced greater packet loss!
Netalyzr was also under-reporting and under-detecting buffers on high bandwidth; note that this data mixes wireless and wired traffic, so may be contaminated with home router bufferbloat.
Current observations and issues with ICSI data

This isn't a cable versus DSL versus FIOS problem; the mistake has been repeated in all of broadband technologies.

ICSI results has wireless and wired tests mixed

I discovered that the Netalyzr test was both under reporting and under detecting buffer sizes particularly on high bandwidth links.

The situation is worse than this dataset shows. ICSI has revised their test, and their next roll up of data will give a better view of incidence

But the broadband edge case for extensive bufferbloat, in both uplinks and downlinks, has already been proven and the case closed by ICSI's data
Paranoia sets in

ICSI has already proven the broadband edge is broken!

But...
I think bufferbloat is (almost) everywhere...
Host and home router bufferbloat
During tests, I caught my home router doing terrible things!

RTT of this path is less than 10ms!

scp of files from my house to expo.x.org, while running speedtest.net

Recent commercial home router, 50/10 Comcast service, 9/10/2010
The Plot Thickens....... Home router experiments
Home router bufferbloat led me to host bufferbloat

Eight second latencies? Insanity… What is going on?

- I knew I had to investigate further
- I replicated this problem on several different modern home routers

Given the experience with broadband, and the ICSI data, I had become seriously paranoid

Eventually, I installed OpenWRT on a router, so I could try to understand better

- I had realized that Linux’s transmit queue might be a problem
- I twisted the txqueuelen knob, but nothing happened, WTF?
- More hair lost, and then I realized: the queue is on my laptop!

Any time broadband’s bandwidth exceeds wireless goodput at home, the bottleneck becomes the device/home router hop - this is an increasing problem, that I suffer from frequently due to two chimneys in my house + high broadband speed
Host bufferbloat, and often your router

Since home routers are usually using general purpose operating systems under the covers (e.g. Linux), the problem is on both sides of your wireless.

Buffers in modern OS’s (Linux, MacIntosh, Windows) include:

- Transmit queue(s): where the OS may be classifying traffic; and often rely on drivers to hint how large they should be: 1000 is now Linux’s default on a lot of drivers!
- Device drivers: sometimes buffer packets internally: e.g. OLPC wireless driver
- Device ring buffers: modern hardware often supports up to 4096 packets, possibly on multiple queues - the default size on current hardware is often ~256 packets
- The device itself may have internal packet buffers: e.g. Marvell device on OLPC has 4 packets
- Bus transfers???

Let’s do a simple calculation, presuming 10Mbps:

- 256 packets is of order 3,000,000 bits. So here’s 1/3 of a second (one way)

What happens at a busy conference, where your “fair share” might be 100Kbps? 30 seconds: applications (and users) timeout entirely....

And your excessive packet loss rate induced by bufferbloat is increasing failures further.
But you say, why don't I see bufferbloat on ethernet?

You can see bufferbloat trivially on Mac OSX and Linux on 100Mbps ethernet:

- Use a test program capable of saturating the connection....

- The network is slower than your 1Gbps NIC, so buffers build on your machine

- The driver ring buffers are/appear to be about 256 packets in size on modern hardware: you get ~10ms of latency with simple file copies

Windows is “interesting”... you get good latency, relative to Mac OSX and Linux

- The fastest any version of Windows will run (by default) is about 85Mbps; so the wire is faster than the OS, and therefore the OS buffers never fill

- A Microsoft tech note explains that to get better multimedia experience, Windows rate limits outgoing traffic

  • I think Microsoft noticed, but did not fully understand bufferbloat, and implemented a pragmatic band-aid
Where does host bufferbloat hurt?
Aggregate network behavior
RED, and all that
Aggregate network behavior - Back to the Future

What happens if a network has buffers “all over”? Such as any wired network without AQM enabled in its routers… Classic congestion of the 1990’s.

Buffers start working in an aggregate fashion

Latency will go up when loaded; but you won't observe much packet drops - Expect a diurnal (daily) pattern: people timeout before packets do

- This is exactly what Dave Reed reported in 2009 in 3G networks
- I’ve observed up to 6 second latency: Dave Reed has observed up to 30 seconds!
- This problem is known to exist in some (all?) RNC’s

- Wireless: first seen in satellite links in the ’80s.
  - Base station protocol adaptation, to cover error correction in high error environments
  - To cover radio bandwidth variation
  - Wireless protocols themselves (802.11, 3g)
  - Back-haul network failing to run RED or other AQM
Ten years ago, Van's wife Kathy walked into Van's office one afternoon, and showed him that RED has two bugs

- Van was unable to get CISCO to implement the fixes to RED
- Van was unable to get the paper with RED's problems and fixes published (twice). So Cisco customers did not ask for it

Van says RED is about 90% right; but the bugs mean that RED requires tuning, and the 100 or more papers about RED tuning in the last decade bear this out. Ergo, network operator's reluctance to enable RED is understandable, even if their fears are (usually) unwarranted

A RED in a Different Light draft did escape; a finished paper should be available soon. Warning: old draft nRED algorithm still has a bug

Van warns that time based behavior in Internet gear could also cause congestion and instability. Note that a number of recent broadband technologies clearly have this property. Van wishes one could pace packet transmission but hardware support is lacking - Need good evidence and writeup! Bunching of initially well paced packets has also been observed already. Synchronization and global resonance does occur!
Aggregate bufferbloat locations

- Head end
- Internet
- Hotel
- MOE Peru
- 3g carrier network
- RNC

CPE

© Alcatel-Lucent 2010
Fat Subnets: e.g. 802.11 or many other wireless technologies

Concrete example, 25 802.11 nodes, each with a single packet buffer, trying to transmit to an access point.

- Some nodes are far away, and the AP adapts down to, say 2Mbps?
- You have 25 * 1500 bytes of buffering; this is > .15 seconds excluding any overhead, if \textit{everything} goes well.

What happens if:

- You buffer 20 packets instead of 1 on each node? 200? or 1000?
- You keep trying to retransmit packets in the name of “reliability”? (some MAC's are known to try to transmit up to 255 times; 8 times in common)
- And, in the name of “reliability”, any inherently unreliable multicast/broadcast traffic drops the radio bandwidth to minimum?
- You then try to run WDS or 802.11s, which both forward packets and/or respond to any multicast (e.g. ARP) with routing messages?
What do you get? An ugly sight

Conferences using 802.11
Schools
Hotels
Some network operator’s networks

Painful personal example/failure: OLPC’s network melted under load

- Do you know what happens in Mongolia, where the teachers have the children all open their laptops at the same minute in the morning?
- Net result: no packets would get through; the MAC couldn't even decode the preamble of the packet
- Well before absolute meltdown, our applications failed due to timeouts
- We had several problems; but we missed bufferbloat entirely in our failure analysis
Dark Buffers

Where are they? How do we detect them?
Buffers are **only** detectable when they are **next** to a saturated bottleneck.

Your hosts, in your applications, and in socket buffers and network layers:
- Your MAC itself may have packet buffers internally;
- Network device drivers themselves;
- Your network interface's ring buffer potentially buffers large numbers of packets, often put there to hide x86 SMM (system management mode) behavior and for marketing;
- And the VM system your OS may be running on top of may add yet more layers.

Your wireless access, at both ends:
- 3g has buffers for fragmentation reassembly (how big?); I don't know about LTE;
- 802.11 has similar issues: long packet delays destroy timely notification.

Your switch fabric (8 ms/switch at 1GBPS): how many hops, how congested?

Your home router - potentially megabytes.

Your CPE/cable modem/FIOS box - potentially megabytes.

The head-ends of those connections (e.g. DSLAM, CMTS, etc.)

Each and every router in your path, and the line cards in those routers.

Speed of light in glass and vacuum.
Other Locations of Bufferbloat

Corporate networks

- Smokeping made me suspect our corporate network wasn't right: latency spikes were present.
- ALU corporate network example: sophisticated classification is present, but no AQM enabled; when we convert to Windows 7 and other systems, this becomes an immediate problem, as all hosts will be able to saturate any bottleneck.

Satellite Links

- I observed > 20 second RTT's on links to the MOE in Peru, for example.

Example Middlebox

- Tunnel devices
  - Detected in our IPSEC infrastructure, apparently at the firewall complex where the tunnels land.
- Firewall relays?
  - Haven't looked; but I expect so; if not in the OS, then in the relay applications.

Elsewhere?
Where do bloated buffers hide?
7

Spikey Latency

The Web browser/server problem
Does routine web surfing create “spiky” latency? YES!

- HTTP/1.1 spec prohibited more than 2 connections to the same path: chrome opens up to 6 connections to the same path. Initial inflight packets are therefore 12 data packets in current browsers such as chrome: already a problem... Firefox is 15 connections, therefore 30....

Need traces to augment the simple smoke tests I did, which showed up to 100ms(!) latency common on a 50/10Mbps broadband link.

Fundamentally, by using parallel connections, browsers are “diluting” the congestion information of the network. Not good....

- Due to XP’s lack of window scaling, it does take multiple connections to fully use a modern broadband link. Vista, Win7, Linux, OSX should not require lots of connections to saturate a link. Another problem caused by Vista’s failure...

To date, I have never seen data refuting our Sigcomm paper, “Network Performance Effects of HTTP/1.1, CSS1, and PNG” which showed 2 connections were sufficient, and in fact, faster, but given XP’s and HTTP’s limitations....
Web servers - TCP initial congestion window changes

Proposal is currently before the IETF to increase TCP's initial congestion window

Microsoft has apparently turned off the initial congestion window entirely.

Google has upped its initial window to 8 packets

The potential “impulse” into the network when visiting a web page with many images is therefore 120 packets (or more!)

This is up to 1,500,000 bits that may be in fight, to arrive at your broadband (or other wireless bottleneck)

And there is no classification, nor any fair queuing deployed at the bottlenecks where these impulses arrive!

Is this a good idea? I think - NOT!

I'd like to be able to use my Internet connection for applications other than surfing the web, without complaints from my family, strangely...

Right solution: Replace current HTTP with something better, please!
Stop papering over the sins of a bad protocol!
Mitigation and Solution

How do we get out of this mess? Exit signs for the theater…
Mitigation/solution depends on where the buffers are

High end home routers sometimes have QOS knobs that can immediately partially mitigate the damage (see following two slides)

- Can't deal with dynamic bandwidth availability (e.g. TurboBoost, TOD)
- Can't deal with possible upstream buffers in devices like DSLAM or CMTS's

In networks not running with AQM, “just” turn it on

- Education may be hard in some organizations - RED has had its problems

Improved AQM algorithms need research and deployment

- I'm trying to extricate Van's RED paper; it should be available soon

Hosts need both immediate mitigation and better queue management solutions

Home routers are often Linux based; only Gargoyle seems to use RED on its broadband link: but RED won't work to solve 802.11 or 3g bufferbloat due to wireless's highly variable bandwidth: nRED or other algorithms are needed
RTT of this path is less than 10ms!

Scp of X Consortium archives from my house to expo.x.org.

The periods of “good behavior” are when I suspended the copy to get work done.
QOS knobs in router set empirically - stable latency at all loads
From > 1 second, to less than 10 milliseconds jitter and latency
- 100 times better!
(DSL reports observation site is 20ms away...)
Mitigation/solution depends on where the buffers are

Market solutions

- Tests to make the problem visible
- “gamer” routers/modems/CPE that implement queue management

Existing plant may or may not be fixable, or only partially....

- Set buffers to something reasonable and small, if firmware update is possible: e.g. bandwidth delay product, where delay is based on geography
- Smarter algorithms between head end and CPE equipment - may require modifying standards in this area (e.g. DOCSIS, others)
- LEDBAT algorithms might be useful for routers to detect their broadband connection's queue's are growing to implement smart bandwidth shaping...

Hybrid: SOC's with existing silicon may be able to solve the problem: AQM everywhere!

New silicon
And the problem is getting worse...

Memory cost is now zero for most products: you can't buy small enough chips for the “right” size buffers. And “right” depends on delay, and bandwidth and number of flows; and bandwidth may depend on radio propagation. There is no single “right answer”.

Sizing buffers on measured RTT, rather than first principles: self fulfilling prophecy

Since there are only bandwidth tests available, if a vendor wants “on” to an ISP's “approved hardware list”, and if they have an underlying problem they can paper over by increasing buffer sizes, what do you think they do?

- This market behavior is absolutely known to be occurring

If you don't measure it, it doesn't exist....

We must must change bandwidth marketing wars to bandwidth/latency wars.

What is the best figure of merit? Better be something simple. We have the opportunity to suggest an answer.
Moral: drop early, drop often, and signal congestion

Tail drop is worst

- You mislead TCP and other congestion sensitive protocols the most, and delay their response, maximizing congestion as the end points can't adapt in a timely fashion.

Head drop is better

- But the buffers are still present, and may take a long time to drain if filled.
- ECN marks should be on all packets leaving a queue with big buildup.
- Issue: some networks, primarily consumer, suppress ECN bits. Don't know what fraction yet; Steve Bauer is working on this.

Buffers are often really badly sized: orders of magnitude larger than they should be.

- Buffers should be of order the delay x bandwidth x $\sqrt{\#\text{flows}}$ of the link.

Dynamic queue management (AQM) is no longer expensive to implement.

Queue management, everywhere....
Working engineers and managers must understand the consequences of excessive buffering and the necessity for active queue management.

Public: Bandwidth != speed; Bandwidth == capacity

- How to change the public discourse?
- Tests (e.g. Ookla, FCC, others).

Get rid of layered thinking: everyone has to understand congestion and their roll in it: we need “Layered thinking is evil” paper, with examples from congestion, wireless, and others.

Need visualizations of the phenomena, to educate engineers, the press and the public.

Make sure the next generation of gear works correctly.
Problematic mitigations/solutions

Changes to protocols - TCP and others

- Hard to get deployed, in some areas (servers, corporate desktops)
- Game theory says that algorithms that put individuals at a disadvantage are unlikely to deploy successfully - remember, it's the other guy who can do you in

Classification is not a solution; it just changes the pain points and is much less necessary if queues are managed in the first place, as the latency is much less

ECN

- End systems test pretty well
- Known to be blocked in a few networks,
- Steve Bauer should have much better information very soon

ECN may be immediately usable in some situations (e.g. smartphones) since they don't exchange much traffic with home networks where the routers were/are

I don't know about the home environment: insufficient data on broken kit
Smoke is in the air
Will our house burn down?
Network Neutrality Implications

I believe (part of) the Bittorrent problems were misdiagnosed, and as a result, consumer ISP’s may believe incorrectly there is an existential threat to lack of control of applications on the network.

- The pain level of consumers was extremely high when Bittorrent deployed; XP didn’t trigger bufferbloat suffering much in that era, due to it’s lack of window scaling.

Carrier’s telephony currently has a major advantage over other VOIP and Skype.

- I do not believe this was planned: but the conspiracy theorists will have a field day anyway. The consumer ISP’s take it on the chin every day by increased service calls caused by bufferbloat - I know, I generated quite a few!

Innovation is at risk.

- Since there is no classification in our broadband networks, there is currently no way to reliably deploy low latency applications (teleconferencing, games, hosted desktops) without, at a minimum, mitigating the home broadband line first.

*Fixing bufferbloat is essential to innovation in the Internet!*
Next Steps

Simple tests to expose the trouble

- Dave Clark is asking SamKnows to improve the FCC broadband test
- Speedtest.net  Others?  Whom?
- Build and make available “break your network” simple test
  - Ideally, it should identify which hop is the bottleneck, so that the right people get the service call
- Improve the netalyzr and m-lab datasets
  - Netalyzr was under reporting and under detecting the problem for the broadband edge, it's the best we've got right now
  - Need tools for testing corporate private networks
  - Can't get mac addresses from Java; need other tools than netalyzr to identify what hardware is bad
  - May be able to separate wireless from wired (would get us better insight into home router behavior) in Netalyzr: Nick Weaver is investigating.
What's needed - in short, HELP!

Van Jacobson's insights into RED's problems mean we must believe the anecdotal and some concrete evidence that many ISP's run without any AQM is correct.

Good examples of ISP's who are not running any AQM and the results of turning it on, before and after!

Tools to detect lack of AQM in ISP networks.

Was my experience to Ministry of Education in Peru bufferbloat, over a satellite link (20 second RTT's)? I think so.

Hall of Fame for now; a Hall of Shame would be much too long and we'd all be bozos to do so.

Visualization of the problem, so that engineers, their managers, the press, and people can understand the problem.

....
What's needed - in short, HELP!

Are there any home routers that work correctly already? Closest I've found is Gargoyle (OpenWRT derivative), but even it doesn't try to handle 802.11 bufferbloat

Fix the existing products we can fix; avoid making the mistake in future products

Simulation of nRED algorithm to tune its constant; test nRED and other possible algorithms in home routers

Is the Internet stable in the face of defeating congestion avoidance? Are we soldiers in step approaching a bridge?

Demonstration of solution of bufferbloat problem in home routers by using nRED or other algorithms such as that in Doug Leith's work - writeup and publication to show competitive advantage: can do immediately with open source router code, given the right hacker....

Education of the trade media who review SOHO gear

Better queue management in operating systems

Better buffer management in operating systems

Lots of help gathering evidence, and writing it up, and QUICKLY!
Worrying trends... As if I wasn't worried enough already...

Deployment of browsers with many TCP connections in parallel, diluting the congestion information

Windows XP is finally being replaced with operating systems with competent TCP implementations, but those systems can much more easily fill all the buffers that have been built into the equipment during this decade; bufferbloat has largely been hidden due to XP's limitations while the problem has ballooned

Many edge technologies are bunching packets together; bufferbloat also induces periodic impulses: will they self synchronize and wreck havoc?

Bufferbloat is likely getting worse with time; no common test exposes the problem, and the marketing dialog is all competition on bandwidth, not latency

Rising broadband bandwidth is often pushing the bottleneck into home routers and hosts, where the problem may be more severe

*We must take concerted action*
People's first reactions are/will be...

Apathy...
Disbelief...
It can't be me...
It's the other guy...
I don't understand...
It can't be as bad as you say...

How could it have been hidden so long? So it can't possibly be true...

And possibly most insidious, some organizations have taken strong, very public positions on issues such as network neutrality and won't want to revisit their belief systems, and also have business axes to grind...
I believe there were/are hidden, but very real, often extremely painful operational problems, that ISP's did not/do not want to admit publicly; and their positions have been formed on a faulty analysis of the root cause.
Remember, we are all bozos on this glass bus
no stones allowed!

My Blog - http://gettys.wordpress.com

Bufferbloat.net - http://bufferbloat.net
Questions ?

www.alcatel-lucent.com