Solving this traffic management problem... and the next, and the next

Bob Briscoe (BT & UCL), Lou Burness, Toby Moncaster & Phil Eardley (BT)

1. A Challenge

Some ISPs say they throttle p2p file-sharing sessions to protect lighter usage like Web. Actually we could make lighter apps go much faster without prolonging p2p transfers. Basic scheduling theory says if shorter jobs go faster they finish earlier, leaving the same capacity on average for longer jobs. As Figure 1 shows, rather than throttling p2p bit-rate, the key is for p2p file-sharing to have a lower weighted *share*. Then it would be much less aggressive to real-time streaming (e.g. VoIP) as well.

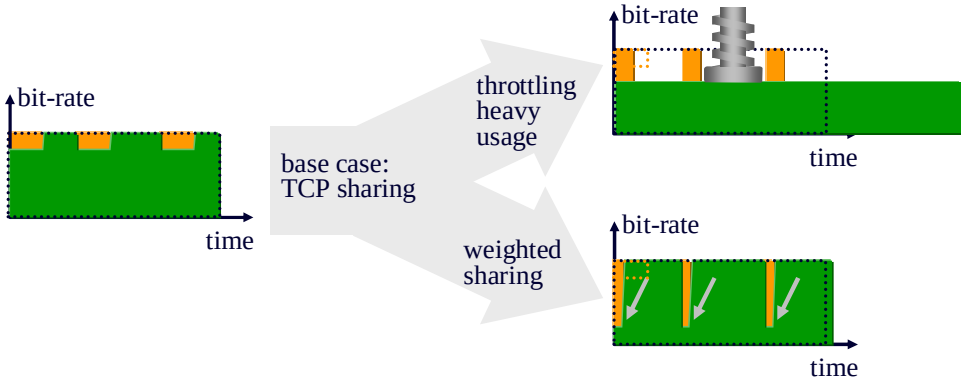

Figure 1—There's no need to prolong p2p downloads (darker shading) to make lighter apps faster

But can the IETF put all the pieces in place to make this happen? And make it as simple to deploy as a deep packet inspection (DPI) box?

2. Solutions... and Knock-on Problems

Control of weighted sharing can sit either more in the network or more on the host, typified by the two broadly equivalent protocols below ([Tussle] says we should not prejudge the winner):

Diffserv: Divide best efforts into two classes, with a `background' class at a lower scheduling weight [3GPPQoS];

Weighted congestion control: Elastic apps can tell the transport to behave like *w* TCP flows, with *w* less than one (for background) or greater than one (for interactive) [WeightCC].

2.1. Questions the Diffserv approach raises

APIs: Will consensus ever emerge on APIs for Diffserv? We need one for an app to detect which classes are available (at least on the first hop) and another for choosing the class.

Control: Will the network or the host decide which packets get priority? Initially operators are likely to take the easy path and use DPI to decide which apps get priority. Even with a Diffserv API, how will the user know the network is doing as asked?

Policing: What might be feasible ways to limit the amount of traffic per user in the higher class? Traditional Diffserv policing needs aggregated traffic from large sites, not individual users. Can there just be a priority volume limit over a month, without regard to whether the volume is sent on more congested paths or during peak periods? Can ISPs state in the contract when peak starts and ends? If they do, won't the problem always shift to just outside peak? Won't there sometimes be problems well outside peak period too?

Interconnect: On most networks (residential, enterprise, campus) more than half the traffic has one end on another network. Diffserv policing is sender-based, but ISPs often judge heavy usage by received traffic. If an ISP classifies me as light, and I'm sending to a peer classified

by their ISP as a heavy, will I use up my priority class allowance only for my traffic to go slowly due to the other network's scheduling?

Moving the problem: Once a large proportion of traffic is in the background class, won't we get the same problem again between users within that class? How do we encourage less urgent file-sharing to time-shift in favour of more urgent file-sharing? Do ISPs want their contracts to become ever more complicated?

2.2. Questions the weighted congestion control approach raises

APIs: The API problem is simpler than Diffserv, but discussion has only recently started.

Control: Control here is unambiguously with the host. Network control is all in the policing...

Policing: What stops high weight light usage causing bursts that harm r-t apps? What stops apps setting the weights of all their flows to maximum?

Interconnect: What makes one ISP police sources causing heavy congestion in other ISPs?

3. Features of a Good Solution

User control within an envelope: To protect the experience of other customers, ISPs should only need to confine each user within an overall envelope. If an ISP wants to prove it's neutral, it should be able to allow full user control of priorities within this envelope. This doesn't stop ISPs offering to prioritise apps to keep within the envelope, but that can be an optional service—not an imposition.

No need for wriggle-room: Acceptable use policies currently have to be woolly, with the ISP as the final judge of what constitutes reasonable usage. This is because we can only define overall envelopes in terms of volume, but harm to others depends on when the volume is sent, and where. We need a way to define an overall envelope that needs no room for later interpretation. Because wriggle room is needed, ISPs trying to be genuinely neutral are confusable with others who aren't, breeding suspicion and conflict.

Congestion volume metric: Unlike volume, an envelope using this metric would need no later interpretation. Congestion volume is greatest when the peaks are greatest, so ISPs needn't try to define when the peak period is. It would make both the above approaches to weighted sharing work, whereas a volume envelope doesn't (volume counts the same whether it's in a peak or a trough). A Diffserv policer defined using congestion volume works correctly even with no aggregation down to a single user.

Congestion volume is easy for your stack to measure – because it's the same as the amount of data discarded from your traffic. But users would need educating about it, just as they were about bytes when p2p first came in.

The intuition is as follows: many operators only count volume during the peak period, which is like weighting the volume you send by 100% during peak hours or zero outside peak. Congestion volume is like that, but the weight can be anything from 0 to 100%, not just one or the other. It's the volume you send weighted by the loss fraction when sending. So if you transfer 1MB of data along a path with a constant 3% loss fraction, your congestion volume will be 30KB.

A congestion volume envelope would encourage considerate behaviour right down to queuing timescales. It discourages burstiness, which helps real-time apps. It answers the earlier question of how heavily weighted short flows (e.g. Web) can be, before they cause excessive harm to others. It also discourages unresponsiveness, because an unresponsive app picks up more congestion volume than a responsive congestion control like TCP, even if they run at the same average rate. That's correct, because it's a true reflection of the cost to others of not responding quickly to each little congestion episode.

Counting across flows and over time: We need a metric (like volume) that adds up over multiple flows and that accumulates the longer you send. Congestion volume accumulates like volume, but without all the deficiencies of volume.

Imagine you have to keep within a congestion volume limit. Then, if you send parallel flows through 20 equally congested bottlenecks, at each you will want to take 1/20 of the share you would take from one alone. This deliberately results in *un*equal rates at each bottleneck.

Sending is the sender's responsibility (and forwarding the forwarder's): Receivers are often responsible for asking the sender to send to them. But ultimately, at the network layer, the sender can always choose how much to send and whether to send.

A metric for judging ISPs, not just usage: Congestion is the result of too much traffic meeting too little capacity. Congestion volume doesn't only measure how much congestion a user's traffic causes. It also measures how much congestion an ISP introduces into traffic, either directly within its own network, or indirectly by the routes it chooses to onward networks. When choosing which ISPs to attach to, you would compare congestion volume scores, which would encourage them to invest to alleviate congestion.

4. Do we know how to end an Arms Race? An architectural problem

By now it should be clear that the Internet would be all sweetness and light if only ISPs could confine users within an overall congestion volume envelope (semi-serious :) But *ISPs can't see the metric*, so they can't make users keep to it. Although it's really easy for endpoints to measure their loss volume, it's really hard for one domain to see losses in other networks. That's because the Internet was designed for endpoints to handle traffic control, not networks.

We've thought about this problem long and hard and succinctly documented our insights for the IETF [Problem]. Others have independently come to similar conclusions on what the problem is [Rest-of-Path] and on a possible solution [Accountability]. In our detailed protocol proposal [re-ECN] we've made it in a sender's interest to reveal congestion to the network in sent packets, so the network can limit congestion volume.

But our purpose isn't to push our own protocol (at least not here). It's to start consensus building on what the desirable features of a solution should be (§3). If anyone can develop a better protocol with those features, so much the better.

Our purpose is also to argue that defusing an arms race is a tricky business. Without deep understanding, attempts at solutions could at best lead to further problems (§2), and at worst add more fuel to the fire. We acknowledge there's some immediate standards work to be getting on with. This might even patch over 60% of the present problem. But in parallel, we ask the IETF to launch an activity to document and agree an answer to the big question that swarmcasting has made us face:

> The great thing about the Internet is that any of the thousand million or so hosts are free to use any network equipment anywhere in the whole Internet without asking. If we're going to introduce control over what share everyone gets, how do we best preserve as much of this freedom as possible?

This question is about the essence of the Internet. If the IETF doesn't want to have to swallow someone else's answer (e.g. DPI), we need to launch an architectural team not just quick fixes. From this high ground, we can also better judge which immediate standards work will still be sensible in the long term, rather than opening the gates to a flood of new signalling band-aids.

Please try to understand our arguments, and please argue back.

Acknowledgement

The authors are partly funded by Trilogy, a research project supported by the European Community under its Seventh Framework Programme. The information in this document reflects only the views of the author(s).

References

[3GPPQoS] 3GPP "Quality of Service (QoS) concept and architecture" TS 23.107, R7 (Sep 2007)

[Accountability] K Argyraki, P Maniatis, O Irzak, S Ashish & S Shenker, "Loss and Delay Accountability for the Internet," In Proc. IEEE ICNP'07 (Oct 2007)

[Tussle] D Clark, J Wroclawski, K Sollins & R Braden, "Tussle in Cyberspace: Defining Tomorrow's Internet," IEEE/ACM Transactions on Networking 13(3)462--475 (2005)

[Problem] B Briscoe, T Moncaster & L Burness, "Problem Statement: We Don't Have To Do Fairness Ourselves" IETF I-D draft-briscoe-tsvwg-relax-fairness-00.txt (work in progress, Nov 2007)

[re-ECN] Bob Briscoe, A Jacquet, T Moncaster & A Smith, "Re-ECN: Adding Accountability for Causing Congestion to TCP/IP" IETF I-D draft-briscoe-tsvwg-re-ecn-tcp-05.txt (work in progress, Jan 2008)

[Rest-of-Path] P Laskowski & J Chuang, "Network Monitors and Contracting Systems: Competition and Innovation," In Proc. SIGCOMM'06, ACM CCR 36(4)183--194 (2006)

[WeightCC] V Siris, C Courcoubetis & G Margetis, "Service Differentiation and Performance of Weighted Window-Based Congestion Control and Packet Marking Algorithms in ECN Networks," Computer Comms 26(4)314--326 (2002)