Workshop on Reducing Internet Latency, 2013

> *"My theory here is when an interface is faster, you feel good, and ultimately what that comes down to is you feel in control. The [application] isn't controlling me, I'm controlling it. Ultimately that feeling of control translates to happiness in everyone. In order to increase the happiness in the world, we all have to keep working on this."*
>
> Matt Mullenweg, WordPress

Summary

Internet latency has become a focus of attention at the leading edge of the industry as the desire to make Internet applications more responsive outgrows the ability of increased bandwidth to address this problem. There are fundamental limits to the extent to which latency can be reduced, but there is considerable capacity for improvement throughout the system, making Internet latency a multifaceted challenge. Perhaps the greatest challenge of all is to re-educate the mainstream of the industry to understand that bandwidth is not the panacea, and other optimizations, such as reducing packet loss, are at odds with latency reduction.

Internet latency has become a focus of attention at the leading edge of the industry as the desire to make Internet applications more responsive outgrows the ability of increased bandwidth to address this problem.

For Internet applications, reducing the latency impact of sharing the communications medium with other users and applications is key. Current Internet network devices were often designed with a belief that additional buffering would reduce packet loss. In practice, this additional buffering leads to intermittently excessive latency and even greater packet loss under saturating load. For this reason, getting smarter queue management techniques more widely deployed is a high priority. We can reduce these intermittent increases in delay, sometimes by up to two orders of magnitude, by shifting the focus from packet loss avoidance to delay avoidance using technology that we already have developed, tested, implemented and deployed today.

There is also plenty of scope for removing other major sources of delay. For instance, connecting to a website could be completed in one roundtrip (the time it takes for packets to travel from source to destination and back again) rather than three or four, by folding two or three rounds of flow and security set-up into the first data exchange, without compromising security or efficiency.

Motivating industry to deploy these advances needs to be aided by the availability of mass-market latency testing tools that could give consumers the information they need to gravitate towards low latency services, providers and products. There is no single network latency metric but several alternatives have been identified that compactly express aggregate delay (e.g., as relationships or a constellation), and tools that make use of these will give greater insight into the impact of changes and the diversity of Internet connections around the world.

In many developing countries (and in rural regions of developed countries), aside from Internet access itself, there are significant structural issues, such as trombone routes through the developed world and a lack of content distribution networks (CDNs), that need to be addressed with more urgency than Active Queue Management (AQM) deployment, but the 'blank slate' of new deployments provides an opportunity to consider latency now. More widespread use of Internet exchange points for hosting local content and fostering local interconnections is key to addressing some of these structural challenges.

The workshop concluded by identifying an action plan to carry the work forward.

Introduction

Reducing Internet latency is an engineering challenge that is gaining attention as we approach the limits to what can be achieved by simply increasing raw bandwidth in many regions of the network. The bufferbloat.net project has been instrumental in raising the profile of this topic in recent years. As we become increasingly dependent on a growing diversity of network applications that mediate social, economic and political interactions, it becomes imperative to remove unnecessary delays at every level of the stack. To explore the issue, the Internet Society, in collaboration with the RITE project, and with support from Simula Research Laboratory and the TimeIn project, convened a two-day workshop on the topic.[1]

Our scope for the discussion was deliberately broad. We included surveys of latency across all layers of the stack in both end systems and intermediate components, analyses of sources of latency and their severity and variability, the cost of latency problems to society and the economy, principles for latency reduction, solutions to reduce latency including cross-layer approaches, deployment considerations for latency reducing technologies, benchmarking and measurement considerations and the role of public policy.

Major goals of the workshop were to identify a metric for network latency, to develop an action plan to educate the industry and motivate deployment of latency reducing solutions, to identify gaps in our knowledge, and to identify any areas of disagreement for further discussion.

We divided the discussion into several sessions and the remainder of this report will document some of the discussion and major findings reached in those sessions. The main outcomes are summarised above.

Taxonomy

We surveyed sources of latency and categorised the solutions by quantifying benefits, considering deployment aspects, and short- and long-term applicability. This analysis provided a common reference framework for the remaining discussions.

Latency is the fundamental metric of computing and communication. All performance is measured as the delay between a question and an answer. The proposed taxonomy for latency focuses on the reason or mechanism of delay. A latency budget is applicable to the application and is consumed by sources of latency. Mitigations reduce the impact of latency sources on the overall budget.

Latency is the fundamental metric of computing and communication. All performance is measured as the delay between a question and an answer.

Latency budgets can be hard or soft and may be derived from biological or computational expectations. There are also cases when pure deadlines provide only an initial requirement; in some applications, reducing the latency further below the target can allow for more detail and/or additional computation to provide a better response. The scale and number of latency sources increase the cost to the application, whereas mitigations reduce the cost. The application operates effectively only when the cost is kept within the budget.

Sources of latency were categorised as:

- **Generation**: the delay between a physical event and the availability of data

[1] Further details including accepted position papers and presentations made during the workshop are available: http://www.internetsociety.org/latency2013/

- **Transmission**: inherent delay in signal propagation
- **Processing**: computational translation of the signal, e.g., for compression, encryption, etc.
- **Multiplexing**: delays necessary to support sharing the communications medium
- **Grouping/batching**: Mitigates some processing latency, but introduces latency of its own. (N.B. A lot of work on batching done over many years is now being undone – this is perhaps a consequence of previous over-optimisation for throughput.)

Specific examples of mitigations were categorized as:

- **Relocation**: Move the endpoints closer together, thereby reducing the transmission latency
- **Speedup**: Increase the number of operations per unit time, thereby reducing the processing impact
- **Dedication**: Reserve resources exclusively, thereby reducing the impact of multiplexing latency on the overall cost
- **Partitioning**: Split groups into individual components, thereby reducing the impact of grouping and batching

Two general mitigations were also identified. Avoiding latency by omission or substitution (which *reduces* the impact of latency from all sources, e.g., AQM etc.), and hiding latency through proactive communication (which *hides* the impact of latency from all sources, e.g., cache preloading, etc.).

A separate analysis of latency sources identified the following classes of delay:

- **Structural delays**: name resolution, server placement, etc.
- **Interaction between endpoints: protocol initialization, security context initialization, etc.**
- **Delays along transmission paths**: propagation delay, queuing delay, etc.
- **Delays related to link capacities**: insufficient capacity, medium-acquisition, etc.
- **Intra-end host delays**: buffering, Operating System latency, etc.

Figure 1 displays a selection of solutions where the colour of each bubble identifies which source of delay each solution attacks. A bubble diagram is used to emphasise that the placement of the bubbles is only approximate. The vertical axis shows that there are significant reductions in session completion time still to be made, and the vertical range of each bubble represents how variable the original source of latency can be. The horizontal axis shows how feasible it is to deploy each solution. Research is in progress to make the techniques with most impact easier to deploy. Where research cannot shift the highest bubbles any further to the right, some industry co-ordination may be necessary to achieve deployment.

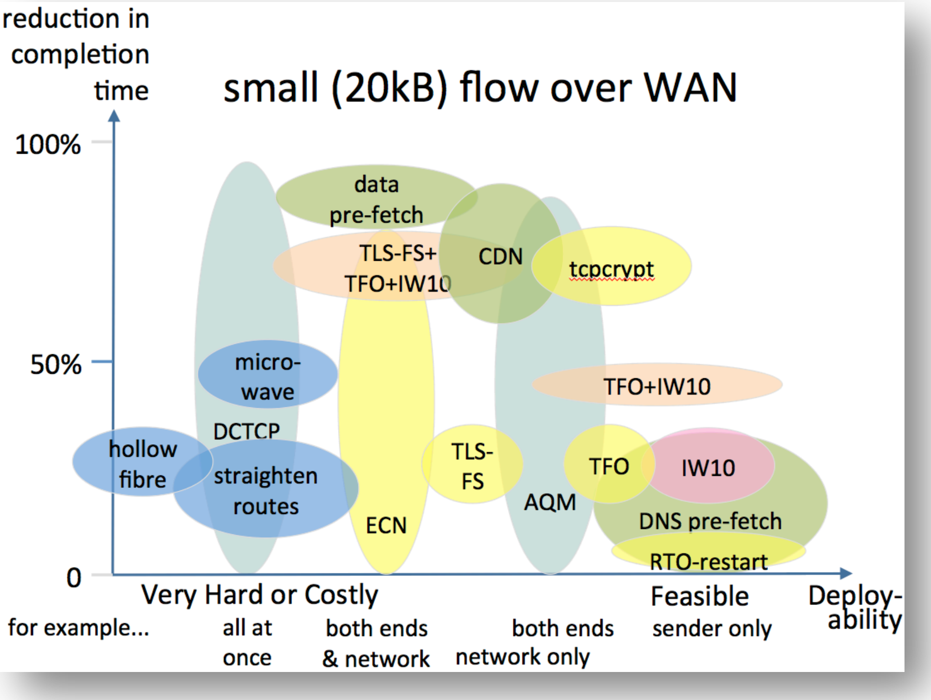


Figure 1: Latency reducing mechanisms

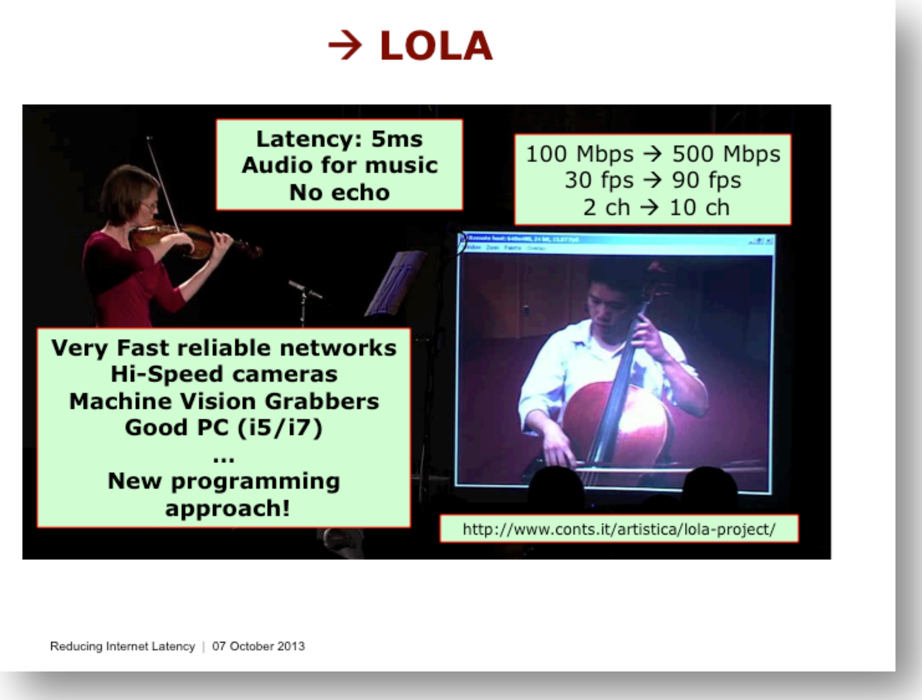
This analysis makes plain that if you care about latency, you have to be very careful and look in a lot of places for potential optimisations, and potential conflicts of those optimisations.

This analysis makes plain that if you care about latency, you have to be very careful and look in a lot of places for potential optimisations, and potential conflicts of those optimisations. The benefits of AQM, for example, are only one part of a much larger picture. For instance, reducing protocol initialisation delays (i.e. the number of roundtrips before payload data transfer can begin in earnest) can have a profound impact on the overall latency experienced for short flows. Work such as TCP Fast Open (TFO) and Transport Layer Security False Start (TLS-FS) is important to minimise protocol initialisation overhead as part of the overall latency cost for a transfer.

Use-cases and demanding applications

When two parties are trying to communicate and each has some state, latency introduces a bubble of uncertainty about the other party's state that grows as latency increases. The question for the application is how big a bubble can be tolerated. Unpredictability of delay is also key – this is related to Mullenweg's point about who's controlling whom – keeping jitter under control can be as important as reducing latency.

In this session we explored the potential for better Internet experiences and applications if our low latency goals could be realised. How much more responsive could the Internet be? What would that mean for applications and users?



→ LOLA

Latency: 5ms
Audio for music
No echo

100 Mbps → 500 Mbps
30 fps → 90 fps
2 ch → 10 ch

Very Fast reliable networks
Hi-Speed cameras
Machine Vision Grabbers
Good PC (i5/i7)
...
New programming approach!

http://www.conts.it/artistica/lola-project/

Reducing Internet Latency | 07 October 2013

Figure 2: LOw LAtency (LOLA) audio visual streaming system

The LOLA audio-visual streaming system (see Figure 2) is an example of a low latency application that, while not practically deployable on the Internet (yet), does help to illustrate some important considerations for low latency applications in general. LOLA has been used on many occasions to demonstrate live musical performance with remote participation, for example see this video: http://www.garr.tv/home/viewvideo/422/performing-arts/lola-internet2-fall-2012.

There may be finite limits to the geographical area within which an application is practicable, given biological factors and engineering/physical limitations.

There may be finite limits to the geographical area within which an application is practicable, given biological factors and engineering/physical limitations. 25ms of one-way delay is the comfort limit for most musicians collaborating remotely, which restricts collaboration using LOLA to continent-sized regions.

Designing and deploying ultra-low latency applications can also help to flush out previously overlooked issues. With LOLA, all the problems associated with competing traffic were sidestepped by deploying over a dedicated network, but this still flushed out issues in the network, such as:

- Suboptimal network paths;
- Asymmetric routes that mean latency is higher in one direction than in the other;
- Fail-over routes that take suboptimal paths leading to spikes in latency in the event of failures on the main path, and;
- Badly configured routing protocols leading to bad paths (from a latency perspective).

Stripping out ‘features’ in codecs can minimise processing latency and similarly, removing filters, access control lists and other ‘intelligent devices’ from the network can cut down on processing latencies. Removing IP altogether and dumping Ethernet frames onto an optical path without error correction is under consideration by the LOLA team, but clearly this is not a practical approach where Internet applications are concerned.

For Internet services, improving multiplexing mechanisms of all kinds (see Taxonomy, above) is at least as important as application optimisations and specialised hardware of the kind employed in LOLA demonstrations.

Measurements to metrics

The symptoms of excessive latency can be hard to isolate, and the causes obscure. What measurements do we have to shed light on the scale and character of the problems we need to address? Can we agree on the need for and definition of a metric for (access) network latency and is a single metric even possible? How can we reliably identify and address latency issues introduced by wireless networks?

We spent most of our time exploring the question, 'Can we agree on the need for and definition of one or more metrics for (access) network latency?' A summary of points from the discussion follows.

It would be useful to define a metric for the latency budget required to connect to the Internet via a given access provider for purposes of comparison (or even service differentiation) in the market.

It would be useful to define a metric for the latency budget required to connect to the Internet via a given access provider for purposes of comparison (or even service differentiation) in the market. However, the extent to which the latency budget attributable to each access network is related to total latency experienced by a given activity is unclear – network topology beyond the access network may be more important, for instance. There is therefore a relatively large design space for such metrics. The latency budget metric may be related to one or more network latency cost metrics, e.g., single-bit exchange delay or bulk transfer delay.

Before any of these metrics could be used as an incentive in the market for access providers to work to reduce latency, there will also need to be education of the user base – they'll need to care about latency before they'll make purchasing decisions based on latency metrics.

Potential latency metrics include:

- Flow size / average rate
- RRUL test (Realtime Response Under Load: application delay measurements with TCP streams in the background to induce load)
- Ratio of unloaded to loaded message latency (as measured by RRUL)
- Average TCP flow initialisation round trip time (RTT) to Alexa top websites (as used in present latency anomaly detection schemes)
- Average load times for Alexa top websites (capturing the impact of bandwidth as well as single-bit delay in a single metric)
- Ratio of network latency to direct geographical speed of light delay between the endpoints (network stretch – see the Structural Issues section below)
- Roundtrip message delay to a specified reference point along the path (e.g. within the access provider's domain (autonomous system or AS), to the border of the first AS after the provider's AS, or to diverse reference points).

On this question, it is clear that more work is needed. It was agreed that there is no single metric, and that we need any such indication to somehow express the way in which different messages with different properties are impacted by the sum of delays. For example, latency cannot be defined by a single measure, but is more of a curve, with an approximate intercept (the delay of a single bit) and an approximate slope (relating the size of the message to the additional time for message transfer). This curve need not be continuous, monotonic, or have any other definite property, however.

As a first step, it seems that initial implementations of clearly-defined, basic measurements that capture network latency from a given access network customer's observation point (including derivation of these from existing data sources) would be a useful place to focus effort, in order to have an experimental environment to explore the design space. Metrics are about incentivising more research, changing behaviour, re-engineering, etc.: without any metric, we're in the dark.

The space is complex enough that metrics for one purpose (describing the latency dimension(s) of the performance of ‘the connection’ to ‘the Internet’ in a way understandable to non-experts) may have quite different properties than metrics for another (e.g. localization of excess routing-policy-induced delays). There's a basic tension here between ease of understanding and accuracy requirements, but it still seems desirable to attempt to define something relatively simple for the former case. Focusing on a metric for use in commerce would allow us to fix some parameters, perhaps arbitrarily, in order to arrive at a reasonable metric that reflects average user experience.

Focusing on a metric for use in commerce would allow us to fix some parameters, perhaps arbitrarily, in order to arrive at a reasonable metric that reflects average user experience.

Finally, promoting the importance of measurements by content providers for the health of the Internet could help change the minds of those that are prejudiced against supporting measurements as they believe they won't make any difference. Carefully analysing incentives and value chains is important here.

A few additional points made during the discussion, to guide further work:

- Latency is always additive, and responsibility is cumulative: while an access network may not have direct control of causes of latency at its peers or upstream providers, it does have control over and responsibility for who it peers with, buys transit from, and its routing policies.
- Basic metrics should be defined in terms of ‘network physics’ – quantities with well-defined measurement methods easily understood by implementers.
- Metrics used in commerce should correlate strongly to quality of experience and where necessary, be derived from or composed of these basic metrics.
- A multidimensional variable would be harder to game, in contrast with ISPs optimising for connectivity to speedtest.net measurement servers.
- On the other hand, easily defined tests that are simple for end users to understand which have some relation to both latency and bandwidth – e.g. the start-to-finish load time of the front pages of a selection of the Alexa Top 500 websites – may be useful as well. Here the target is ease of measurement and alignment with end-user intuition as opposed to simplicity of definition.
- For quantifying queue-related latency, the ratio of unloaded latency to latency under load is a useful metric; this may be applied to devices for benchmarking as well as to paths in the network.
- Certain users are more latency-sensitive than others, and have an intuitive understanding of the effects of latency; gamers are the prototypical example here. It may be useful to reach out to game companies/networks (e.g. Steam) to do large-scale latency measurements.
- Tools for reducing page load times are fairly mature: Firefox, Chrome, and Safari all ship with detailed tools for visualizing load time and determining the causes of load delay. Network latency metric design (and the design of tools to measure them) can take guidance from these efforts.
- One data point for latency metrics in the wild: Operational latency measurements at one operator use TCP flow initialisation RTT on web requests from defined measurement points to defined websites (Alexa list). Monitoring focuses on detecting anomalies and changes to

help with pro-actively identifying and troubleshooting operational issues (e.g., bad cache selection).

- As capacity growth continues, the number of transfers limited by the TCP slow-start algorithm increases. One analysis shows that the distribution of flow sizes on the Internet over the last decade means that only a tiny percentage of flows will achieve average transfer rates close to modern access link capacities: the bulk of transfers will never get out of the slow-start phase of TCP congestion control.

An additional point about metrics from the policy discussion: while it can be difficult to reason about metrics for latency with unknown or complex technical causes, it may be much simpler to reason about these metrics for known causes (in many developing countries, latency is often caused by long-distance international peering as opposed to more expensive in-country connections). Here, coarser metrics are useful, and designing metrics for this situation may inform metric design for finer-grained situations as well.

Congestion control and AQM

Some of the questions that we began this session with were:

- Can we agree a set of congestion control requirements?
- What does it mean to do no harm?
- What kinds of harm (if any) are acceptable?
- Is parameter-less AQM a realistic goal?
- Is there a role for Explicit Congestion Notification (ECN)?
- Is tight coupling between Layer 2 and Layer 3 queuing and retransmission mechanisms necessary?
- Does it matter if we all deploy different smart queuing techniques?
- Can delay based congestion control be made to work in the presence of competing packet loss based flows (and is using delay gradient the answer here)?

The workshop attendees spent some time discussing the potential for making application limited streams (where the sending rate is application limited rather than congestion controlled) more aggressive. For these kind of streams, latency is the key performance metric and the delay of each individual message is important. More redundancy and more aggressive retransmissions can help minimise latency incurred through the packet loss recovery techniques of a reliable transport (TCP). Application limited streams were demonstrated to be at a disadvantage when sharing resources with greedy (bulk-transfer, throughput maximising) streams. Allowing more aggressiveness (for example by performing a fast retransmit on the first duplicate acknowledgement, and allowing up to 6 retransmissions without any retransmit time-out backoff) for application-limited streams creates a more level playing field. However, such aggressive behaviour should be carefully designed to ensure that it is not susceptible to abuse either through overuse or inappropriate use.

Regarding ECN deployment, the consensus of the group was that ECN has failed to deploy in part due to the original semantics that provide insufficient benefit. Some optimism was expressed that different semantics (in particular ‘data centre’ style or ‘low threshold’ ECN) might be more deployable because they provide an early and more frequent signal that could be used to implement more accurate control. This would run in conjunction with existing packet loss-based mechanisms (but with different parameters). The details were not discussed.

Delay based congestion control has its uses in closed environments (e.g. data centres) and for scavenger traffic (e.g. RFC6817, low extra delay background transport – Ledbat). It is especially useful in combination with other signals. The delay signal is naturally attractive as it is a measure of the quantity we're trying to reduce, but it's a very noisy signal and delay based congestion control algorithms don't play well with packet loss-based algorithms. Delay based congestion control mechanisms may not be *the* solution for the Internet, but could be part of the solution.

AQM deployment clearly has traction now – both fq_codel (Flow Queuing Controlled Delay) and PIE (Proportional Integral controller Enhanced) have been implemented in Linux and fq_codel is already seeing deployment. PIE is specified for implementation in the recently finalized DOCSIS3.1 specification. A warning was sounded that once both AQM and modern TCP advances are deployed, serious capacity allocation problems could be exposed. It is therefore important that we avoid deploying any AQM mechanisms that prevent us from doing something in the transport layer at a later date to address these capacity allocation issues. There are unresolved differences between workshop participants about embedding per-flow queuing in network devices such as home gateways. It may be that some people see a lack of empirical evidence of problems, while others are concerned about these predicted interactions, even though they are not visible today.

Cross-layer primitives would be very nice to have and there might be something there for a latency-related research group to work on. It was noted that the developer documentation for Apple's iOS7 operating system includes an application-programming interface (API) intended to realize callbacks across all layers of the stack so that application developers can respond and react appropriately to network-related events.

There was violent agreement amongst the workshop participants that there is no single drop algorithm that always helps and never hurts. It is unclear whether this reality is in conflict with the desire for a 'no knobs, just works' solution.

One very important point that needs promotion to a wider audience now is that we can reduce latency caused by excessive buffering under loaded conditions by up to two orders of magnitude with technology that we already have developed, tested, implemented and deployed today (see Figure 3). The Linux fq_codel queuing discipline is enabled by default in the OpenWRT firmware Linux distribution for embedded devices like home gateways. However, the existence of scenarios where new AQM algorithms do require configuration means we still have work to do.

One very important point that needs promotion to a wider audience now is that we can reduce latency caused by excessive buffering under loaded conditions by up to two orders of magnitude with technology that we already have developed, tested, implemented and deployed today

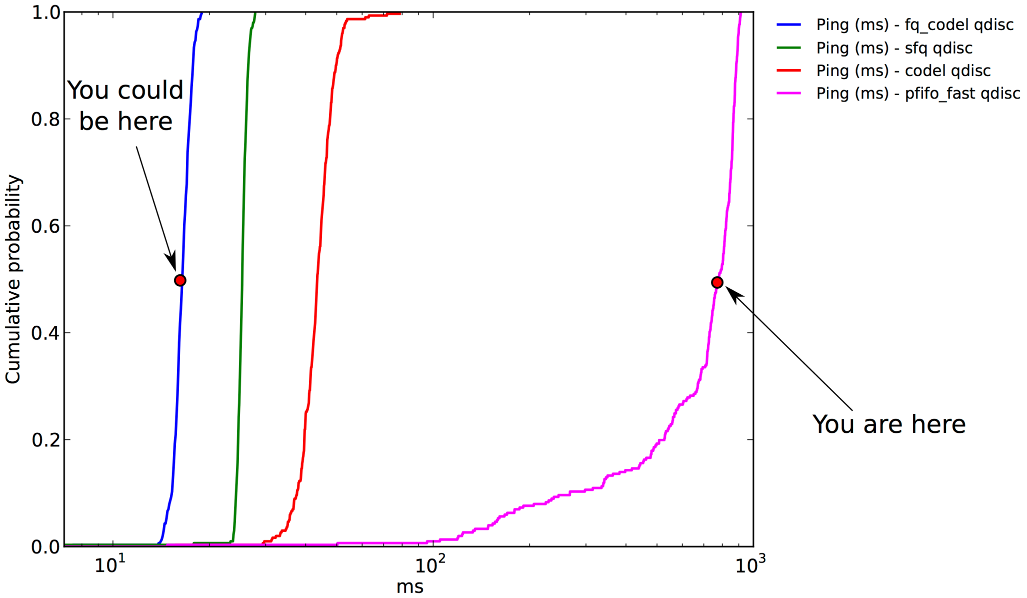


Figure 3 Realtime Response Under Load (RRUL) test results[2]

Figure 3 indicates that combining flow queuing with an effective drop algorithm can have synergistic benefits. However, it was not universally agreed that flow queuing is a pre-requisite for safe AQM deployment. It *was* agreed that any safe and effective AQM is much better than no AQM.

The DOCSIS 3.1 specification for cable modems mandates the PIE AQM algorithm without any flow queuing component. Other algorithms, possibly combining flow queuing, are optional. Enabling some form of AQM will be mandatory in DOCSIS networks going forward. It is clear that we have now passed the stage of waiting for lab tests to complete before making decisions about algorithms to implement in silicon/firmware, as for the cable industry at least, a choice has been made.

In some cases, retrofitting new AQM and packet scheduling technologies into existing equipment is entirely infeasible. PIE may meet the need for AQM algorithms that can be applied in existing routing and switching platforms that do not require operator tuning.

Structural issues, regulation & public policy considerations

In many developing countries (and in rural regions of developed countries) there are significant structural issues that pertain to Internet latency. This session identified and characterized these issues.

There are some fundamentals:

- Local content hosting – Content Delivery Networks are a priority;
- Interconnection via exchanges in the developed world needs to evolve into local peering – thereby keeping local traffic local.

In some cases these issues are best addressed through improved public policy and market regulation rather than purely technological approaches.

In some cases these issues are best addressed through improved public policy and market regulation rather than purely technological approaches. Developing Internet exchange points as local infrastructure for hosting and interconnection is key and the Internet Society already has a

[2] http://files.toke.dk/bufferbloat/bufferbloat-final.pdf

considerable amount of capacity building work underway here, for example: http://www.internetsociety.org/what-we-do/issues/internet-exchange-points-ixps.

Geo-location of users to content sources has to be applied carefully – for example, in Africa, geo-locating to a content server in a nearby country rather than using a server in Europe seems like a good idea, unless the nearby country is routed *via* Europe. While the ratio of unloaded latency to latency under load measures the presence of bufferbloat, the ratio of network latency to geographical distance between the endpoints (network stretch) measures the quality of the routing infrastructure - both are important components.

Structural issues aren't just a developing country concern. Monitoring of popular web destinations by a major North American ISP provides insight into various misconfigurations and transient problems that can result in needlessly high transmission latencies. By continuously monitoring the RTT and traceroutes to popular destinations, the ISP can rapidly take action either to fix its network, or contact a third-party content provider to address high latency issues. This is obviously of benefit to the ISP's subscribers, but also serves to raise awareness amongst the broader community of the kind of misconfigurations that can lead to problems. High RTT cases typically fell into the following categories:

- website provider's algorithm for assigning customers to servers not adequate
- website providers using criteria (e.g. load) other than RTT in assigning customers to servers
- components involved in serving web content have inadequate resources or non-optimised configurations
- smaller websites may not have multiple server locations
- occasional misconfigurations

Workshop participants discussed the regulatory landscape as it applies to the topic and identified some potential roles for regulators:

- Gathering and publishing of statistics - several regulators do already provide some information on latency measurements, e.g. UK Ofcom, US FCC, Singapore Infocomm Development Authority and the EU have all employed SamKnows to support measurement activities in their regions;
- Setting benchmarks - e.g. the regulations regarding potential imposition of minimum quality of service in the EU;
- Gaming of metrics – ensuring the game resistance of metrics may lead to regulatory requirements.

It was generally agreed that more tools, in the hands of more end users, generating more data would always be preferable to regulatory intervention.

It was generally agreed that more tools, in the hands of more end users, generating more data would always be preferable to regulatory intervention.

Action plans, deployment and co-ordination

> *"Everybody talks about the speed of light, but nobody ever does anything about it."*
>
> Joe Touch

Establishing demand for better technology requires users to become aware that better technology is available, and that it is within reach. As already mentioned, identifying and working with existing industry incentive structures and value chains will be key to getting deployment of new technologies that can reduce Internet latency.

End-host solutions (e.g. removing rounds of protocol handshaking) will be a big improvement in the short term although there is a key requirement that such solutions not interfere with the deployment of network-based solutions in the longer term (e.g. there has been concern that the increase of TCP's initial window to ten segments (IW10) may create pressure for larger buffers). Algorithm performance is secondary to the need to ensure that no barriers or disincentives to deploying network solutions are introduced. Delay-sensing algorithms must do no harm (e.g. Ledbat's 100ms delay target) especially when not needed

Network-based algorithms and systems should not increase latency unless necessary either. For example:

- Reducing packet loss by increasing buffers (which is why we now have AQM solutions);
- Hiding packet losses in broadband lines using interleaving can add about 20ms of delay, even though modern transports and applications are robust to such low packet loss levels;
- AQM algorithms delay any congestion signals for a worst-case roundtrip (e.g. 100ms), which is necessary if the signals are drops, but not if they are explicit congestion notifications (ECN).

The choice of AQM algorithms deployed on the Internet does not need to be uniform, therefore debating the comparative merits of different algorithms should be a niche activity. Any algorithm that manages sharing efficiently in the relevant deployment conditions is fine. Of course, understanding the deployment conditions is crucial, as is understanding the goal of 'efficiency' - again, targeting packet loss reduction is how we ended up with excessive queuing latency under load.

While AQM deployment can mitigate the impact of buffer bloat, it runs the risk of exposing TCP's underlying RTT unfairness. Flow queuing techniques in combination with AQM appear to be a powerful tool for delivering per-flow fairness and flow isolation. By flow queuing we mean queuing that continues to help performance and reduce the impact of sharing capacity, while reducing queuing delay. Further research and discussion is necessary to establish consensus on the desirability of flow isolation as a goal.

Identifying and adopting a mass-market test for network latency behavior can help stimulate the inclusion of network latency as a feature in descriptions of Internet service provision.

There is a strong need to educate and improve user expectations of performance (both end users and web developers) to demonstrate that a better experience is possible and is within reach thus helping people to know when to demand improvements. Identifying and adopting a mass-market test for network latency behavior can help stimulate the inclusion of network latency as a feature in descriptions of Internet service provision. Latency behavior of networking hardware products should also be visible in marketing material and benchmarking activities. Likewise, application vendors should include transaction latency considerations in their support and marketing messages.

The inclusion of simplistic performance indicators related to packet loss in Service Level Agreements (SLAs) is a problem because packet loss is not necessarily bad. People are deploying buffers to minimise packet loss because of the commercial implications of SLAs. Shifting these commercial arrangements to take account of the underlying engineering is a particularly tough challenge.

The workshop concluded by identifying a set of actions to carry the work forward, as follows:

1. **Educational material (video clips, video lectures, whitepapers)**
    a. To explain the importance of latency compared to bandwidth and packet loss

    b. Aimed at vendor and operator audiences
    c. The RITE project has a relevant deliverable due this autumn.

2. **Developing a latency under load metric**
    a. This could be pursued in the IP Performance Metrics working group at the IETF where a relevant milestone could be added to the charter if there was a draft describing what is needed, and providing a specific statement of applicability.
    b. A useful metric must quantify things that somehow strongly correlate with user-perceived quality.
3. **Latency/cross-layer interactions research group**
    a. This needs someone to draft a coherent charter.
    b. This could include work on updating host implementation recommendations.
4. **Routing and topology metric**
    a. More data is needed to determine whether this is a problem.
    b. The ratio of geographic distance to network distance (stretch) could be a useful metric.
5. **Tooling initiatives**
    a. This may involve working with speedtest.net.
    b. This may involve working with the Large-Scale Measurement of Broadband Performance (LMAP) Working Group in IETF.
6. **New definition of Explicit Congestion Notification (ECN) semantics**
    a. This is intended to improve the incentives for ECN deployment.
7. **Conflict between latency and other priorities**
    a. For example, it is proving hard to remove the latency impact of Transport Layer Security (TLS) that is used to encrypt web traffic, which must be weighed against growing calls for ubiquitous encryption.
    b. These trade-offs could be addressed in an architectural document.

References & Further reading

The Bufferbloat Project - http://www.bufferbloat.net
Joe Touch's Latency page - http://latency.org
Jim Gettys' blog - http://gettys.wordpress.com
The RITE Project – http://riteproject.eu
Bufferbloat: Dark buffers in the internet - http://queue.acm.org/detail.cfm?id=2063196
RFC970: On Packet Switches With Infinite Storage - http://www.rfc-editor.org/info/rfc970
Google Make the Web Faster project - http://developers.google.com/speed/
WebPageTest - http://webpagetest.org
SamKnows regulator collaboration - http://www.samknows.com/broadband/regulators
Active Queue Management Algorithms for DOCSIS 3.0
http://www.cablelabs.com/downloads/pubs/Active_Queue_Management_Algorithms_DOCSIS_3_0.pdf
DOCSIS3.1 MAC and Upper Layer Protocols Interface Specification -
http://www.cablelabs.com/specifications/CM-SP-MULPIv3.1-101-131029.pdf
DOCSIS3.1 Press Release -
http://www.businesswire.com/news/home/20131030005843/en/CableLabs®-Announces-Generation-DOCSIS-®-Technology
Ookla Speedtest - http://speedtest.net

Glossary

AQM	Active Queue Management
CDN	Content Delivery Network
CoDel	Controlled Delay
DCTCP	Data Centre TCP
DNS	Domain Name System
DOCSIS	Data Over Cable Service Interface Specification
ECN	Explicit Congestion Notification
Fq_codel	Flow Queuing Controlled Delay
IETF	Internet Engineering Task Force
ISP	Internet Service Provider
IW	Initial Window
IXP	Internet Exchange Point
PIE	Proportional Integral controller Enhanced
RTO	Retransmit Timeout
RTT	Round Trip Time
TCP	Transmission Control Protocol
TFO	TCP Fast Open
TLS	Transport Layer Security
TLS-FS	TLS Fast Start
WAN	Wide Area Network

Internet Society
Galerie Jean-Malbuisson, 15
CH-1204 Geneva
Switzerland
Tel: +41 22 807 1444
Fax: +41 22 807 1445
www.internetsociety.org

1775 Wiehle Ave.
Suite 201
Reston, VA 20190
USA
Tel: +1 703 439 2120
Fax: +1 703 326 9881
Email: info@isoc.org