A Survey of Latency Reducing Techniques and their Merits

Bob Briscoe* Anna Brunstrom David Ros David Hayes
Andreas Petlund Ing-Jyh Tsang Stein Gjessing Gorry Fairhurst

1 Introduction

Our working assumptions are that i) the performance of data communications is increasingly held back by limits other than bandwidth and ii) removing these limits will often involve simple inexpensive changes to protocols and code. If true, removing these limits would be a more effective use of time and resources than expensive link hardware upgrades. Indeed, upgrading link speeds would waste investment if it made little difference to performance because the limits were elsewhere.

This position paper gives a status report on work we have recently started to survey techniques for reducing the delays in communications. The immediate aim is to organise all the techniques into a meaningful categorisation scheme, then to quantify the benefit of each approach and produce visualisations that highlight those approaches that are likely to be most fruitful.

In these visualisations we also want to show how difficult it is likely to be to deploy each technique. This will result in a map of the gain and pain involved in each technique that the industry can work through. We don't solely want to identify the low-hanging fruit (high gain, low pain); we also want to identify high-hanging fruit (high gain, high pain) that will need to be picked off eventually, but may require some industry co-ordination or long-term planning to reach.

Many people's mental model of the applications that people most value consists of long-running data-transfers. This reinforces the obsession with bandwidth and causes latency gains to be undervalued. The performance of transactional traffic (e.g. Web, financial applications, gaming) is much more dependent on latency than bandwidth. And recent work [SBA13] shows that a large proportion of long-running TCP flows actually consist of numerous brief 'flowlets'. So, even more Internet traffic than we thought is either in short transactions or shorter flows within longer connections.

Such work on characterising Internet traffic complements our survey work. Having quantified the latency gain from different techniques, it becomes possible to also quantify how much this gain will benefit typical Internet users, providing hard evidence to alter the mindset of the mainstream data communications industry, which we believe is unhealthily obsessed with bandwidth.

2 Potted Survey

The full survey will be made available[1] before the workshop, in parallel to submitting it for publication.

We tried various alternative ways of organising all the techniques, and found that arranging by the source of delay that each approach addresses was the most useful—it accommodates them all with the least overlaps and gaps. Here we give a brief outline of the full survey's structure, populated with a few examples of delay-reducing techniques it will feature:

Reducing structural delays:

1. Server placement, e.g. cache placement, CDN
2. Software architecture techniques

Reducing delays due to interaction between end-points:

1. Name resolution, e.g. DNS cache pre-fetching
2. Authentication, e.g. SSL False Start
3. Protocol Initialisation, e.g. TCP Fast Open

*bob.briscoe@bt.com, BT Research & Technology, B54/77, Adastral Park, Martlesham Heath, Ipswich, IP5 3RE, UK

[1] via http://riteproject.eu/publications/

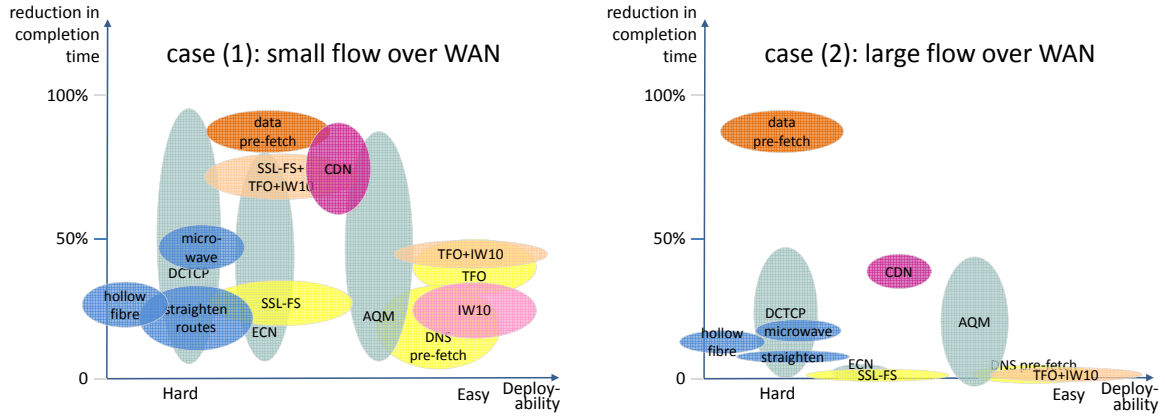

Figure 1: Bubble plots of rough latency gains against ease of deployment for a selection of techniques

Reducing delays along transmission paths:

1. Propagation delay, e.g. straighter cable routes, higher signal propagation velocity (microwave does both, hollow fibre does the latter)
2. Forwarding and switching/routing delay
3. Queueing delay, e.g. Active queue managment
4. Delays repairing errors and discards, e.g. ECN

Reducing delays related to link capacities:

1. Related to sharing capacity, e.g. multiplexing
2. Related to sensing capacity, e.g. TCP initial window = 10

Reducing delays in end-hosts:

1. Application delays, e.g. data pre-fetching
2. Operating System delays, e.g. parallelisation

3 Quantifying the Benefits

Quantifying the benefit of each technique requires consensus on a figure of merit. We decided on percent reduction in session completion time (100% – (delay/original delay)). Unfortunately very good techniques all bunch up just under 100%. Session speed-up (original delay/delay) would solve this, but then the majority of reasonable techniques bunch around 1–1.5, which would be worse.

Figure 1 arranges a small selection of the techniques in the full survey as a bubble plot with reduction in session completion time on the vertical and ease of deployment on the horizontal. Bubble diagrams are generally approximate, which suits the rough precision of the data being presented. In general, bubbles higher and to the right are better. However this interim view should not be used to prioritise work, because it only includes a small selection of the techniques in the full survey.

The vertical extent of each bubble represents the likely variance of the latency reduction, while the vertical positioning of each bubble's caption represents the typical reduction to be expected.

The benefit of each technique depends on the scenario: specifically a) the size of the data flow and b) how far apart the end-points are (or were originally), e.g. WAN, LAN. It should be sufficient to visualise just two cases for each of these two dimensions, leading to a 2×2 matrix of cases. Given space restrictions, Figure 1 illustrates only the two WAN cases; both for short flows (less than a dozen packets) and for long.

Space does not allow for further commentary here, but it will be given in the full survey and at the workshop.

Acknowledgement

Nick Gates (Uni Cambridge) produced the initial report on which this work was based. The authors are funded by the European Community under its Seventh Framework Programme through the Reducing Internet Transport Latency (RITE) project (ICT-317700). The views expressed are solely those of the authors.

References

[SBA13] Matt Sargent, Ethan Blanton, and Mark Allman. Modern Application Layer Transmission Patterns from a Transport Perspective. Under submission, May 2013.