Congestion Control in Real Time Media - Context

Input paper for the congestion control workshop to be held in Vancouver, July 28, 2012
Harald Alvestrand, Google

Abstract

This paper explains the background and reasons for the interactive media congestion control being such a hot topic at this exact moment in time, and delves somewhat into the need for metrics by which we can understand whether a proposed algorithm achieves its purpose or not.

Introduction

The surge of interest in congestion control for real time media has been accelerated by two major trends:

- The rapid deployment and increased usage of real-time communication across the Internet, exemplified by Skype, Google Talk and Google Hangouts
- The WebRTC effort, which has held out the hope that writing such applications could go from being something that only large corporations could support to something that anyone with access to a browser and a Web server could participate in.

The new features of these efforts, compared to the previous wave of in-house, intra-company, intra-service or otherwise controlled-environment real time communication equipment (IP phones and video rooms) are:

- They envision operating on the open Internet, without any ability to call upon network operators or IT support staff to ensure special configurations or bandwidth guarantees
- They may get deployed in large enough numbers that they constitute a significant part of Internet traffic in some times and places - meaning that what they do may affect others.

The last point means that congestion control algorithms for these types of applications is not only of interest to those who deploy them; those who use other services on the Internet have an interest in making sure the Internet is robust against congestion collapse even with very widespread deployment of these services.

Knowing when we succeed

When designing a new protocol function for use in the Internet, we always face this problem: How do we know that we're winning?
We've had numerous technologies defined and deployed; some of them have become staples of the Internet (SMTP, HTTP), others have failed to gain adoption (SNMPv3, PEM). On some, we don't know, even after many years (DNSSEC, IPv6).
One solution to this conundrum is to define a metric even at the outset, some baseline to compare against, and to figure out a way to measure it. If the metric is compelling enough, we can argue for deployment of the solution - saying "you can measure that you're winning".

In this case, there are two high-level metrics that we'd like to optimize for:

- Service Quality: Is the quality of the video conversations "good enough to be useful"?
- Internet Impact: Does the Internet suffer congestion collapse?

Neither of these are directly measurable. As stand-ins for those metrics, I'd offer the following as examples of things that can be measured, and are related enough to our high-level metrics that they matter:

- There exist a set of measurement conditions for which, If <proposed solution> is not in use, a video conversation has significantly degraded quality or is impossible, while if it is in use, the degradation is not enough to be called "significant". Note that the quality of an interactive conversation is affected both by picture quality and by delay.
- There exist a set of measurement conditions for which, if a video conversation is not active, average packet loss for other traffic is < 1%; if a video conversation without <proposed solution> is active, average packet loss is > 5%; if a video conversation with <proposed solution> is active, average packet loss for other traffic is still < 1%.

Detailed definition: I'm using the term "significant degradation" by analogy to a MOS score (ITU-T P.800) of 2 or less; the assumption here is that a system that adapts to a bandwidth restriction and has low packet loss will deliver better pictures than a system that does not adapt.
The 5% loss figure corresponds to a max TCP throughput of ~250 Kbits/sec on a 200-ms RTT link, according to the TCP throughput equation as quoted in RFC 3448; it is a reasonable stand-in for "congestion collapse" in an Internet where high-speed transfers are expected to be the norm.
The term "video conversation without <proposed solution>" is another undefined term; given the current state of technology, a solution with no congestion control at all that responds to network condition is a reasonable baseline.

These are measurement criteria that are clearly demonstrable and measurable, even though the conditions under which they are observed can be quite contrived; if it's impossible to demonstrate either of these properties by either simulation or demonstration, I'd argue that we have not given a compelling demonstration that the proposed solution solves our problem.

Thus, we have solved a different problem: We've found a criterion by which we can demonstrate that we've failed - because an algorithm that can't pass these tests isn't showing that it's useful.

Verifying that we have an effect

Another error we'd like to avoid is to spend lots of effort guarding against a specific issue - only to find that this issue does not occur in practice.
That is the job of instrumentation, which is different from testing: Making sure we collect from real deployments data that proves that we're having an effect.
For a real time video congestion control algorithm, such measures might include:

- The time that the bandwidth is reduced to less than the app's desired maximum
- The typical difference between max-desired and available bandwidth (bandwidth change over time)
- The packet delay and jitter observed before, during and after the congested times
- The packet loss observed before, during and after the congested times
- Objective quality measures on the video that is presented during the congested times

(frames dropped, frames with known decode damage....)

The important thing is to observe that the mechanisms we have deployed made a difference, and to verify that the times they made a difference, that difference was positive. The latter may actually be harder than the former.

Dealing with change

No problem stays the same for long on the Internet. Changing conditions - such as fiber to the home, LTE wireless, terabit fibers, stateful firewalls, AQM routers, BitTorrent deployments - will make sure the landscape we have to adapt to changes.
For each change, the challenge is to demonstrate that our algorithms are useful; most of the time, that means our measurement suites have to be improved; sometimes that means that the algorithms have to change.
Measurement is the key.