Issues with Using Packet Delays and Inter-arrival Times for Inference of Internet Congestion

Wesley M. Eddy
MTI Systems, Inc.

Background and Introduction

Congestion control for real-time voice and video media traffic has been a topic of interest in the IETF for many years. One of the motivations for DCCP and TFRC is to support such flows and support a more desirable congestion response than the existing standard TCP algorithms. Since such media flows have been using RTP, support for RTP over DCCP was put onto the IETF Standards Track in RFC 5762, however it is not widely used at this time.

Due to “bufferbloat” and the desire to minimize latency for realtime media traffic, loss-based congestion control is problematic. This is because loss only occurs when queues overflow and significant latency has been created for the involved flows. Delay-based congestion control may be preferable, since it can potentially maintain low latency for flows, when they are protected from competing traffic. With the recent IETF RTCWEB Working Group activity that may result in wider-spread use of RTP over the Internet, there is a desire to find and deploy appropriate congestion control for RTP. However, no DCCP-supported algorithms are using delay-based congestion control. There is at least one proposal for a delay-based congestion control algorithm [1] that would most likely be implemented in RTP itself rather than in DCCP. The approach is based on sensing congestion through use of increases in the one-way delays or packet inter-arrival times (relative to packet size). This has a number of currently unaddressed issues that make it a longer-term research project compared to simply using existing mechanisms (like DCCP/TFRC).

This brief paper describes a number of the issues with the currently-proposed approach that have not been addressed clearly to-date, and concludes with a number of derived requirements that may provide some direction to longer-term research in the applicability of delay-based congestion control for real-time flows.

Brief Overview of Currently-Proposed Approach

The one currently-proposed algorithm for delay-based congestion control of RTP flows [1] involves use of RTP timestamps generated per-frame at the sender and arrival timestamps recorded by the receiver. The packet inter-arrival time model in this algorithm includes an additive Gaussian white-noise component assumed to have a mean over time that indicates congestion. This is used to derive a means for estimating Internet channel capacity and congestion through Kalman filtering of relative inter-arrival time samples. A receiver-side component of the algorithm attempts to detect over-use of the channel and provide rate updates to the sender. A sender-side component of the algorithm uses the receiver-provided estimates as one input, in addition to loss rate and round-trip time, into its rate-selection decision.

Open Issues in the Current Approach

To understand the challenges in measuring, filtering, and utilizing packet delay samples, first requires a systematic description of the components of one-way delay and other factors that influence the measurement. There is not enough real-estate in this paper for such a description, but this section summarizes the main issues, several of which will need to be addressed in any viable proposal for use of one-way delay or packet inter-arrival time measurements as congestion control signals.

1. Comparison of timestamps generated on different hosts is subject to error due to offset, skew, and drift in the respective hosts' timing mechanisms (as described in RFC 2330). Offset does not necessarily pose a problem for this algorithm, however skew and drift can be major impediments. Skew is known to possibly account for accumulation of errors in one-way delay estimates on an order of 6 milliseconds per minute, and can be bidirectional. Drift, being the rate of change in skew, can further complicate any mechanism that might accommodate for skew. As it stands, the current proposal for delay-based RTP congestion control does not discuss clear means of dealing with these issues.
2. The resolution of timestamps generated on multiple hosts may differ, and depending on the implementation, may be generated at different levels in the software stack, at differing proximity to the network interface, and with different disturbances to their timing, due to factors such as operating system task/thread scheduling. To high-level software, where timestamps are generated via libraries or cascades of lower-level calls, the resolution may not be visible, or may even be inaccurately inferred due to attempts made at lower-layers to provide monotonically increasing timestamps even when the clock update rate is at a lower frequency than timestamps are requested from it. The variation in latency between timestamping and actual "wire-time" of the packets, combined with complications in timestamp resolution and update rate can cause disturbances to the delay measurement algorithm, if not accounted for. In some cases, these may be Gaussian, but it is not well-characterized for the scope of devices and software systems that RTCWEB targets.
3. If the rate of packet production is periodic (e.g. driven by frame rate of audio/video capture and playback), then this will result in periodic sampling of the network, which can be sensitive to phase issues if there are other periodic processes at work (either at the network, or at the edges). This may cause either over-pessimistic results, or addition of non-Gaussian noise to the samples.
4. One-way delay is a property of an individual packet's journey across the network. There are no guarantees that subsequent (or prior) packets follow the same path. Particularly if devices that are mobile between access networks (e.g. laptops, pads, mobile phones, etc) are envisioned to be supported, the state estimation algorithms need to be able to cope with impulses that occur during horizontal handovers, as well as step-changes in the network path, possibly on several orders of magnitude in the case of access technology transitions. Even for a fixed-device, however, packets can potentially be balanced (and even re-ordered) across multiple links within the network. Although this is known to be problematic to transport protocol performance, it does still occur and is a real phenomenon.
5. High-levels of variation in both capacity and delay can be imposed by the adaptive modulation/coding and packet/frame scheduling algorithms in various MAC protocols (including very commonly in recent flavors of WiFi) due to aggregation, support of broadcast/multicast, power-saving, and other factors not directly correlated to congestion. Since this is likely to be present on the access links used at both ends of RTCWEB flows, the level of disturbance it implies will need to be understood and
6. The network capacity estimates can only be confidently generated when network queues are large enough to both avoid significant loss rates and measure variations. However, short queues are desired in order to minimize latency of interactive applications. If the queues are very short, measurable delay variations will not occur, only packet losses, that are not utilized within the receiver-side algorithm.

7. The sender-side component of the algorithm uses a measured loss rate in order to update its sending rate, with an increase below 2% loss, no change between 2% and 10% loss, and a decrease proportional to the loss rate at more than 10% loss. These thresholds are arbitrarily selected, and represent very large loss rates in terms of normal Internet congestion. Further, since the motivation for using delay-based components is to avoid building queues, and typically only large queues generate losses, a 2% or higher loss rate should only be reached if the algorithm is not performing correctly.
8. The sender-side component uses the TFRC equation to generate a lower-bound, not an upper-bound as the current standards do. By definition, the described algorithm will then not be TCP-friendly, and this may be considered an issue for a class of traffic that the proponents expect to generate large numbers of flows on the public Internet.

Proposed Requirements for Algorithms Using One-Way Delay

Based on the issues identified, in this section we conclude by proposing a number of requirements for algorithms that attempt to use one-way delay or packet inter-arrival time measurements in RTP congestion control.

1. Include clear means for compensating offset, skew, and drift in timestamp sources, if the measurements depend on comparison of timestamps generated on multiple hosts.
2. Include comprehension of timestamping issues due to multi-tasking delays, clock resolution, and possible interpolation of timestamps, and (at least) ensure that such errors will not result in over-optimistic estimates for available network capacity.
3. Avoid assumptions of particular statistical distributions (e.g. Gaussian) on quantities, when these are not validated by extensive measurement campaigns, and determine among the many contributors to noise in the measurements which of them are significant to the algorithm, and which should be filtered or otherwise corrected for.
4. Incorporate detection and appropriate response for conditions such as reordering, delay steps or spikes, and cases where there is simply too much high-rate variation in network conditions for accurate estimations to be made based on the available measurements and algorithms.
5. During both formulation and later validity testing of the algorithm, include consideration of shared wireless links (e.g. multi-user WiFi), in order to ensure the algorithms are valid in cases that are now perfectly normal (and not pathologically odd), even though they may be much more complex than wired Ethernet access.
6. Avoid fragility in the case of short queues; assume that small buffers are a desirable property of the network, and that current large buffers may not always be the norm.
7. Avoid fixed thresholds unless they can be strongly motivated, and especially avoid relying on thresholds that would indicate extremely poor network latency or losses for other flows.
8. Address TCP-friendliness as a clear goal if the traffic will be competing for network resources with other typical Internet flows, until the network infrastructure supports a usable level of flow isolation or advanced congestion sharing methods that can be utilized.

References

[1] H. Lundin, S. Holmer, and H. Alvestrand, “A Google Congestion Control Algorithm for Real-Time Communication on the World Wide Web”, draft-alvestrand-rtcweb-congestion-02, work in progress, April 25, 2012.