Network Working Group
Internet Draft
Intended Status: Informational
Expires: September 14, 2014

X. Zhu
R. Pan
Cisco Systems
March 13, 2014

NADA: A Unified Congestion Control Scheme for Real-Time Media
draft-zhu-rmcat-nada-03

Abstract

This document describes a scheme named network-assisted dynamic adaptation (NADA), a novel congestion control approach for interactive real-time media applications, such as video conferencing. In the proposed scheme, the sender regulates its sending rate based on either implicit or explicit congestion signaling, in a unified approach. The scheme can benefit from explicit congestion notification (ECN) markings from network nodes. It also maintains consistent sender behavior in the absence of such markings, by reacting to queuing delays and packet losses instead.

We present here the overall system architecture, recommended behaviors at the sender and the receiver, as well as expected network node operations. Results from extensive simulation studies of the proposed scheme are available upon request.

Table of Contents

1. Introduction

Interactive real-time media applications introduce a unique set of challenges for congestion control. Unlike TCP, the mechanism used for real-time media needs to adapt fast to instantaneous bandwidth changes, accommodate fluctuations in the output of video encoder rate control, and cause low queuing delay over the network. An ideal scheme should also make effective use of all types of congestion signals, including packet losses, queuing delay, and explicit congestion notification (ECN) markings.

Based on the above considerations, we present a scheme named network-assisted dynamic adaptation (NADA). The proposed design benefits from explicit congestion control signals (e.g., ECN markings) from the network, and remains compatible in the presence of implicit signals (delay or loss) only. In addition, it supports weighted bandwidth sharing among competing video flows.

This documentation describes the overall system architecture, recommended designs at the sender and receiver, as well as expected network nodes operations. The signaling mechanism consists of standard RTP timestamp [RFC3550] and standard RTCP feedback reports.

2. Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

3. System Model

The system consists of the following elements:

* Incoming media stream, in the form of consecutive raw video frames and audio samples;

* Media encoder with rate control capabilities. It takes the incoming media stream and encodes it to an RTP stream at a target bit rate R_v. Note that the actual output rate from the encoder R_o may fluctuate randomly around the target R_v. Also, the encoder can only change its rate at rather coarse time intervals, e.g., once every 0.5 seconds.

* RTP sender, responsible for calculating the target bit rate R_n based on network congestion signals (delay or ECN marking reports from the receiver), and for regulating the actual sending rate R_s accordingly. A rate shaping buffer is employed

to absorb the instantaneous difference between video encoder output rate R_v and sending rate R_s. The buffer size L_s, together with R_n, influences the calculation of actual sending rate R_s and video encoder target rate R_v. The RTP sender also generates RTP timestamp in outgoing packets.

* RTP receiver, responsible for measuring and estimating end-to-end delay d based on sender RTP timestamp. In the presence of packet losses and ECN markings, it also records the individual loss and marking events, and calculates the equivalent delay d_tilde that accounts for queuing delay, ECN marking, and packet losses. The receiver feeds such statistics back to the sender via periodic RTCP reports.

* Network node, with several modes of operation. The system can work with the default behavior of a simple drop tail queue. It can also benefit from advanced AQM features such as RED-based ECN marking, and PCN marking using a token bucket algorithm.

In the following, we will elaborate on the respective operations at the network node, the receiver, and the sender.

4. Network Node Operations

We consider three variations of queue management behavior at the network node, leading to either implicit or explicit congestion signals.

4.1 Default behavior of drop tail

In conventional network with drop tail or RED queues, congestion is inferred from the estimation of end-to-end delay. No special action is required at network node.

Packet drops at the queue are detected at the receiver, and contributes to the calculation of the equivalent delay d_tilde.

4.2 ECN marking

In this mode, the network node randomly marks the ECN field in the IP packet header following the Random Early Detection (RED) algorithm [RFC2309]. Calculation of the marking probability involves the following steps:

* upon packet arrival, update smoothed queue size q_avg as:

$$q\_avg = alpha*q + (1-alpha)*q\_avg.$$

The smoothing parameter alpha is a value between 0 and 1. A value of

alpha=1 corresponds to performing no smoothing at all.

* calculate marking probability p as:

p = 0, if q < q_lo;

$$p = p\_max * \frac{q\_avg - q\_lo}{q\_hi - q\_lo}, \text{ if } q\_lo <= q < q\_hi;$$

p = 1, if q >= q_hi.

Here, q_lo and q_hi corresponds to the low and high thresholds of queue occupancy. The maximum parking probability is p_max.

The ECN markings events will contribute to the calculation of an equivalent delay d_tilde at the receiver. No changes are required at the sender.

4.3 PCN marking

As a more advanced feature, we also envision network nodes which support PCN marking based on virtual queues. In such a case, the marking probability of the ECN bit in the IP packet header is calculated as follows:

* upon packet arrival, meter packet against token bucket (r,b);

* update token level b_tk;

* calculate the marking probability as:

p = 0, if b-b_tk < b_lo;

$$p = p\_max * \frac{b - b\_tk - b\_lo}{b\_hi - b\_lo}, \text{ if } b\_lo <= b - b\_tk < b\_hi;$$

p = 1, if b-b_tk>=b_hi.

Here, the token bucket lower and upper limits are denoted by b_lo and b_hi, respectively. The parameter b indicates the size of the token bucket. The parameter r is chosen as r=gamma*C, where gamma<1 is the target utilization ratio and C designates link capacity. The maximum marking probability is p_max.

The ECN markings events will contribute to the calculation of an equivalent delay d_tilde at the receiver. No changes are required at the

sender. The virtual queuing mechanism from the PCN marking algorithm will lead to additional benefits such as zero standing queues.

4.4 Comments and Discussions

In all three flavors described above, the network queue operates with the simple first-in-first-out (FIFO) principle. There is no need to maintain per-flow state. Such a simple design ensures that the system can scale easily with large number of video flows and high link capacity.

The sender behavior stays the same in the presence of all types of congestion signals: delay, loss, ECN marking due to either RED/ECN or PCN algorithms. This unified approach allows a graceful transition of the scheme as the level of congestion in the network shifts dynamically between different regimes.

5. Receiver Behavior

The role of the receiver is fairly straightforward. It is in charge of four steps: a) monitoring end-to-end delay/loss/marking statistics on a per-packet basis; b) aggregating all forms of congestion signals in terms of the equivalent delay; c) calculating time-smoothed value of the congestion signal; and d) sending periodic reports back to the sender.

5.1 Monitoring per-packet statistics

The receiver observes and estimates one-way delay d_n for the n-th packet, ECN marking event 1_M, and packet loss event 1_L. Here, 1_M and 1_L are binary indicators: the value of 1 corresponding to a marked or lost packet and value of 0 indicates no marking or loss.

The equivalent delay d_tilde is calculated as follows:

$$d\_tilde = d\_n + 1\_M\ d\_M + 1\_M\ d\_L,$$

where d_M is a prescribed fictitious delay value corresponding to the ECN marking event (e.g., d_M = 200 ms), and d_L is a prescribed fictitious delay value corresponding to the packet loss event (e.g., d_L = 1 second). By introducing a large fictitious delay penalty for ECN marking and packet losses, the proposed scheme leads to low end-to-end actual delays in the presence of such events.

While the value of d_M and d_L are fixed and predetermined in our current design, we also plan to investigate a scheme for automatically tuning these values based on desired bandwidth sharing behavior in the presence of other competing loss-based flows (e.g., loss-based TCP).

5.2 Calculating time-smoothed values

The receiver smoothes its observations via exponential averaging:

$$x_n = alpha*d_tilde + (1-alpha)*x_n.$$

The weighting parameter alpha adjusts the level of smoothing.

5.3 Sending periodic feedback

Periodically, the receiver sends back the updated value of x in RTCP messages, to aid the sender in its calculation of target rate. The size of acknowledgement packets are typically on the order of tens of bytes, and are significantly smaller than average video packet sizes. Therefore, the bandwidth overhead of the receiver acknowledgement stream is sufficiently low.

5.4 Discussions on one-way delay measurements

At the current stage, our proposed scheme relies on one-way delay (OWD) as the primary form of congestion indication. This implicitly relies on well-synchronized sender and receiver clocks, e.g., due to the presence of an auxiliary clock synchronization process. For deployment in the open Internet, however, this assumption may not always hold.

There are several ways to get around the clock synchronization issue by slightly tweaking the current design. One option is to work with relative OWD instead, by maintaining the minimum value of observed OWD over a longer time horizon and subtract that out from the observed absolute OWD value. Such an approach cancels out the fixed clock difference from the sender and receiver clocks, and has been widely adopted by other delay-based congestion control approaches such as LEDBAT [RFC6817]. As discussed in [RFC6817], the time horizon for tracking the minimum OWD needs to be chosen with care: long enough for an opportunity to observe the minimum OWD with zero queuing delay along the path, and sufficiently short so as to timely reflect "true" changes in minimum OWD introduced by route changes and other rare events.

Alternatively, one could move the per-packet statistical handling to the sender instead, and use RTT in lieu of OWD, assuming the per-packet ACKs are present. The main drawback of this latter approach, on the other hand, is that the scheme will be confused by congestion in the reverse direction.

Note that either approach involves no change in the proposed rate adaptation algorithm at the sender. Therefore, comparing the pros and cons regarding which delay metric to use can be kept as an orthogonal direction of investigation.

6. Sender Behavior

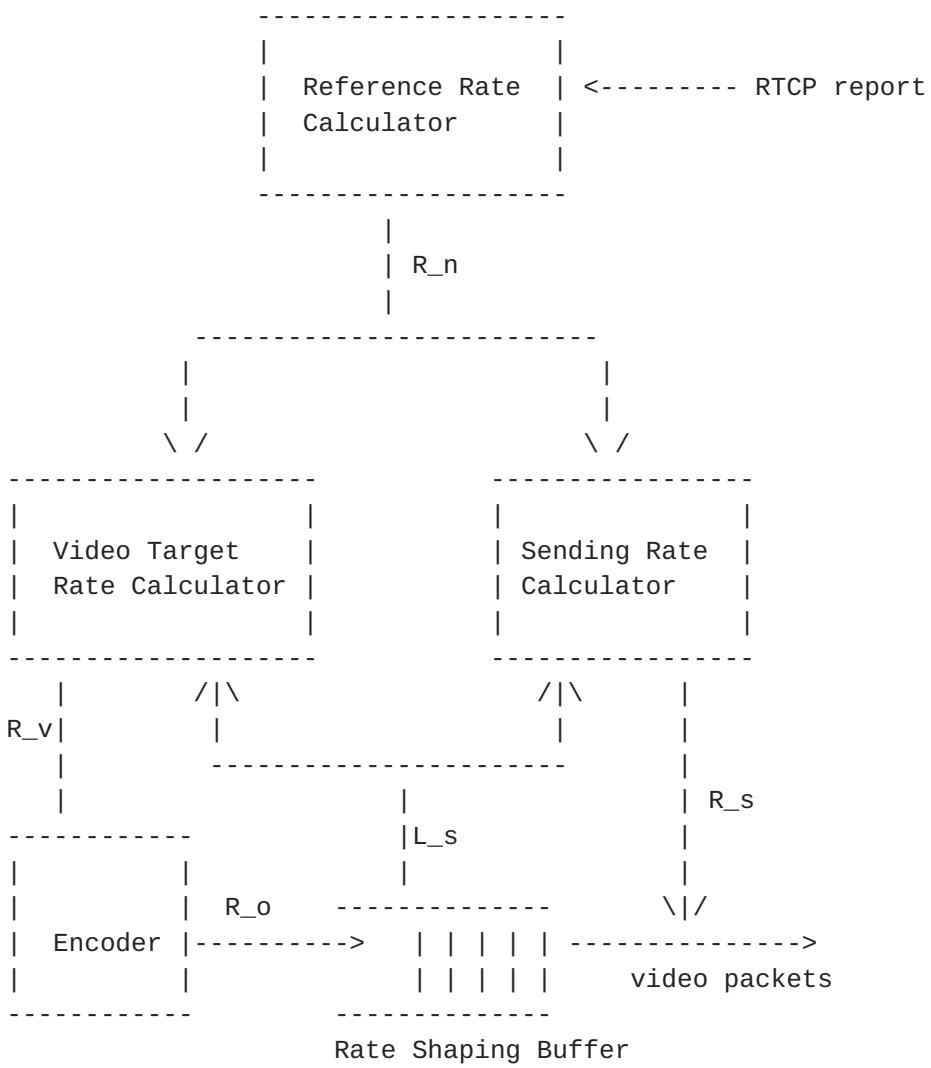

Figure 1 NADA Sender Structure

Figure 1 provides a more detailed view of the NADA sender. Upon receipt of an RTCP report from the receiver, the NADA sender updates its calculation of the reference rate R_n as a function of the network congestion signal. It further adjusts both the target rate for the live video encoder R_v and the sending rate R_s over the network based on the updated value of R_n, as well as the size of the rate shaping buffer.

The following sections describe these modules in further details, and explain how they interact with each other.

6.1 Video encoder rate control

The video encoder rate control procedure has the following characteristics:

* Rate changes can happen only at large intervals, on the order of seconds.

* Given a target rate R_o, the encoder output rate may randomly fluctuate around it.

* The encoder output rate is further constrained by video content complexity. The range of the final rate output is [R_min, R_max]. Note that it's content-dependent, and may change over time.

Note that operation of the live video encoder is out of the scope of our design for a congestion control scheme in NADA. Instead, its behavior treated as a black box.

6.2 Rate shaping buffer

A rate shaping buffer is employed to absorb any instantaneous mismatch between encoder rate output R_o and regulated sending rate R_s. The size of the buffer evolves from time t-tau to time t as:

$$L_s(t) = \max [0, L_s(t-\tau)+R_v*\tau-R_s*\tau].$$

A large rate shaping buffer contributes to higher end-to-end delay, which may harm the performance of real-time media communications. Therefore, the sender has a strong incentive to constrain the size of the shaping buffer. It can either deplete it faster by increasing the sending rate R_s, or limit its growth by reducing the target rate for the video encoder rate control R_v.

6.3 Reference rate calculator

The sender calculates the reference rate R_n based on network congestion information from receiver RTCP reports. It first compensates the effect of delayed observation by one round-trip time (RTT) via a linear predictor:

$$x\_hat = x\_n + \frac{x\_n - x\_n\text{-}1}{delta} * tau\_o \qquad (1)$$

In (1), the arrival interval between the (n-1)-th the n-th packets is designated by delta. The parameter tau_o is pre-configured to a fixed value. Typically, its value is comparable to the RTT experienced by the flow, but does not needs to be an exact match. Throughout all our simulation evaluations (see [Zhu-PV13]), we have been using the same fixed value of tau_o = 200ms.

The reference rate is then calculated as:

$$R\_n = R\_min + w * \frac{R\_max\text{-}R\_min}{x\_hat} * x\_ref \qquad (2)$$

Here, R_min and R_max denote the content-dependent rate range the encoder can produce. The weight of priority level is w. The reference congestion signal x_ref is chosen so that the maximum rate of R_max can be achieved when x_hat = w*x_ref. The final target rate R_n is clipped within the range of [R_min, R_max].

The rationale of choose x_ref to be the value of absolute one-way delay (i.e., only propagation delay along the path) is that ideally, we would want the video stream to reach the highest possible rate when the queue stays is empty, e.g., when bottleneck link rate exceeds R_max of video. In practice, the stream can simply set x_ref to be the minimum value of OWD observed over a long time horizon. Note also that the combination of w and x_ref determines how sensitive the rate adaptation scheme is in reaction to fluctuations in observed signal x.

The sender does not need any explicit knowledge of the management scheme inside the network. Rather, it reacts to the aggregation of all forms of congestion indications (delay, loss, and marking) via the composite congestion signal x_n from the receiver in a coherent manner.

6.4 Video target rate and sending rate calculator

The target rate for the live video encoder is updated based on both the reference rate R_n and the rate shaping buffer size L_s, as follows:

$$R_v = R_o - beta_v * \frac{L_s}{tau_v}. \quad (3)$$

Similarly, the outgoing rate is regulated based on both the reference rate R_n and the rate shaping buffer size L_s, such that:

$$R_s = R_o + beta_s * \frac{L_s}{tau_v}. \quad (4)$$

In (3) and (4), the first term indicates the rate calculated from network congestion feedback alone. The second term indicates the influence of the rate shaping buffer. A large rate shaping buffer nudges the encoder target rate slightly below -- and the sending rate slightly above -- the reference rate R_n. Intuitively, the amount of extra rate offset needed to completely drain the rate shaping buffer within the same time frame of encoder rate adaptation tau_v is given by L_s/tau_v. The scaling parameters beta_v and beta_s can be tuned to balance between the competing goals of maintaining a small rate shaping buffer and deviating the system from the reference rate point.

6.5 Slow-start behavior

Finally, special care needs to be taken during the startup phase of a video stream, since it may take several roundtrip-times before the sender can collect statistically robust information on network congestion. We propose to regulate the reference rate R_n to grow linearly in the beginning, no more than: R_ss at time t:

$$R_{ss}(t) = R_{min} + \frac{t-t_0}{T}(R_{max}-R_{min}).$$

The start time of the stream is t_0, and T represents the time horizon over which the slow-start mechanism is effective. The encoder target rate is chosen to be the minimum of R_n and R_ss during the first T seconds.

7. Incremental Deployment

One nice property of proposed design is the consistent video end point behavior irrespective of network node variations. This facilitates gradual, incremental adoption of the scheme.

To start off with, the proposed encoder congestion control mechanism can be implemented without any explicit support from the network, and rely solely on observed one-way delay measurements and packet loss ratios as implicit congestion signals.

When ECN is enabled at the network nodes with RED-based marking, the receiver can fold its observations of ECN markings into the calculation of the equivalent delay. The sender can react to these explicit congestion signals without any modification.

Ultimately, networks equipped with proactive marking based on token bucket level metering can reap the additional benefits of zero standing queues and lower end-to-end delay and work seamlessly with existing senders and receivers.

8. Implementation Status

The proposed NADA scheme has been implemented in the ns-2 simulation platform [ns2]. Extensive simulation evaluations of the scheme are documented in [Zhu-PV13].

The scheme has also been implemented in Linux. Initial set of testbed evaluation results have focused on the case of a single NADA flow over a single low-delay bottleneck link. More investigations are underway.

9. IANA Considerations

There are no actions for IANA.

10. References

10.1 Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC3550] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", STD 64, RFC 3550, July 2003.

10.2 Informative References

[RFC3168] Ramakrishnan, K., Floyd, S., and D. Black, "The Addition of Explicit Congestion Notification (ECN) to IP", RFC 3168, September 2001.

[RFC2309] Braden, B., Clark, D., Crowcroft, J., Davie, B., Deering, S., Estrin, D., Floyd, S., Jacobson, V., Minshall, G., Partridge, C., Peterson, L., Ramakrishnan, K., Shenker, S., Wroclawski, J., and L. Zhang, "Recommendations on Queue Management and Congestion Avoidance in the Internet", RFC 2309, April 1998.

[RFC6187] S. Shalunov, G. Hazel, J. Iyengar, and M. Kuehlewind, "Low Extra Delay Background Transport (LEDBAT)", RFC 6817, December 2012

[ns2] "The Network Simulator - ns-2", http://www.isi.edu/nsnam/ns/

[Zhu-PV13] Zhu, X. and Pan, R., "NADA: A Unified Congestion Control Scheme for Low-Latency Interactive Video", in Proc. IEEE International Packet Video Workshop (PV'13). San Jose, CA, USA. December 2013.

Authors' Addresses

Xiaoqing Zhu
Cisco Systems,
510 McCarthy Blvd,
Milpitas, CA 95134, USA
EMail: xiaoqzhu@cisco.com

Rong Pan
Cisco Systems
510 McCarthy Blvd,
Milpitas, CA 95134, USA
Email: ropan@cisco.com