Network-Assisted Dynamic Adaptation (NADA): A Design Summary

Xiaoqing Zhu and Rong Pan
Cisco Systems Inc.

ABSTRACT

We present Network-Assisted Dynamic Adaptation (NADA), a design for video endpoint rate adaptation and error resiliency. The proposed scheme highlights the benefit of explicit congestion notification from network nodes, yet maintains consistent and robust endpoint behaviors even in the absence of such explicit information. This documentation briefly describes the overall NADA system architecture, recommended behaviors at the video sender and receiver, and expected behavior at the network nodes.

I. SYSTEM OVERVIEW

Figure 1 provides an overview of the proposed NADA scheme. Each video sender adjusts its rate in a distributed manner, by reacting to receiver RTCP reports on aggregate network congestion level. Depending on the capabilities of the network node, the congestion level information can take the form of either end-to-end delay measurement, ECN marking ratios, or PCN marking ratios. In the following, we will elaborate on the respective behaviors at the network nodes, as well as at the senders and receivers.

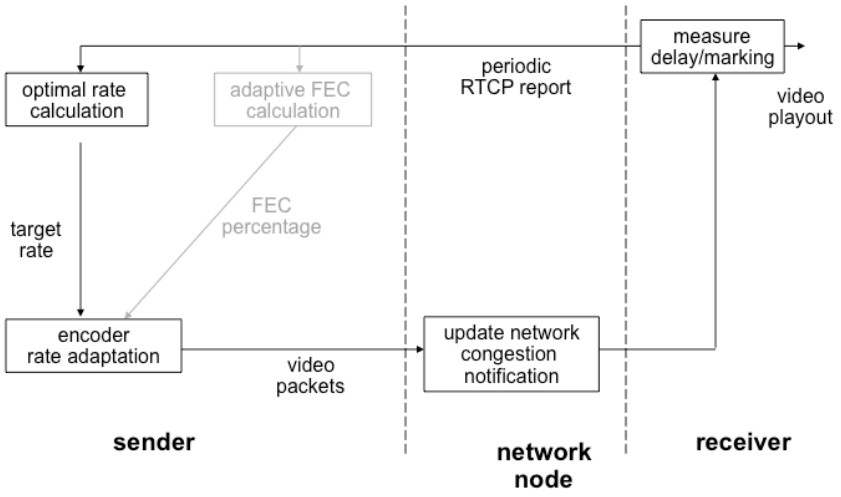


Fig. 1. NADA system overview.

II. NETWORK NODE BEHAVIOR

We consider three variations of queuing behaviors at the network node, which provide an indication of network congestion in either an explicit or implicit manner.

A. Delay-based

Network congestion is conveyed via the implicit information of queuing delay. No specific action is needed at the network node, except to confining the overall queuing delay within limits of video conferencing applications.

B. ECN-marking

The network node randomly marks the ECN field in the IP packet header following the Random Early Detection (RED) algorithm. Basically, the marking probability p grows linearly with respect to observed queue length, with exponential averaging over time, at the node.

C. PCN-marking

The network node randomly marks the ECN field in the IP packet header, using a token-bucket algorithm. Choice of the token bucket rate can be slower than the physical link rate, e.g., 90% of capacity. This effectively leads to probabilistically marking the packets according to the length of a *virtual* queue, thereby leads to zero standing queues at steady state.

In all three flavors, the network queue operates with the simple first-in-first-out (FIFO) discipline, without any need of maintaining per-flow states. Such a simple design allows the system to scale easily with large number of video flows and high link rates.

III. Sender Behavior

The video sender is composed of three main modules: a) video encoder rate controller; b) rate shaping buffer; and c) sending rate calculator. Figure 2 provides an overview of how these three components interact with each other. In the following, we will describe the behavior of each component inside the sender in further details.

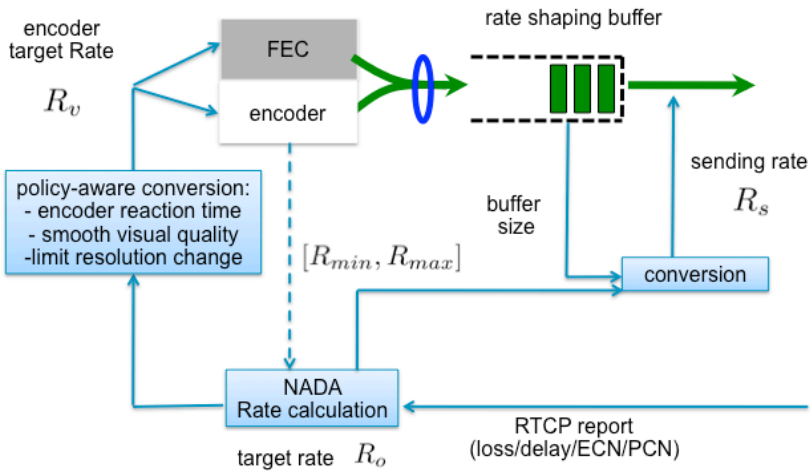


Fig. 2. NADA sender structure.

A. Video encoder rate controller

The video encoder acts as a traffic generator that periodically takes in a target rate R_v, and sends out traffic with a rate that randomly fluctuates around the target. The interval over which the rate controller settles on a new target is defined as τ_f. Typically, the encoder output rate is further constrained by scene complexity, confining the final rate to range within $[R_{min}, R_{max}]$. Note that the rate range may also change over time, alone with changing video contents.

B. Rate shaping buffer

The rate shaping buffer size L_s evolves over time, as $L_s(t) = \max[0, L_s(t-\tau)+8B_v-R_s\tau]$. Here, B_v denotes encoded video frame size in bytes, and τ denotes the video frame interval. The buffer draining rate corresponds to the sending rate R_s. Its value is periodically calculated following the steps described in the next session.

C. Sending rate calculator

The sending rate calculation is broken down into two stages: slow-start and steady update.

1) Slow-start: The initial sending rate is regulated to grow linearly, to no more than R_{ss}: $R_{ss}(t) = R_{ss}(t-t_0) + \frac{t-t_0}{T}(R_{max} - R_{min})$. Here, the start time of the stream is t_0, the start rate $R_{ss}(t_0) = R_{min}$, and T represents the time horizon over which the slow-start mechanism is effective.

2) Steady update: Once a stream with a priority weight of w gets out of slow-start, calculation of the sending rate R_s in the steady update stage is as follows:

- Retrieve either the ECN/PCN marking ratio m_{avg}, or end-to-end relative packet delivery delay d_{avg}, from the last RTCP report;
- Obtain sequence-specific parameters for the stream in consideration, including R_{min}, R_{max}, and w;
- Calculate the ideal target rate R_o based on network feedback m_{avg} or d_{avg}, as well as rate range $[R_{min}, R_{max}]$ and priority weight w.
- The sending rate closely follows R_o, while balancing the pressure from rate shaping buffer build-up.

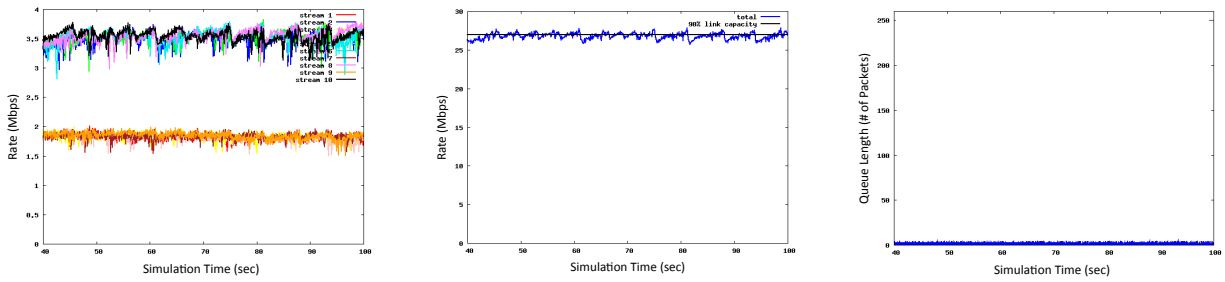
It can be shown that by following the steps in the above two stages, the final rates chosen for each stream will be in proportion with respect to their relative rate dynamic range $(R_{max} - R_{min})$'s, as well as their priority weight w's. This guarantees weighted fairness sharing among all competing streams.

IV. Receiver Behavior

The task of the receiver is relatively straightforward. It keeps a running average of the observed end-to-end packet statistics in terms of delay, loss, and marking ratios, and periodically feeds back such information via RTCP reports to the sender.

V. Example Results

We show an example of 10 streams competing over a bottleneck link of 30Mbps, acting on PCN markings from the network. Half of the streams have a rate range between 1Mbps and 3Mbps, whereas the other half have a rate range between 2Mbps and 6Mbps. All streams have the same priority weight. Target link utilization for PCN marking is chosen as 90%. It can be observed from Fig. 3 that the streams with wider rate range achieve twice allocated rate as others. In addition, the total rate of all streams reaches the target utilization, while maintaining close to zero queuing delay. Results from more extensive studies of the NADA system are available upon request.



(a) Per-flow rate (b) Total incoming rate (c) Queue occupancy

Fig. 3. Ten flows competing over a bottleneck link of 30 Mbps.

VI. Discussions and Future Work

We have presented three different flavors of NADA, depending on the available features of network node behaviors. The delay-based version of the scheme can be implemented without any network support. The ECN-based version only requires RED-based ECN marking, a feature already supported by many routers and switches today. Ultimately, networks equipped with proactive marking based on token bucket level metering can reap some additional benefits: zero standing queues and more smooth streaming rates. On the other hand, behaviors of video endpoints stay fairly consistent and robust, regardless of which flavor of the scheme is chosen by network nodes. This is helpful for gradual adoption of the framework.

As illustrated in Fig. 1 and Fig. 2, the same network congestion feedback can also be used for guiding the choice of forward error correction (FEC) protection strength. We have documented such possibility in our prior research [1]. The next step of the NADA project can be the investigation of such network-assisted adaptive error protection schemes within the same framework.

References

[1] X. Zhu, R. Pan, M. S. Prabhu, N. Dukkipati, V. Subramanian, and F. Bonomi, "Layered internet video adaptation (liva): Network-assisted bandwidth sharing and transient loss protection for video streaming," *IEEE Transactions on Multimedia*, vol. 4, no. 13, pp. 720–732, Aug. 2011.