Thin Stream: An Architecture for Multicasting Layered Video

Xie Qing

1 Summary

This paper defines a new architecture using layered video compression schemes with multiple multicast groups for scaleable video delivery. The authors try to solve 2 problems: 1. How to deal with the expensive failed experiment. 2. How to make the join/leave experiment scalable (avoid interference across sessions).

To solve first problem, first they split the typically thick stream into multiple equally "thin" stream. It reduces the sending rate, make the join experiment use a smaller bandwidth. Secondly, they try to use differences between expected throughput and actual throughout rather than packet loss (Mccanne) to reduce the time for detecting a failed join experiment.

They define a formula to compute the actual throughput and expected throughput.
And their join/leave algorithm is based on the difference between Expected and actual throughput. Using join-threshold, leave-threshold and hold_off_time to decide when to join or leave. Make receiver joined few groups more aggressively than those joined more groups. By carefully select these three parameter, the algorithm can achieve: fairness bandwidth among different session.

To solve the second problem, they use "thinstream" clock pulse to synchronize join experiments within session and desynchronize join experiments across session.

Some experiments with different network topology is conducted to verify their architecture. SPHIT, HVQ can already use the thinstream, JPEG,MPEG,H.261,Haar need some more work to exploit the architecture. They also mentioned some optimization is available for tuning the thinstream.

2 Comments

Some drawbacks of the paper: It requires lots of multicast addresses, some codec like JPEG,MPEG didn't support for this. They also didn't solve the format heterogeneity problem. I understand how to conduct experiment less expensively and how to minimize the experiment interference. The using of clock pulse can be applied elsewhere to "synchronize" and "desynchronize". Detect a failed experiment before real packet loss occurred is also useful in other places. I like this paper that it did a good job to solve the expensive failed experiment and make the experiment scalable. However they didn't do a comparison with other similar solutions.