Paper Analyses

Paper: "A Digital Fountain Approach to Reliable Distribution of Bulk Data"

Some applications sends bulk data to a large number of autonomous clients, and need this to be done in an a reliable fashion. The authors say that these applications call for a design of multicast and broadcast protocols, and present an ideal, fully scalable protocol. A real life example of usage is the distribution of new software.

Tornado codes scale better as the file size increases. This is due to the number of packets that has to be received in order to rebuild the source data only grows linearly to the size of its data, as an interleaved approach grow super-linearly to the size of the source data.

Regarding congestion control, the protocol follows the lead of layered multicast: server transmits data over multiple layers where the layers are ordered by increasing transmission rate, and the layers are cumulative.

The requirements listed are reasonable and a good base for the paper. It has to reliable, efficient, on demand and tolerant.

I did not find the introduction to be very motivating. It has an example and describes some problems. As good as it may sound, it did not really get to me.

The paper is heavy to read with a lot of formulas and numbers. This makes it harder to get an overview and the reader loses his/her flow. However, some of the tables make it better: table 2 and 3 are great to get a summary of encoding/decoding times.

Like the authors mentions in the conclusion, it would be of interest to test a similar system with a large number of users and see how it performs, in terms of effectiveness. In periods with delay due to congestion, the destination can recover the data when a sufficient number of packets have arrived, irrespective of the paths they took. The drawback of Tornado codes is that, if the stretch factor is low, then you would receive duplicate packets frequently. On the other hand, if the stretch factor is high, the space and time requirements for decoding become prohibitive.

Paper Analyses

Paper: "An Alternative Paradigm for Scalable On-Demand Applications: Evaluating and Deploying the Interactive Multimedia Jukebox"

Interesting paper about the new paradigm Interactive Multimedia Jukebox (IMJ). Services like PPV (pay-per-view) and VoD (video on-demand) were exciting topics in the 80's, but suffered from technical problems and expensive systems. This paper provides a better, scalable system.

VoD had too little demands, and was not cheap either, making the service providers lack motivation. There are two types of VoD: true VoD and near VoD. True VoD is given on a one-to-one basis (all users get their own resources), and the scheduling is made in real-time. Systems with true VoD, require a lot of performance and therefore very expensive. A less expensive solution is near VoD. This scheme batches multiple requests by adding an artificial delay. It is, however, important that the wait-time is rather short, as long wait-times can be disastrous and make the user experience really unpleasant. Needed performance is lower and it is a scalable alternative to true VoD. Near VoD needs large viewer populations to achieve sufficient economies of scale. True VoD would probably not the very profitable as every user would require their own program stream, and renting a movie is quite cheap.

The jukebox approach is, not surprisingly, based on the music jukebox. Everyone listens and everyone is able to make requests. Multiple, distinct channels are used, and it is scalable (which is done by using a fixed number of channels). It also provides flexibility in request scheduling and is independent of a network's topology. As is turns out, smaller videos were the most popular. Which also happens to be the case at YouTube.

A good deal of figures and tables illustrates the results from the tracking. The results were obtained during a one and a half year period. Among others, figure 6 and table 2 have a rather short explanation. Some deeper insight would be good. On the other side, most of the figures are great. Especially figure 7, it presents a good overview of the wait time results.

An interesting observation is that the more programs in the library and the lower the skew value, the probability that multiple requests will arrive for the same program decreases. Also, the average wait time decreases significantly by adding channels before it flattens as the wait time approaches 0.

By using votes, a scheduling of a request will not start until a minimum number of votes have been obtained (unless some channels are free).

The assumption about users being well behaved, are not necessarily a good one. However, they do provide an idea to fix this problem: charge the not-well-behaved users more, which is a reasonable approach.

In addition, they made a simulation environment. IMJ gives the service providers an advantage by providing the opportunity to influence uses choices. It also allows for further performance improvements.