# Predicting connection quality in peer-to-peer real-time video streaming systems

Alex Giladi
Jeonghun Noh
Information Systems Laboratory, Department of Electrical Engineering
Stanford University, Stanford, CA 94305
Email: {agiladi, jhnoh}@stanford.edu

## Introduction

In server-based video streaming systems, the content provider transmits a video stream to each client (e.g. PC, PDA, set-top box, etc). However, peer-to-peer (P2P) streaming systems provide a more efficient alternative, utilizing the uplink bandwidth of each participating peer. In server-based streaming systems the number of media servers required to serve an audience grows linearly with the number of viewers. However, the P2P approach is self-scaling, since each peer acts as a media server.

In our P2P video streaming system, multiple multicast trees are constructed as a distribution structure. A video source (or the server) is the root of all trees, while other participating peers at either intermediate nodes or leaf nodes. When a new peer joins a video multicast group, it needs to find parents, i.e. nodes from which it will be receiving its video input.

The JOIN stage of connection establishment protocol, each joining peer contacts the server to obtain a list of randomly chosen connected peers. The joining peer contacts all the members of the list and waits for replies. From the replies, peers which report to have enough throughputs to support an additional peer are considered as parent candidates. Several additional criteria (discussed below) can also be considered in the process of selecting a parent. Once the parents are chosen for each tree, the joining peer sends out connection requests to them and waits until it is accepted as a child.

So far, in many literatures, the number of peer nodes separating a peer from the video source has been extensively used a parent selection metric. Thus well-balanced trees are constructed, minimizing the likelihood of connection disruption due to loss of one of the intermediate peers. However, congestion, bandwidth utilization and connection quality can be neglected in such tree construction scheme.

To develop a new parent selection algorithm, we propose a connection prediction method based on several additional connection-related parameters as a preliminary work. This method will predict connection quality from peers, and shows the best connection that minimizes the likelihood of video packet loss, which should correspond to better perceived video quality. We will use supervised learning to devise such a metric, which will be used by each peer for selecting its parent. Since peers run in real time, we operate under complexity and memory usage constraints.

## System

### *Simulation*

We have developed our peer-to-peer video streaming system as an extension module in the ns-2 network simulator. Since simulations can be run at our disposal, it is rather easy to create data sets from many different point of views. In other words, adding or removing features is simple in the sense that we run a new simulation by modifying the code and produce or suppress relevant observation values in the simulation output.

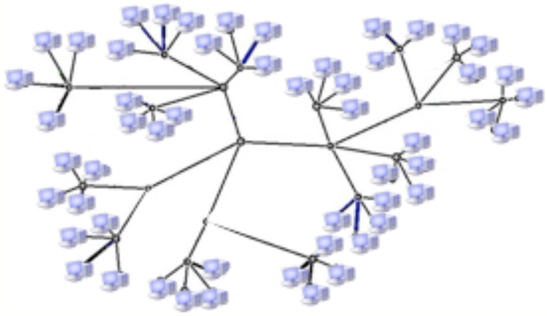

Figure 1: Topology used in the simulation

| Downlink | Uplink | Percentage |
|---|---|---|
| 512 kb/s | 256 kb/s | 56% |
| 3 Mb/s | 384 kb/s | 21% |
| 1.5 Mb/s | 896 kb/s | 9% |
| 20 Mb/s | 2 Mb/s | 3% |
| 20 Mb/s | 5 Mb/s | 11% |

**Table 1: Bandwidth distribution**

Each simulation consists of one video source and 300 peers. Peers have heterogeneous bandwidth described in the table 1. Multiple multicast trees are constructed to distribute video packets from the source to all other peers using our distributed p2p protocol. The network topology used in the simulation is depicted in the figure 1. The simulation length is 15 minutes. We ran several simulations to obtain raw data. Then, post-processing is done to generate the final training data sets by evaluating the performance statistics of each selected parent. As peers join and leave the group randomly according to Poisson process, we collect data from session lengths larger than 45 seconds to reduce the effect of different length of sessions. Also, we removed the sessions from the initial period when peers only join and none of them leave. Here, we define a session as a block of time in which a child peer is connected to a parent peer. When disconnect from existing parent peers occurs, the child peer starts a new session with a new parent peer. The following figure 2 shows the cumulative distribution of the session lengths obtained from the simulation results.

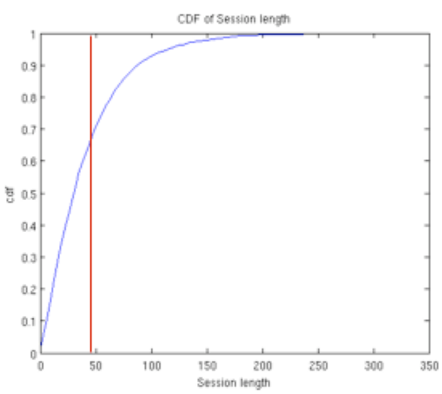


**Figure 2: Distribution of session lengths**

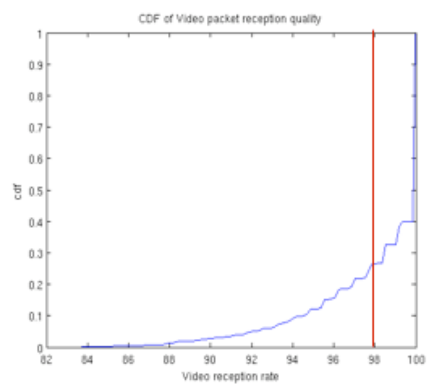


**Figure 3: Distribution of video reception rate of peers**

## *Input features*

We obtain the following measurements for data sets:

- **Connection time:** session time for which the parent-child connection is active
- **Hop count:** number of peers in the logical path from the source to the peer
- **Number of children**: total number of immediate children of the potential parent (i.e. on all trees)
- **Bandwidth:** peer's uplink bandwidth
- **Network Jitter:** jitter in video packets inter-arrival times
- **RTT**: round-trip time to the parent at the time connection was established
- **Input video stream quality**: number of received packets divided by the number of packets sent out from the source over T seconds (T is set to 45 seconds)
- **Output video stream quality**: number of transmitted packets to a specific peer divided by the number of packets enqueued into the output queue

## *Output of the learning algorithm*

Evaluation of each peer, in other words, to predict connection quality that will be given by the peer is based on the features obtained from the peers as input and the label we put on such features. Since the timely reception of video packets is critical for real-time video streaming, we define in-time packet reception rate as the label.

While loss probability estimation seems to be a regression problem, using regression techniques can result in several problems. It has been shown that randomization of parent selection is important for maintaining balanced multicast trees. Moreover, reducing the subtree size makes the system more resilient to peer disconnections. On the other hand, the degree of precision of a regression technique will lead to a very deterministic performance, since all peers will try to attach to the same 'best' parents. Moreover, the training data might have a certain amount of outliers if not discretized. In order to avoid this case, we chose to define the problem as a multi-class classification problem, where 4-8 discrete classes of loss probability might be defined. However, due to the way the quality values are distributed (where more than half of peers report 100%-98% quality), we reduced this further into a binary classification problem.

# Experiment, results and discussion

The P2P system and the data loggers were implemented as plug-ins for the NS-2 network simulator. The generated data was preprocessed using a C++ program, which modeled the system from the log data generated on NS-2. The program summarized the data (so that a single training record reflects a single parent-child relationship), rescaled it, and calculated few statistics.

## *Feature selection using logistic regression*

We used logistic regression for feature selection. To see which feature is the most important, we filtered out the unnecessary features based on their correlation with the ground-truth values and between themselves.

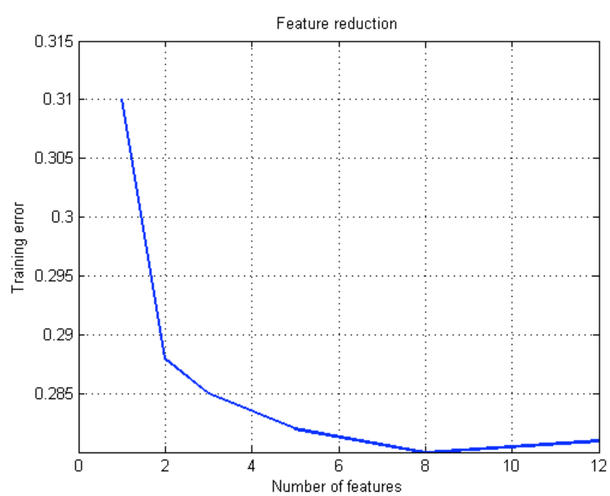


**Figure 4: Training error with different feature sets**

Several stages of filtering brought the number of features down to the following three, with a negligible increase in training error (as can be seen from the picture above):

- Mean number of immediate children
- Hop distance to the source
- Uplink bandwidth

One reasoning behind this result is as follows: The more peers register as children, the less resources will be left to a child, which will cause a 'fan-out effect', when all the peers will be equally starved. The influence of the logical hops from the source can be easily explained by the fact that the probability of a packet arrival is $\prod_{i=0}^{i<D}(1-p_{loss}(i))$, for distance $D$, where i is the link label.

The most of the training sets we used showed significant correlation between the number of children and hop distance, however we are unsure whether this would also be the case in significantly larger (several orders of magnitude) data sets that we are unable to simulate. We tried all possible subsets of the three latter features, while in most cases the performance suffered immediately.

### *Binary classification using SVM*

Using the features selected by the previous method, we utilized SVM for a binary classification to distinguish good peers and fair peers.

| | **BW only** | **BW/nHops /nChild** | **BW/nHops/nChild/last nHops/last nChild** |
|---|---|---|---|
| Training error | 26.63% | 26.55% | 22.87% |
| Testing error | 28.99% | 28.96% | 26.42% |

**Table 2: Bandwidth distribution. BW stands for peer uplink bandwidth, nHops for number of logical hops to source, nChild for number of immediate child peers, last for the last report from peers**

We tried several different combinations of features and the results are shown in Table 2 above. If a peer wants to use only one metric to predict a connection, the table suggests the uplink bandwidth of the parent peer can act as the most influential single factor.

When the combination of the features suggested by the logistic regression results is used, it achieves the lower testing error, but the difference is negligible. Whereas, when we try the last combination, it outperforms the other cases, including all possible choices of the features not shown in the table. In fact, this leads us to a further investigation with a larger feature with similar characteristics of peers. In other words, sampling the same feature from the peers can be a good measure for predicting connection qualities. In the table, 'last' means the last report from the parent peers.

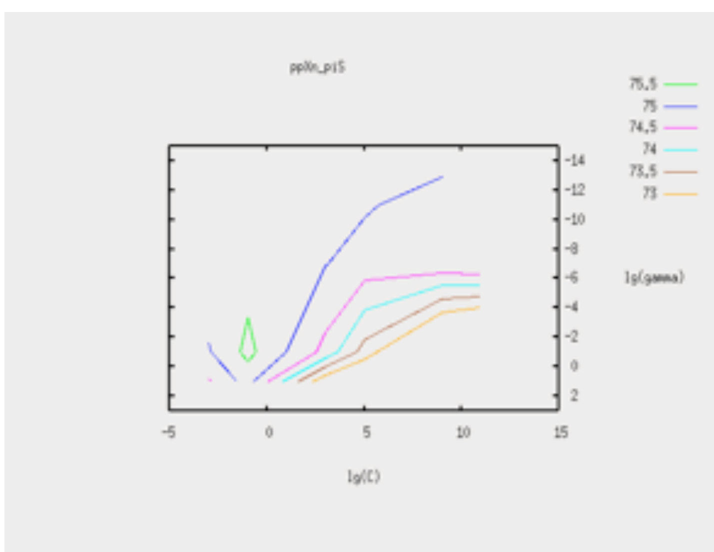


**Figure 5: Choosing the optimal Gamma and C**

Figure 5 shows the optimal gamma and C values used in SVM. We adopted the Gaussian kernel with the L1 regularization. Using the optimal kernel parameters, about 2% of the error rate can be reduced.

## Conclusion

We have seen that connection quality can be well predicted using the statistics of its potential parent peers at the time of connection establishment. The most important statistics were the peer bandwidth, number of immediate children and hop distance from the source (i.e. depth level). Our prediction performance is fair, but not excellent. However, it is good enough for our purposes. The algorithm provides a child a relatively large list of parents. Thus, the probability of selecting a misclassified parent is still relatively small. Also, getting the same features over time can lead to better prediction on connection qualities. It is also inferred from the fact that the relatively high errors both in the training and testing sets may result from high bias in the mode selection. With further investigation, this work may lead to develop a new method of parent selection algorithm using connection prediction supported by this learning algorithm.

## Reference

E. Setton, J. Noh, and B. Girod, "Rate-Distortion Optimized Video Peer-to-Peer Multicast Streaming," Workshop on Advances in Peer-to-Peer Multimedia Streaming at ACM Multimedia , pp. 39-48, Nov. 2005, Singapore, invited paper.