2008

Variable Neighbor Selection in Live Peer-to-Peer Multimedia Streaming Networks

Jagannath Ghoshal
Sprint Nextel, Overland Park, KS

Miao Wang
University of Nebraska-Lincoln

Lisong Xu
University of Nebraska-Lincoln, xu@cse.unl.edu

Byrav Ramamurthy
University of Nebraska-Lincoln, bramamurthy2@unl.edu

BROADNETS 2008. 5th International Conference on Broadband Communications, Networks and Systems, 2008.
Digital Object Identifier: 10.1109/BROADNETS.2008.4769101
Publication Year: 2008 , Page(s): 344 - 346

Variable Neighbor Selection in Live Peer-to-Peer Multimedia Streaming Networks

Jagannath Ghoshal, Miao Wang, Lisong Xu and Byrav Ramamurthy

***Abstract*—Data-driven (or swarming based) streaming is one of the popular ways to distribute live multimedia streaming traffc over Peer-to-Peer (P2P) networks. The efficiency and user satisfaction highly depend on the constructed overlays. The common neighbor selection algorithms in existing overlay construction schemes usually randomly select a fixed number of neighbors which satisfy the selection requirements, such as end-to-end delay or a peer's sojourn time. However, this fixed random neighbor-selection algorithm (*FRNS*) neglects the peers' upload bandwidth heterogeneity and therefore, the upload bandwidth cannot be efficiently used. In this paper, we propose a variable random neighbor-selection (*VRNS*) scheme to alleviate the problems due to bandwidth heterogeneity, and in which the number of neighbors with different upload bandwidths is dynamically determined by the statistical bandwidth information of the system. Our proposed scheme is shown to outperform *FRNS* based upon a large volume of carefully designed simulations.**

***Index Terms*—Peer-to-peer (P2P) networks, multimedia streaming, neighbor selection, FRNS, VRNS.**

I. Introduction

THERE is a significant difference between live multimedia streaming and file downloading systems, which are the most widely used P2P systems in the Internet. Live multimedia streaming has a tight delay constraint in that the playback starts soon after the streaming begins and the stream should be played back continuously; whereas file downloading has no such requirement on the downloading order of different blocks of a file. In addition, a file is accessed by a user only after the whole file has been downloaded. These differences required improvements to the architectural design of P2P file downloading protocols to readily address the timing constraints and to provide good media quality for P2P media streaming protocols.

In this paper, we focus on the neighbor selection algorithm, which is a part of overlay construction. Although this is a basic issue and the authors of [2] had proposed an optimal peer selection algorithm for both streaming and file sharing systems, the peers' upload bandwidth heterogeneity is neglected by the authors. This however, greatly affects the streaming quality, since upload bandwidth is the competitive resource in cooperative P2P streaming networks. Our proposed variable random neighbor-selection algorithm (*VRNS*) is bandwidth-aware, which means that the fraction and the maximum number of neighbors are dynamically determined based on the statistical bandwidth information of the system. Our goal is to efficiently utilize the upload bandwidth of all peers and balance the distribution of neighbors with different upload bandwidths at end peers.

Mr. Ghoshal is a Network Engineer with Sprint Nextel, Overland Park, KS 66212. Phone: +1 913 957–5064, e-mail: jagannath.ghoshal@sprint.com

Mr. Wang is a PhD Candidate with the Department of Computer Science and Engineering, Unversity of Nebraska-Lincoln, Lincoln, NE 68588. Phone: +1 402 326–8787, e-mail: mwang@cse.unl.edu

Dr. Xu is an Assistant Professor with the Department of Computer Science and Engineering, Unversity of Nebraska-Lincoln, Lincoln, NE 68588. Phone: +1 402 472–1053, e-mail: xu@cse.unl.edu

Dr. Ramamurthy is an Associate Professor with the Department of Computer Science and Engineering, Unversity of Nebraska-Lincoln, Lincoln, NE 68588. Phone: +1 402 472–7791, e-mail: byrav@cse.unl.edu

II. Problem Description

A major problem with *FRNS* style neighbor-selection is the unawareness of bandwidth heterogeneity, which means that peers with low bandwidth ("bandwidth" refers to "upload bandwidth" thereafter) have the same number of neighbors as the peers with high bandwidth. The drawbacks are twofold. On the one hand, low bandwidth peers are overloaded and become the bandwidth bottleneck [6] of the system; on the other hand, the bandwidth resources of high bandwidth peers are not fully utilized, since they have extra capacity to handle more peers.

One straightforward idea to solve this problem is to dynamically change the number of neighbors according to peers' bandwidth in the system. However, it is a challenging task, for the following reasons:

1. The system bandwidth information is required to seek the optimal operating point of determining the upper limit on the number of neighbors; otherwise, a peer may handle too few or too many peers, where the former case is a waste of its own bandwidth and the latter case is a waste of others' bandwidth;
2. A peer has limited visibility, which makes it difficult to efficiently obtain system bandwidth information in terms of low control overheads. Therefore, it can be stated as a decision problem below: given a peer with known bandwidth, how to determine the maximum number of neighbors that can be handled by this peer?

In the following sections, we describe the *VRNS* algorithm by addressing the above problems.

A. Framework of Variable Random Neighbor-Selection Algorithm

In this subsection, we will describe our framework of *VRNS* by explaining how to solve the two problems mentioned above. The basic idea behind solving the problems is to utilize system's statistical information to estimate the number of peers with similar upload bandwidths and then to determine the maximum number of neighbors based on the estimated information. In our framework, we take advantage of bootstrap servers, which are widely used in existing P2P systems, to collect required information with very low control overhead.

In our proposed P2P streaming system, each peer is required to not only register at a bootstrap server at the time of joining, but is also required to report its lifetime

to the bootstrap server when it departs. The lifetime of a peer is the time for which a peer stays in the system. As we are interested in only the population of each type of peers with a certain amount of upload bandwidth, the bootstrap server maintains only the statistical information of each type of peers. With the registration information, the bootstrap server estimates the average arrival rate for each type of peers, where the average arrival rate is the average number of peers arriving at the system within a unit of time. With the lifetime information, the bootstrap server estimates the average lifetime for each type of peers.

Note that the server only counts peers reporting departures and therefore the estimated population might be higher or lower than the true values, since servers cannot distinguish between leaving peers and failing peers. The accuracy can be improved if more control overhead is allowed. However, this method may overload servers and the accuracy depends on how frequently the keep-alive messages are sent. In our current framework, we do not use this method in order to reduce the control overhead.

In P2P streaming systems, peer arrivals and departures are more complex and random than ideal Poisson arrivals and Pareto lifetime distributions. With ARIMA [3], the future population is a linear combination of the historical stationary populations and random noises, where the number of stationary terms is defined as order p and the number of noise terms are defined as order q. To derive the stationary series, the difference technique is used and the order of differences is defined as d. Therefore, we only have to determine the parameters p, d, q for ARIMA with finite historical population data, which is well studied in [3].

III. Results

OUR simulation is based on six parameters namely, 'Average Playback Delay', 'Average Packet Delay', 'Average Hop Count', 'Average Quality', 'Total Eliminated Peers', and 'Request Success Ratio'. Each of these is related to the dynamic environment of a live P2P mesh-topology. Due to the page limitations of the paper we only present 'Average Hop Count', 'Average Quality', 'Total Eliminated Peers', and 'Request Success Ratio'. These characteristics help us to compare the performance of a live P2P network based on neighbor selection for the two algorithms. In every simulation, the statistics or performance of the network is monitored on all the online nodes available at every 10-second interval. Additionally after every 50 seconds of simulation time, the above parameters related to performance of the network are written to the output. It should be noted that the simulation time is not the same as the actual time. Note, in these sets of simulations, we do *NOT* simulate *ARIMA* estimation method, in order to focus on the effect of VRNS.

p2pstrmsim [1] is a discrete event-based simulator used to simulate the present work. For the underlying topology, it uses the random model of GT-ITM [4] to generate a topology with 2000 routers and sets delays proportional to the distance metric of the resulting topology within a range [5ms, 300ms].

The simulation is carried out with a network topology of 2500 nodes with a pull-based random protocol and a broadcasting peer sending data at a streaming rate of 300 Kbps with 2 Mbps upload bandwidth. The network consists of three types of peers and a requesting window of size 20 seconds (explained in [1]) in every peer. The total simulation duration for every run is set to 500 seconds. A random user behavior algorithm is used in order to simulate a dynamic network. Two environments namely, static and dynamic, are simulated with three different settings of bandwidth ratio distribution, which are 15%-25%-60%, 30%-40%-30% and 60%-25%-15% representing the number of different types of DSL/Cable peers. Due to the page limitations of the paper only the dynamic environment with 15%-25%-60% bandwidth ratio distribution is presented. The other scenarios/environments also perform similarly to the presented results.

A. Dynamic Environment:

The arrival rate (or joining rate) of the peers is set to 10 users per second. The churn rate of network is simulated by referring to the traces from 'Gridmedia' [5]. The elimination of a peer is based on a threshold condition, such as, a peer should have the downloading rate of more than 15 Kbps. All the results shown below are obtained by changing the number of neighbors assigned.

A.1 Average Hop Count

The packet delay and hop count of a packet are interrelated since a packet needs to travel several hops before it can reach the destination node from where it is requested. The propagation delay of a link is fixed in our scenario and therefore the performance of the average packet delay shows a similar trend to the performance of the average hop count in Figure 1. This signifies that, the hop count of a packet has decreased as a result of which the delay of the packet from the source node to the destination peer is also minimized.

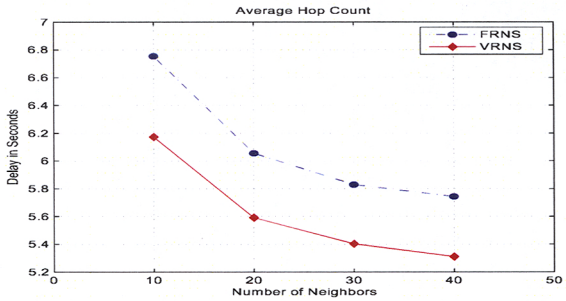

Fig. 1. Average Hop Count with Bandwidth Ratio of 15% 25% 60%

A.2 Average Success Ratio

Figure 2 show the performance of our algorithm in terms of 'Average Success Ratio' for 10 neighbors and the performance for 20, 30 and 40 neighbors are similar and hence

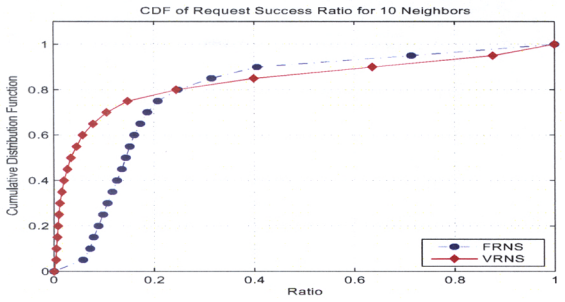

Fig. 2. CDF of Request Success Ratio with 10 Neighbors and Bandwidth Ratio of 15% 25% 60%

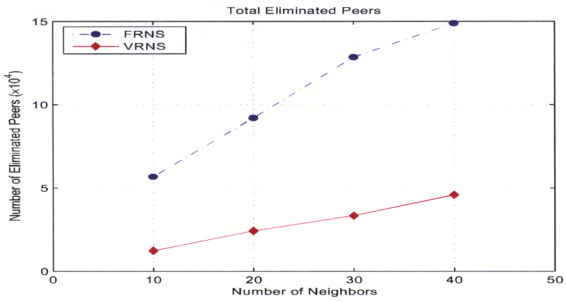

Fig. 4. Total Eliminated Peers with Bandwidth Ratio of 15% 25% 60%

not presented due to the limited length of the paper. In our simulation for a packet to be requested we set a request period of 1-6 seconds. If any packet requested by a peer is not received, then the peer can again request the same packet within this time period. The request is only made to its neighboring peers and not the whole network. If the packet is not available at its neighboring peer then these peers further send the request to their neighbors. However, this operation is only carried out if the packet is said to lie within the respective requesting window, otherwise the request is not met.

A.3 Average Quality

The quality or delivery ratio of a peer is calculated as the total number of blocks arriving at this peer before the playback deadline divided by the total number of blocks available in the encoded stream. The total number of streaming blocks remains constant and hence this value is affected based on encoding and packetization [7]. Fundamentally, it represents the throughput of this peer from the source node to itself. Thus the values obtained (as shown in Figure 3) is the average quality (or delivery ratio) of all the online peers in the network.

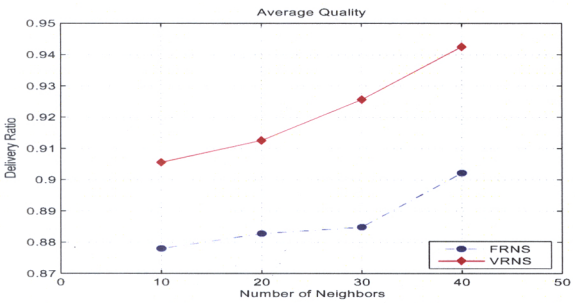

Fig. 3. Average Quality with Bandwidth Ratio of 15% 25% 60%

A.4 Peer Dynamics

The elimination of a peer is based on a threshold condition, such as that, a peer should have the minimum receiving rate of more than 15 Kbps when the streaming rate is set to 300 Kbps. As shown in Figure 4, our *VRNS* algorithm performs better (around 60% decrease) than *FRNS* and shows a consistent trend when we have an increase in the number of links to neighbors.. This is so because in the case of *FRNS*, the total number of links for a peer is fixed and so the low bandwidth peers have the same number of neighbors as the high bandwidth peers, which obviously becomes a problem in streaming for low bandwidth peers because the buffer size is the same for all the peers in the network. So if these peers have large buffer sizes in order to stream the data properly, then they would suffer from large packet delays. But in this work we maintain the same buffer size and try to achieve a better streaming rate at individual peers. Thus *VRNS* distributes these links appropriately based on the ratio, which results in less number of peers getting eliminated.

IV. Conclusion

LIVE P2P streaming networks, as a promising Internet application, have attracted a lot of research interest. In this paper, we have proposed the *VRNS* neighbor-selection algorithm for unstructured, mesh-based data-driven networks, which has been shown to outperform the bandwidth heterogeneity unaware *FRNS* algorithm through a large volume of simulations. In the future, we plan to evaluate our proposed method in real networks with our P2P streaming system. In particular, we will evaluate the predication model through collected peer population traces to find out how to derive the optimal operating point for the system.

References

[1] Peer-to-peer streaming simulator, available online at: http://media.cs.tsinghua.edu.cn/~zhangm/.
[2] M. Adler, R. Kumar, K. W. Ross, D. Rubenstein, T. Suel, and D. D. Yao. Optimal peer selection for p2p downloading and streaming. In *Proceedings of IEEE INFOCOM*, Miami, FL, March 2005.
[3] G. Box, G. M. Jenkins, and G. Reinsel. *Time Series Analysis: Time Series Analysis: Forecasting and Control (Third Edition)*. Prentice Hall, 1994.
[4] K. C. Ellen, W. Zegura, and S. Bhattacharjee. How to model an internetwork. In *Proceedings of IEEE INFOCOM 1996*, Philadelphia, PA, USA, June 1996.
[5] Gridmedia. http://www.gridmedia.com.cn.
[6] N. Magharei and R. Rejaie. PRIME: Peer-to-peer receiver-driven mesh-based streaming. In *Proceedings of IEEE INFOCOM*, Anchorage, Alaska, May 2007.
[7] M. Zhang, Y. Xiong, Q. Zhang, and S. Yang. On the optimal scheduling for media streaming in data-driven overlay networks. In *IEEE GLOBECOM*, November 2006.