Randomized load balancing strategies with churn resilience in peer-to-peer networks

Song Fu a,*, Cheng-Zhong Xub, Haiying Shen c

a Department of Computer Science and Engineering, University of North Texas, Denton, TX 76207, USA
b Department of Electrical and Computer Engineering, Wayne State University, Detroit, MI, USA
c Department of Electrical and Computer Engineering, Clemson University, Clemson, SC, USA

1. Introduction

Peer-to-peer (P2P) networks have become, in a short period of time, one of the fastest growing and most popular Internet applications. An important class of P2P overlay networks is the distributed hash tables (DHTs) that map keys to peer nodes based on a consistent hashing function. Representatives of DHTs include Chord (Stoica et al., 2003), Pastry (Rowstron and Druschel, 2001), Tapestry (Zhao et al., 2004), CAN (Ratnasamy et al., 2001), and Cycloid (Shen et al., 2006). In a DHT, each node and key has a unique ID, and a key is mapped to a node according to the DHT definition. The ID space of a DHT is partitioned among nodes and each of them is responsible for the keys whose IDs are located within its ID space. An important goal in designing DHTs is to achieve a balanced partition of the hash space among peer nodes. It is often desirable that each node takes responsibility for a portion of the hash space which is proportional to its processing capacity, measured in terms of its processor speed, available bandwidth, and/or storage capacity. This property should be sustained as nodes join and leave the system. A similar goal is desirable in unstructured P2P networks as well.

However, consistent hashing (Stoica et al., 2003) produces a bound of $O(\log n)$ imbalance of key distribution among peer nodes, where $n$ is the number of nodes in a P2P network. Things are even worse in unstructured P2P systems, where no commonly accepted load distribution mechanism is supported. In addition, users may query geographically adjacent nodes and those that have popular files. These lead to even more imbalanced distribution of workload in the network. When a node becomes overloaded, it cannot store any additional files or respond to user queries any more, which affects the system utilization and user satisfaction. To balance load among peer nodes in a network, lightly loaded peers should be selected to store files or serve queries. Load balancing in P2P networks is an important topic and many related works have been conducted recently (Tewari and Kleinrock, 2006; Shen and Xu, 2006; Zheng et al., 2006; Godfrey and Stoica, 2005; Bienkowski et al., 2005).

It is known that simple randomized load balancing schemes can balance load effectively while incurring only a small overhead in general parallel and distributed computing environments (Xu and Lau, 1997), which makes such schemes appealing for practical systems. The approach of multiple random choices in P2P networks was used in Shen and Xu (2006), Godfrey and Stoica (2005), Kenthapadi and Manku (2005), Zhu and Hu (2005), and Byers et al. (2003). Several peer nodes are probed and the one with the least load is selected to store a file on it or serve a user query. Random choice algorithms are scalable and they require a small number of control messages and data structures (Shen and Xu, 2006; Byers et al., 2003). More importantly, they work well in P2P systems with churn, a situation where a large percentage of