Topology Adaptation in P2P Networks Using Schelling's Model

Atul Singh and Dr. Mads Haahr

Distributed Systems Group, Department of Computer Science. Trinity College, Dublin
Atul.Singh@cs.tcd.ie, Mads.Haahr@cs.tcd.ie,
WWW home page: http://www.dsg.cs.tcd.ie

Abstract. The paper presents a study on utilising Thomas Schelling's model, to perform topology adaptation in unstructured decentralised P2P networks. Schelling's model suggests an explanation for the existence of segregated neighbourhoods in America. The paper uses an abstract version of Schelling's algorithm. A simulator to study different variations of Schelling's model has been implemented. The paper presents a case study demonstrating how the abstract algorithm can be used to improve bandwidth usage in P2P networks.

1 Introduction

P2P applications allow symmetric communication between peer computers running the software. P2P applications allow us to utilise the resources (such as processing power, disk space) of powerful desktop computers which in a client-server model are restricted to be mere consumers of services, running light-weight applications like web browsers and email clients. P2P applications allow all the computers running the P2P software to become both consumers and suppliers of information and services.

P2P applications organise the peer computers in a virtual communication network, also called the overlay network. The overlay network generally has self-organising characteristics. It is established and maintained by the P2P software without any human intervention. P2P software manages events like peers joining and leaving the network. The overlay network uses the communication infrastructure, such as TCP/IP or HTTP, provided by the transport layer. The overlay network allows peers to discover other peers and communicate with them.

On the basis of the overlay network architecture P2P applications can be divided into three major categories: centralised, decentralised structured and decentralised unstructured. Centralised architectures uses a well known central server for performing organisational tasks to facilitate P2P communication. Napster [1] is a prominent example of a centralised architecture. In decentralised architectures, peers collaborate without a central server. In structured decentralised networks, the location of resources and peers in the network is controlled, typically using a distributed hash table (DHT) of resources available on the network. Chord [13] and Pastry [2] are prominent examples of this architecture. Unstructured P2P networks do not control the location of resources and peers. Gnutella [4] and Kazaa [14] are popular P2P applications which use an unstructured P2P network. The topology adaptation technique suggested in the paper is useful for decentralised unstructured networks only. The location of a peer in a decentralised structured network is determined by the key space it it responsible for which makes topology adaptation difficult.

In existing P2P overlay networks, there is typically no control over the type of peers which are connected together. This leads to suboptimal grouping of peers. For example, in a file-sharing application, peers with high bandwidth capacity may be grouped together with peers with low bandwidth capacity, which may lead to a degradation in performance. Also peers which are geographically distant (in terms of the underlying network infrastructure) may be grouped together, increasing the cost of communication between them. In file sharing applications, it is beneficial if peers with similar properties like bandwidth or geographic location are adjacent to each other. Other categories of P2P applications could group peers using different characteristics. The purpose of the present paper is to investigate an algorithm which may be used to modify or adapt network topologies so that peers with similar properties are close to each other.

In 1960 American economist Thomas Schelling proposed a model to explain the segregated neighbourhoods in America [3]. He observed that the segregated American neighbourhood are not caused by a central authority, or desire of people to stay away from dissimilar people; but is a cumulative effect of simple actions of individuals. Schelling's model is self-maintaining and decentralised in nature. This makes it suitable for topology adaptation in unstructured decentralised P2P networks, which lack central authority and are self-organising.

The paper uses Schelling's algorithm [1] in a P2P network to change the overlay network's topology. The topology of the overlay network is the graph whose vertices are the peers in the network and edges are all the connections between the peers. The paper uses an abstract form of algorithm used in Schelling's model. The abstract algorithm uses the *satisfaction state* of a peer to decide whether it should execute its *topology adaptation steps*. *Satisfaction state* criteria and *topology adaptation steps* will vary across different versions of Schelling's algorithm. They are discussed in detail along with the abstract algorithm in Section 4.

The rest of the paper is organised as follows. Section 2 describes Schilling's work in detail. Section 3 discusses the motivation behind the work in this paper. Section 4 presents an abstract form of Schelling's algorithm which can be executed by peers to change the topology to satisfy particular constraints. Section 5 discusses the design of the simulator used for the experiments. Section 6 presents a case study containing a sample *satisfaction state* criteria and *topology adaptation steps*, and a concrete realisation of the abstract algorithm; which can be used to improve the performance of a P2P network by bringing together peers with similar bandwidth.

2 Thomas Schelling's Model

In Schelling's model [10] [11] the world is an $m \times n$ grid. A random number of cells in the grid are populated by blue or red turtles [2]. A cell can host only one turtle. In the beginning, a random number of blue and red turtles are randomly distributed on the grid. About one third of the cells in the grid are left empty. All the turtles desire a certain percentage of their neighbours to be of the same colour. If a turtle is not satisfied with its neighbours, it moves to an adjacent empty cell, chosen randomly. The simulation goes on till all the turtles are satisfied with their neighbours.

The model can be simulated using Netlogo [5], a modelling environment used to simulate complex multi-agent systems. Netlogo comes with a sample [6] which can be used to simulate Schelling's model. The sample model has two parameters: the total number of turtles (N) and the percentage of similar neighbours (PSND) that each turtle desires. In the NetLogo

[1] The algorithm executed by individuals in Schelling's model is referred as Schelling's algorithm in the paper.

[2] Schelling studied his model using nickels and pennies on a chess board.

model, segregation can be observed visually in a graphical display of the turtle world. The model tracks percentage of turtles having all similar neighbours (PTASN) over time. The metric gives a numerical idea of segregation. A small value of PSND, leads to the emergent behaviour of a very high percentage of turtles having all similar neighbours (PTASN). The parameter PSND, has a critical value $PSND_{unstable}$, beyond which the simulation do not converge to a state where all the turtles are happy with their neighbours. Segregation does not depend upon the number of turtles. The sample NetLogo model was modified so that the world is populated with more than two type of turtles. An emergent behaviour similar to the old model was observed.

3 Motivation

Schelling's model is an example of a complex system, in which the desire of the individual turtles to stay in a neighbourhood with a very small number of similar turtles leads to an emergent behaviour of highly segregated neighbourhoods. The turtles act using their awareness of the local network topology, which makes this model especially attractive for P2P systems in which the peers lack a global picture of the network topology. Schelling's model is perhaps the most famous model of self-organising behaviour [7]. Clustering occurs in the model without any central direction or control. The self-organising and decentralised clustering in Schelling's model makes the model a suitable candidate solution for clustering peers in P2P networks.

The network of individuals in Schelling's model is different from the network in P2P. Schelling's world is an $m \times n$ grid. A turtle can only be placed in an empty cell in the grid. In a P2P network there is no constraint on the location of a peer. In Shelling's model the neighbours of a turtle can only be turtles located in the eight cells adjacent to its own. In a P2P network neighbours can be located anywhere. To the author's knowledge, the effect of applying Schelling's algorithm to a P2P overlay network has not yet been studied. This paper studies the effect of applying Schelling's algorithm to peers in a P2P network.

P2P networks are graphs and can be studied easily in a simulation environment that supports graph-related functions. NetLogo lacks support for graph-related functions, which makes modifying the NetLogo model to simulate P2P networks a tough task. Instead a simulator has been developed using C++, which can be used to study the impact of applying Schelling's algorithm to P2P networks.

In Schelling's algorithm a turtle is satisfied if a percentage of its neighbours has the same colour as the turtle. If the turtle is not satisfied with its neighbourhood, then it moves to a randomly chosen empty adjacent cell. Numerous variations of Schelling's algorithm can be created by choosing different satisfaction criteria, and steps to be performed if a turtle is not satisfied. The next section 4 presents some examples of satisfaction criteria and steps to be performed if a turtle is not satisfied.

4 Algorithm for Topology Adaptation

This section presents an abstract form of Schelling's algorithm that can be executed by peers in a P2P network in order to perform topology adaptation. The *Template method* design pattern [8], is used to separate the steps of Schelling's algorithm. These steps may be changed to produce different versions of Schelling's algorithm.

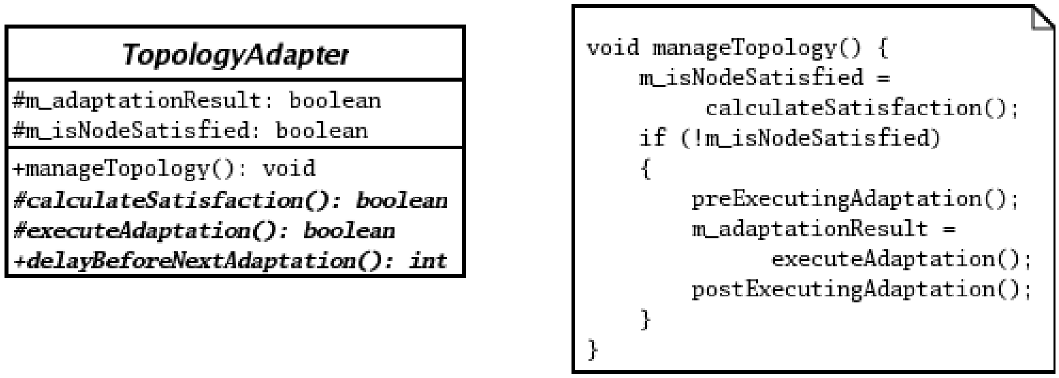

Fig. 1. UML Class diagram of TopologyAdapter Interface

In the *Template method* design pattern the skeleton of an algorithm is defined in an operation, deferring some steps to subclasses. The subclasses implement the steps that vary. This makes it possible to redefine the steps in an algorithm without changing the algorithm's structure. The steps which may vary across various versions of Schelling's algorithm are the satisfaction criteria, the actions to be performed if a peer is not satisfied and the frequency with which satisfaction state should be checked.

Figure 1 presents the class diagram for the TopologyAdapter interface. The interface encapsulates the topology adaptation algorithm. The simulator expects the peers to implement the TopologyAdapter interface. The manageTopology method contains the skeleton of the algorithm which will be executed by individual peers. The pseudo-code for the method is presented as a note in the figure. A peer calculates its satisfaction state and, if not satisfied, executes topology adaptation steps.

Satisfaction state is a boolean value indicating whether a peer is satisfied with the overlay network's topology. The satisfaction state of a peer is calculated using the calculateSatisfaction method. Satisfaction criteria will vary depending upon the topology desired. For example, a file-sharing P2P application might like to have a topology so that peers with high number of files are mixed together with peers having a small number of files. In such a scenario, the satisfaction criteria could be desire for a certain percentage of the neighbours to be dissimilar.

If a peer is not satisfied with its neighbours then topology adaptation steps are performed by calling the executeAdaptation method. As a part of *Topology Adaptation Steps*, a peer may decide to add a similar (or dissimilar) peer as its neighbour. Satisfaction criteria and topology adaptation steps may vary across peers in an overlay network. For example, in an overlay network some peers may be satisfied if they are with all dissimilar neighbours and the rest may be satisfied if they are with all similar neighbours.

The topology management algorithm, specified in manageTopology method, is executed successively. The time delay between successive executions of topology management algorithm is determined by the return value of the method delayBeforeNextAdaptation. The time delay will vary with application. For example, the algorithm may be executed repeatedly after a constant delay of ten minutes.

All the three methods: calculateSatisfaction, executeAdaptation and delayBeforeNextAdaptation are virtual abstract methods implemented in subclasses.

5 Simulator

An overlay network simulator has been developed, in which all the peers are within one simulator process. The simulator is single threaded, which means that the peers execute their algorithm sequentially. Peers execute an implementation of Schelling's algorithm. Each peer is assigned a numeric identifier. Based on the result from delayBeforeNextAdaptation() the peers calculate the time to execute the TopologyAdaptation algorithm. In each iteration the simulator goes through the peers in an increasing order of identifier's. An iteration is counted as one time unit. The peers executes the topology adaptation algorithm when the simulator time matches the time to execute topology adaptation.

The simulator can be used for evaluating different versions of Schelling's algorithm. It is used for evaluating the algorithms used in the case studies presented in this paper.

6 Case Study

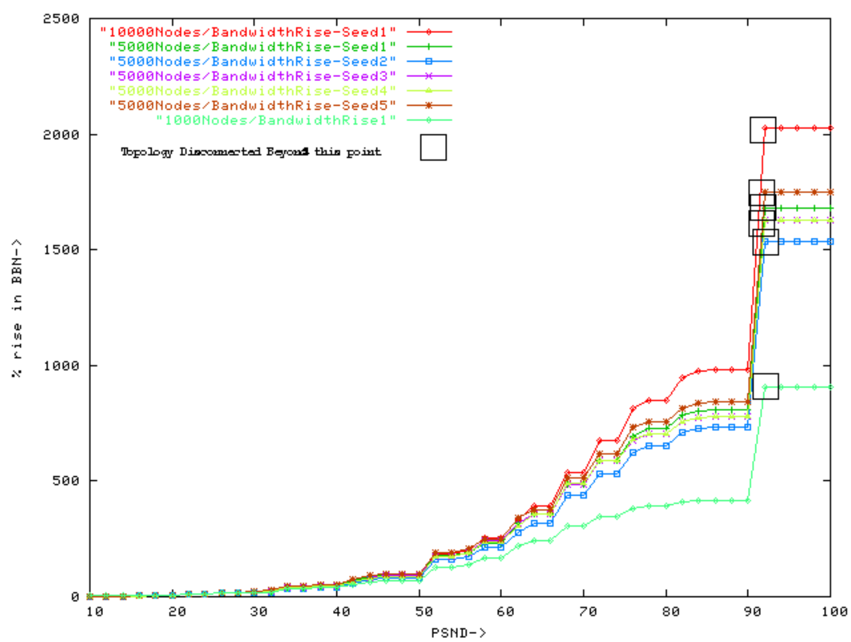

Fig. 2. Plot of percentage increase in BBN against PSND, after dropping dissimilar neighbours and adding similar neighbours, for random networks with 1000, 5000 and 10000 nodes.

This section presents a case study that demonstrates how Schelling's algorithm can be utilised in a P2P network. The case study shows how Schelling's algorithm can be used to increase the bottleneck bandwidth in a P2P network, so that information is exchanged between peers at a faster rate.

In an overlay network a message is routed through a number of peers before it reaches its destination. The bottleneck bandwidth between the source and the destination is the minimum of the bandwidth of the hops in the route. The bottleneck bandwidth gives an accurate upper bound of the rate at which information will be exchanged between peers [9]. In an unstructured decentralised network the location of the peers is decided randomly, and therefore peers with high bandwidth may be adjacent to peers with low bandwidth, introducing undesired low bottleneck bandwidths in the network. The *bottleneck bandwidth for the network* (BBN) is the average of the bottleneck bandwidth between all possible pairs of peers in the network. This case study demonstrates that BBN can be increased by applying Schelling's algorithm so that peers with similar bandwidth are clustered together.

For the case study two set of simulations using different *topology adaptation steps* have been performed using random networks of peers with 1000, 5000 and 10000 nodes. In each random network, half the peers (chosen randomly) have an available bandwidth of 10Mbps, and the other half a bandwidth of 1Mbps. The connections between the peers is chosen at random. A peer can have a maximum of ten connections. Self loops, parallel loops and duplicate connections are not allowed. It is ensured that the generated random topology is connected. The generated topology is similar to the topology of unstructured P2P networks. In the simulations, PSND is varied from 10 to 100 in steps of 2. The algorithm is applied on each random network, with different values of PSND and the rise in BBN is calculated. The Topology management algorithm is executed successively at a constant interval. Simulations stop if all the peers are satisfied or if the simulator has gone through five hundred iterations.

In both sets of simulations the *satisfaction state criteria* used is that a peer is satisfied if a percentage of its existing neighbours have similar bandwidth. Two different *topology adaptation steps* are used. They are:

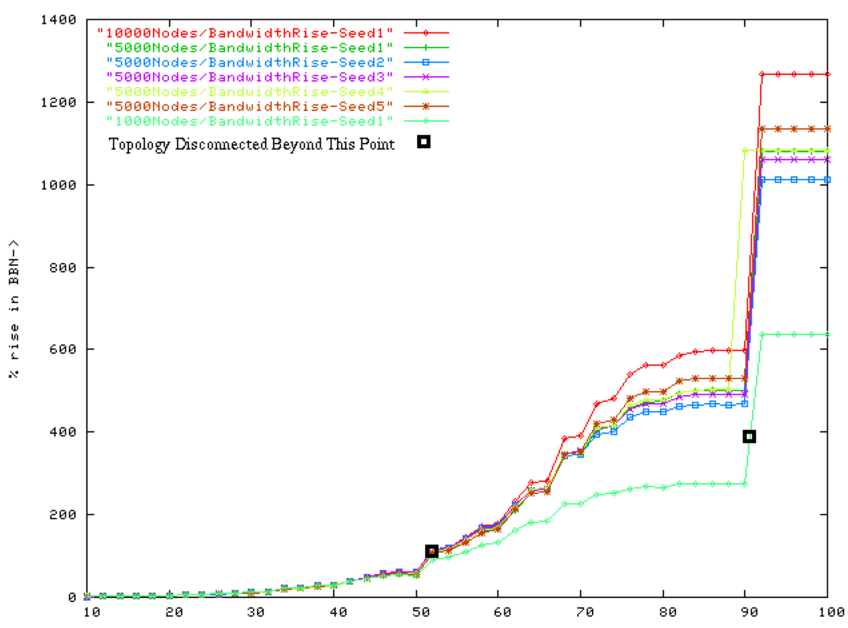

Fig. 3. Plot of percentage increase in BBN against PSND after dropping dissimilar neighbours selfishly for random networks with 1000, 5000 and 10000 nodes.

6.1 Drop dissimilar and add similar neighbour (DDAS)

For the first set of simulations, a dissatisfied peer executes its *topology adaptation steps*, in which it:

1. Drops a dissimilar neighbour if it is not the only neighbour of that peer (to ensure that the topology remains connected).
2. Searches for a similar neighbour which has a free connection slot. A Depth First Search (DFS) is performed. The horizon of the search is five. If a suitable peer is found then it is added as neighbour.

Figure 2 plots the percentage increase in BBN against PSND, for random networks with 1000, 5000 and 10000 nodes. BBN increases with an increase in desire to have similar neighbours. For the same PSND, a higher rise in BBN is observed if the network has more nodes. In the simulations, all nodes are satisfied in less that ten iterations. The network topology was disconnected beyond a certain PSND called $PSND_{disconnected}$. The value of $PSND_{disconnected}$ is 92 for all the random networks.

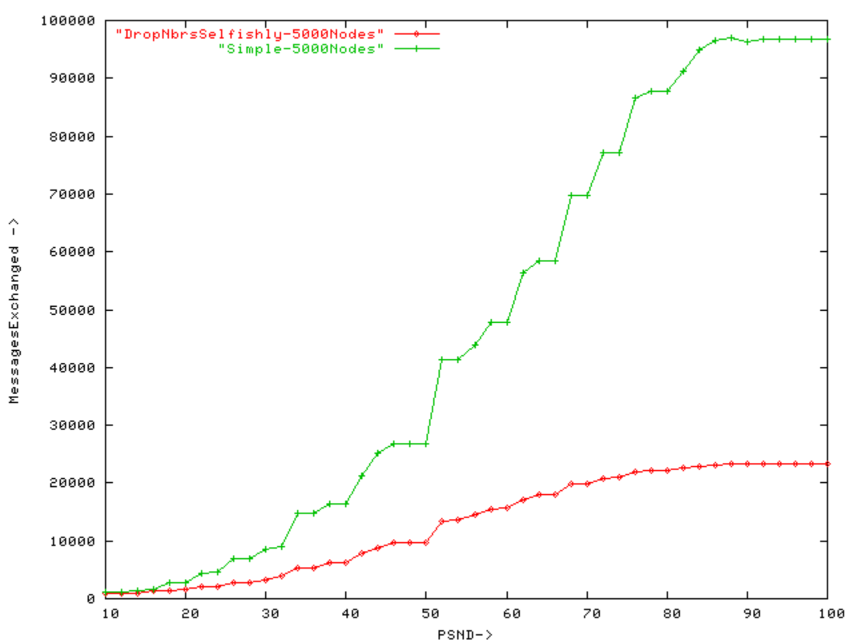

Fig. 4. Number of messages exchanged to perform topology adaptation against PSND, for a 5000 Node random network. Plot DropNbrsSelfishly-5000Nodes is for dropping dissimilar neighbours selfishly. Plot Simple-5000Nodes is for dropping dissimilar neighbours and adding similar neighbours.

6.2 Drop dissimilar neighbours selfishly (DD)

For the second set of simulations, a dissatisfied peer executes its *topology adaptation steps* in which it selfishly drops a dissimilar neighbour, in each iteration.

Figure 3 plots the percentage increase in BBN against PSND, for random networks with 1000, 5000 and 10000 nodes. Like the previous set of simulations, BBN increases with an increase in desire to have similar neighbours. For the same PSND, a higher rise in BBN is observed if the network has more nodes. In the simulations, all nodes are satisfied in less that twenty iterations. The network topology was disconnected beyond a certain PSND called $PSND_{disconnected}$. The value of $PSND_{disconnected}$ was 52 for random networks with 5000 and 10000 nodes. For random networks with 1000 nodes $PSND_{disconnected}$ varied from 58 to 92.

6.3 Cost of Executing Topology Adaptation

Figure 4 plots the number of messages exchanged for topology adaptation against PSND, using DDAS(simple-5000Nodes) and DD(DropNbrsSelfishly-5000Nodes), for a random network with 5000 nodes. With DDAS, for a PSND of 22, the number of messages exchanged are of the order of 4000. The number of messages exchanged are of the order of 25000, for a PSND of 50; and 95000 for a PSND of 92. In DD nodes without any neighbour reconnect to the network. The number of messages exchanged for DD include the messages exchanged by these isolated nodes to reconnect to the network.

There is a trade off between rise-in BBN and the number of messages exchanged. The optimal choice of PSND and *topology adaptation steps* depends upon the application and its characteristics. For an overlay network with a long life it may be advisable to gain a huge rise in BBN by exchanging lots of messages. Whereas an overlay network with a short life span may choose PSND and *topology adaptation steps* which do not provide a substantial rise in BBN, but are less demanding in number of messages exchanged. For example an overlay network with a long life may choose DDAS, with a PSND of 80 which will give it more than 600 percent rise in BBN, if there are around 5000 nodes in the network. But with this PSND around 80000 messages will be exchanged, which is a very large number. An overlay network with a short life may choose DD with a PSND of 50 which is just below $PSND_{disconnected}$. This will provide 50 percent rise in an overlay network with 5000 nodes, by exchanging 10000 messages.

Rise in BBN is high in an overlay network with higher number of nodes for both DDAS and DD. But below 50 percent PSND there is no dramatic difference in the rise in Bandwidth for overlay network with different number of nodes.

7 Conclusion

The paper has demonstrated that Schelling's algorithm can be used effectively for adapting P2P network topology. The abstract algorithm and the simulator presented can be used to develop and evaluate different variations of Schelling's algorithm. The paper presented a case study where all the peers in a P2P network desire a percent of their neighbours to be similar (PSND) to them. The case study demonstrates that the BBN of a P2P network can be increased by using Schelling's algorithm.

References

1. Napster, http://www.napster.com, 1999 version.

2. A. Rowstron and P. Druschel "Scalable Pastry Scalable, decentralized object location and routing for large-scale peer-to-peer systems." Proceddings of the 18th IFIP/ACM International Conference on Distributed Systems Platforms(Middleware 2001).
3. Jonathan Rauch "Seeing Around Corners" The Atlantic Monthly, April 2002.
4. The Gnutella Protocol Specifications v0.4 http://www.clip2.com
5. NetLogo, http://ccl.northwestern.edu/netlogo/.
6. Wilensky, U. NetLogo Segregation model. http://ccl.northwestern.edu/netlogo/models/Segregation Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL.
7. Mark Pollicott and Howard Weiss "The Dynamics of Schelling-Type Segregation Models and a nonlinear graph laplacian variational problem" http://citeseer.ist.psu.edu/570913.html.
8. Gamma, E., Helm, R., Johnson, R., Vlissides, J., Design Patterns: Elements of Reusable Object-Oriented Software, Reading, Mass.: Addison-Wesley, 1995.
9. Saroiu S, Gummadi K. and Gribble S., "Measuring and analyzing the characteristics of Napster and Gnutella hosts" Multimedia Systems Journal, Volume 8, Issue 5, November 2002.
10. Schelling C. T. "Micromotives and Macrobehaviour", Nortan and Company 137-166: W. W. Norton, 1978.
11. Schelling "Dynamic Models of Segregation", Journal of Mathematical Sociology 1:143-186..
12. Barbasi, Albert and Jeong, "Mean-field theory for scale-free random networks", Physica 1999.
13. Ion Stoica, Robert Morris, David Karger, M. Frans Kaashoek, and Hari Balakrishnan, Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications, ACM SIGCOMM 2001, San Deigo, CA, August 2001, pp. 149-160.
14. Kazaa, http://www.kazaa.com/us/index.htm.