This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the ICC 2007 proceedings.

Decentralized Clustering In Pure P2P Overlay Networks Using Schelling's Model

Atul Singh and Mads Haahr
Distributed Systems Group,
School of Computer Science and Statistics,
Trinity College Dublin,
Ireland.
Email: Atul.Singh@cs.tcd.ie, Mads.Haahr@cs.tcd.ie

Abstract— **Clustering involves arranging a P2P overlay network's topology so that peers having certain characteristics are grouped together as neighbors. Clustering can be used to organize a P2P overlay network so that requests are routed more efficiently. The peers lack of a global awareness of the overlay network's topology in a P2P network makes it difficult to develop algorithms for clustering peers. This paper presents two decentralized algorithms for clustering peers. The algorithms are concrete realizations of of an algorithm called the abstract Schelling's algorithm (based on a model from sociology by Thomas Schelling) that can be used to create a family of self-* topology adaptation algorithms for P2P overlay networks. The proposed clustering algorithms are easy to implement, are not designed for clustering on a specific criteria and do not require separate algorithms to handle the flux of peers on the overlay network. The paper presents simulation results for applying the algorithm on random small-world topologies.**

I. Introduction

The term Peer-to-Peer (P2P) is used to refer to distributed systems without any central control, where all the nodes (called peers) are equivalent in functionality. In a P2P system, peers can collaborate and communicate with each other without utilizing expensive and difficult to maintain central infrastructure. P2P systems organize the peer computers in a virtual communication network called an overlay network. Overlay networks generally have self-organizing characteristics, which means that they are established and maintained by P2P software without any human intervention and that P2P software manages events such as peers joining and leaving the network. The self-organizing and decentralized nature of P2P helps reduce the management cost of the computer infrastructure.

The topology of the overlay network is a graph whose vertices are the peers in the network and whose edges are the connections between the peers. On the basis of the overlay network architecture, P2P applications can be divided into three major categories: centralized, decentralized structured and decentralized unstructured (the latter is also called pure P2P in this paper) [1]. P2P applications arrange the topology of the overlay network to satisfy certain criteria. For example, file-sharing applications like KaZaA [2] arrange the topology so that ordinary peers are connected to powerful peers (called super-peers) and the super peers are connected to each other to form a backbone network. In this paper, the term *topology adaptation* refers to adjusting the topology of an overlay network to satisfy certain criteria when peers join or leave the network. The topology adaptation algorithm can be used to create and maintain a desired topology. The decentralized nature of P2P networks makes it difficult to develop distributed algorithms for topology adaptation.

Clustering involves arranging the topology so that peers having certain characteristics are grouped together as neighbors. Clustering can be used to organize the overlay network so that requests are routed more efficiently. In existing literature the term clustering is often used to refer to identification of clusters in a network. However, in this work the term clustering refers to the creation of clusters. Cluster identification can be done by using algorithms proposed in [3], [4], [5], [6].

The type and effect of clustering depends on application concerns but the technique is useful in a variety of settings. For example, the performance of messaging on an overlay network can be improved by clustering peers that are close to each other on the underlying physical network [7], [6]. In a file-sharing application, peers sharing similar files may be clustered together, which can be used to improve search performance because search requests can be routed to an appropriate cluster and then a deep search can be performed within the cluster [7], [8], [9]. The performance of messaging on a P2P overlay network can be improved by clustering peers in the same geographical location [10]. Clusters with low intra-cluster network latencies can be used to provide coarse-grained parallelism in which parts of a parallel application are distributed across hosts in the cluster [4]. In a distributed game, clustering can be used to bring together players with similar levels of competency so that their gaming experience can be improved.

In 1969, Thomas Schelling, an economist, proposed a model [11], [12], [13], [14], [15] to explain the existence of segregated neighborhoods in America. He observed that the appearance of such segregated neighborhoods is caused neither by a central authority, nor by the desire of people to stay away from dissimilar people; instead, it is the cumulative effect of simple actions (moves) by individuals who want at least a certain proportion of their neighbors to be similar to themselves. Schelling's model is decentralized and self-maintaining in nature. This makes it attractive for topology adaptation in

1-4244-0353-7/07/$25.00 ©2007 IEEE

Operation	**Details**
count(*property*)	The number of neighbors of a given node matching the given property.
add(*peers*)	Add the given peer or peers as a neighbor
drop(*peer*)	Drop the given peer as a neighbor
neighbor(*property*)	Returns a neighbor with the given property.
search(*property*)	Search for peers on the overlay network with the given property.

TABLE I

EXAMPLE OF OPERATIONS THAT CAN BE EXECUTED BY THE PEERS. THE OPERATIONS ARE USED IN THIS PAPER TO DESCRIBE THE CLUSTERING ALGORITHMS.

dynamic environments such as pure P2P networks, which lack a central authority.

In our earlier work [16], we have presented an abstract algorithm (called the abstract Schelling's algorithm) based on Schellings model, which can be used to create a family of topology adaptation algorithms for P2P networks. The topology adaptation algorithms can be executed by the peers to create a P2P topology that satisfies certain criteria. The topologies developed using this approach can adapt to the continuous arrival and departure of peers in the network. The algorithms developed using the abstract Schelling's algorithm are only useful for pure P2P networks. In a decentralized structured network, the location of a peer in the overlay network is determined by the key space for which it is responsible, thus making topology adaptation difficult.

This paper describes two algorithms based on the abstract Schelling's algorithm, called SelflessClustering and SelfishClustering algorithms, which can cluster peers in a pure P2P network. The paper presents the simulation results of applying the clustering algorithms on P2P networks. The simulations are used to study the dynamic properties of the overlay networks when the clustering algorithms are applied to it. The simulations show that the clustering algorithms reduce the number of groups of similar peers (clusters) on the overlay network by bringing similar peers together as neighbors.

The rest of the paper is organized as follows: Section II describes Schelling's original work. Section III presents the abstract Schelling's algorithm that can be used to generate a family of topology adaptation algorithms. Section IV defines the problem of peer clustering and presents two concrete realizations of the abstract algorithm, that can be used to cluster peers. Section V presents the simulations and the results of applying the clustering algorithms on an overlay network. Section VI reviews a number of existing decentralized algorithms for clustering peers. Section VII presents the conclusion and future work.

II. SCHELLING'S MODEL

An agent-based model is a tool that can be used to study the emergence of complex behavior from simple rules in decentralized systems. An agent-based model consists of large number of agents that change their properties and their environment by using their knowledge of the local neighborhood. In the 1969, economist Thomas Schelling [11], [12], [13] proposed an agent-based model that can be used to explain the existence of segregated neighborhoods in urban areas.

In Schelling's model, the world is modeled as a 2-dimensional grid. Approximately two thirds of the cells in the grid are populated by blue or red turtles. The remaining cells are empty. Each cell can host a maximum of one turtle. In the beginning, a random number of blue and red turtles are randomly distributed on the grid. All the turtles desire at least a certain percentage of their neighbors to be of the same color as themselves. If a turtle is not satisfied with its neighbors, it moves to an adjacent empty cell (if available) chosen randomly. The simulation goes on until all the turtles are satisfied with their neighbors. As the simulation progresses, segregation can be observed on the grid. Such segregation is an *emergent behavior* caused by the desire of the turtles to assure that a certain minimum percentage of their neighbors are the same color as themselves. Schelling's model is different from cellular automata in which cells change their state based on the state of their neighboring cells.

Variations of Schelling's model have been mathematically analyzed by Zhang in [17], [18] and Young in [19], as a game played between people. In the game, each player has a strategy and a payoff that is determined by the status of the player's neighborhood. By using theories from stochastic dynamical systems, it has been shown that the stable state for the system is a segregated state.

The clustering algorithms presented in this article are inspired by Schelling's model. In Schelling's model, the turtles act using their awareness of the local network topology, which makes it especially attractive for P2P systems in which the peers lack a global picture of the network topology. In the model, grouping is maintained even when turtles join or leave the system (self-organizing), which makes the model ideal for the dynamic environments of P2P networks.

III. ABSTRACT SCHELLING'S ALGORITHM

The abstract Schelling's algorithm [16] is motivated by the Schelling's model and can be used to create a family of topology adaptation algorithms. In Schelling's algorithm, the steps that may vary are the satisfaction criteria, the actions to be performed if a peer is not satisfied and the frequency with which the satisfaction state should be checked. In the

Algorithm 1 SelflessClustering Algorithm

```
$PNSP_{desired}$ ← % of neighbors with similar property desired
while true do
  $PNSP_{actual} \leftarrow \frac{\textbf{count}(\textit{same property})*100}{\textbf{count}(\textit{all})}$
  if $PNSP_{actual} < PNSP_{desired}$ then
    if count(all) $> 1$ then
      drop(neighbor(different property and count(all) $>$ 1))
    end if
    add(search(same property))
  end if
  sleep(delay)
end while
```

Algorithm 2 SelfishClustering Algorithm

```
$PNSP_{desired}$ ← % of neighbors with similar property desired
while true do
  $PNSP_{actual} \leftarrow \frac{\textbf{count}(\textit{same property})*100}{\textbf{count}(\textit{all})}$
  if $PNSP_{actual} < PNSP_{desired}$ then
    drop(neighbor(different property))
  end if
  sleep(delay)
end while
```

abstract Schelling's algorithm a peer periodically calculates its *satisfaction state* (SC) at pre-defined intervals and if it is not satisfied then it executes its *topology adaptation steps* (TAS).

Satisfaction state is a boolean value indicating whether a peer is satisfied with its local view of the overlay network's topology. If a peer is not satisfied with its neighbors then topology adaptation steps are performed. The satisfaction criteria, the topology adaptation steps and the time delay between successive calculation of the satisfaction state will vary with the application and the topology desired.

Designing satisfaction state and topology adaptation steps that can be executed autonomously by the peers without requiring a global knowledge of the overlay network is a challenge. In our previous work [16] we have presented a concrete realization of the abstract Schelling's algorithm that can be used to create a network of hubs in a pure P2P network. The next section presents two concrete realizations of the abstract Schelling's algorithm that can be used for clustering peers in a pure P2P network.

IV. Clustering Algorithms

A. Clustering

The P2P overlay network can be considered as a graph $G = (V, E)$, where V is the set of peers on the overlay network and E is the set of connections between the peers. Let $P = \{p_1, p_2 ....p_n\}$ be the set that enumerates the types of peers on the graph G. Let $SG_{p_i} = (V(p_i), E(p_i))$ be a subgraph of graph G that contains all the peers from the original graph with property p_i and the connections between them, so that:

$$V(p_i) = \{v \epsilon V : v.property = p_i\}$$

$$E(p_i) = \{\{v, w\} \epsilon E : v, w \epsilon V(p_i)\}$$

A connected component of a graph is a sub-graph in which all the peers are connected to each other. We use the term **cluster** to refer to a connected component of the graph SG_{p_i}. Let $CC_{p_i} = \{CC_1, CC_2, ...CC_n\}$ be the set of connected components of the graph SG_{p_i}. The number of clusters NC in the graph G is :

$$NC = \sum_{i=1}^{n} |CC_{p_i}|$$

Clustering is used to rearrange the overlay network so that peers with similar characteristics are brought together as neighbors. Clustering changes the overlay network topology so that $|CC_{p_i}|$ is minimized.

B. Algorithms

This section presents two clustering algorithms, called the SelflessClustering algorithm and the SelfishClustering algorithm, which can be executed by the peers on an overlay network in order to achieve clustering. Algorithms 1 and 2 present the pseudo-code for these algorithms. Table I describes the operations used in the pseudo-code.

The clustering algorithms are self-organizing, meaning that they organize the peers using their local awareness of the overlay network topology and without any central authority. The clustering algorithms are self-maintaining in nature which means that they maintain the reorganized topology by calculating a peer's satisfaction state at regular intervals and taking topology adaptation steps if the peer is not satisfied with its neighborhood. The algorithms are described in detail below:

- **SelflessClustering Algorithm:** The satisfaction criteria states that a peer is satisfied if more than a certain percentage of its existing neighbors are similar to it. This percentage is called the desired percentage of neighbors with similar property ($PNSP_{desired}$). In this algorithm a dissatisfied peer performs the steps below:
  - **step 1** - Drop a dissimilar neighbor if it is not the only neighbor and it is connected to more than one peer (to ensure that the topology remains connected).
  - **step 2** - Search for similar peers that have a free connection slot. If suitable peers are found then add them as neighbors.
- **SelfishClustering Algorithm:** This algorithm uses the same satisfaction criteria as the SelflessClustering algorithm. However, in the SelfishClustering algorithm, a dissatisfied peer drops one randomly chosen dissimilar neighbor regardless of whether the peer being dropped is its only neighbor or it is the only neighbor of the peer being dropped. The selfish dropping of a dissimilar

neighbors can result in peers that are disconnected from the overlay network.

The topology adaptation steps taken by the peers create a feedback effect that results in an overlay network with a $PNSP$ value that is higher than $PNSP_{desired}$. When an unsatisfied peer adds another similar peer or drops a dissimilar peer as its neighbor, it increases its $PNSP$ value as well as the $PNSP$ value of the peer it is interacting with.

The number of messages exchanged to perform the search operation for similar peers, is a major cost of utilizing the SelflessClustering algorithm. The search operation can be implemented in a variety of ways. For example, the search can be performed by using a Gnutella like breadth first search (BFS) that floods the overlay network with a search request [20]. The number of messages exchanged to perform BFS is high but it performs a thorough search of the peer's neighborhood and is more likely to find a similar peer. If the high traffic overhead caused by BFS is an issue then the search can be performed by random walks [21], which is cheaper but will not search a peer's neighborhood as thoroughly and is less likely to find a similar peer. Another alternative is biased random walk as used in [22] that performs a more exhaustive search when compared to random walk. In [22] each peer maintains a directory of resources available on its neighbors and the random walk is biased towards peers with a high degree because they have more information about resources on the overlay network.

A fourth possibility is to implement the search operation as a gossip-based search. In a gossip-based search, a peer uses a list of peers, populated by using information from messages routed through it on the overlay network and periodic exchange of information about other peers on the overlay network with neighbors, to find a similar peer [23]. The performance of a gossip-based search is stochastic, and the chance of finding a suitable peer will depend upon the diversity (in terms of information about other peers on the overlay network) of the messages that are routed through the peer looking for a similar peer. Other search algorithms are of course also possible and the most suitable approach depends on application requirements.

The time delay between successive estimation of the satisfaction state is another important factor that affects the cost of utilizing the clustering algorithms. A satisfied peer need not estimate its satisfaction state at small continuous intervals. Similarly a peer that is unable to successfully execute its topology adaptation steps need not estimate its satisfaction state at small continuous intervals. They can instead use an exponentially increasing time delay between successive estimation of the satisfaction state.

V. Simulations

This section presents the simulation results of applying the clustering algorithms on P2P overlay networks with small-world characteristics [24]. In the simulations, the overlay networks have equal proportions (selected randomly) of five different types of peers. Each type of peer has a unique property that distinguishes it from other types of peers. In a real-life scenario this unique property could be the geographical location of the peer or the content type that is being shared. Two types of simulations have been done using the clustering algorithms: *static simulations* which use an overlay network that has no flux of peers and *dynamic simulations* in which peers are added at regular intervals in the system. The clustering algorithms are applied on the overlay networks and the results are studied using metrics described in the next section.

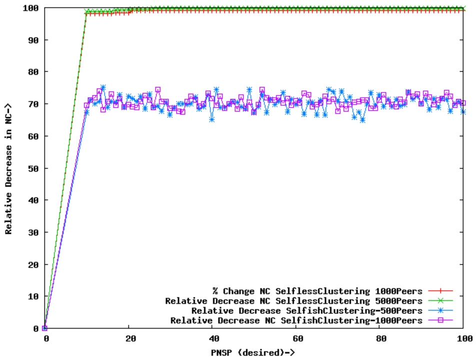

Fig. 1. A plot of % decrease in number of clusters after applying the Clustering and SelfishClustering algorithms against PNSP(desired), on overlay networks with 1,000 and 5,000 peers.

A. Metrics

1) Relative Decrease in number of clusters: Let NC_{orig} be the number of clusters in the original overlay network. Let NC_{curr} be the number of clusters in the overlay network after applying the clustering algorithms. The relative decrease in number of clusters ($RDNC$) is:

$$\frac{(NC_{orig} - NC_{curr}) \times 100.00}{NC_{orig}}$$

A high value of $RDNC$ indicates that a large number of groups of similar peers on the overlay network are merged to form small number of generally large sized groups. The desired value of $RDNC$ will vary with application.

2) Cluster Accuracy: ClusterAccuracy is a metric that measures the cliquishness of the cluster [25]. Each peer v, in the graph G is a member of a single cluster C. Let mp be the set of peers outside the cluster C to which the peer v is connected. Let ncc be the set of peers in the cluster C to which the v is not connected. The scaled coverage measure (SCM of the peer v, in the graph G is:

$$SCM(v) = \frac{|C| - |ncc|}{|C| + |mp|}$$

The Cluster Accuracy (CA) of the graph G is :

$$CA = \frac{\sum_{v \epsilon V} SCM(v)}{|V|}$$

A high value of CA is desirable for a clustering algorithm. CA can have a maximum value of 1 when there are no inter-cluster edges and all the peers in every cluster are connected to all the other peers in their cluster. The maximum value of CA is undesirable because it will lead to an unacceptable situation of a disconnected overlay network topology.

B. Static Simulations

Simulations have been done on four different networks of 500, 1,000 and 5,000 peers each using $PNSP_{desired}$ values from 10 to 100 in increments of 1. The simulations go on till all the peers are satisfied or 1,000 simulator iterations are reached. In the simulations all the peers are satisfied and the simulations converge within 10 simulator iterations at the maximum when SelflessClustering algorithm is used and within 60 simulator iterations at the maximum when SelfishClustering is used.

In the SelflessClustering algorithm the unsatisfied peers search for a similar peer on the overlay network. The search operation in the simulations has been implemented using Breadth First Search (BFS).

1) $RDNC$: Figure 1 plots the relative decrease in the number of clusters in the graph against the $PNSP_{desired}$, after applying the SelflessClustering and SelfishClustering algorithms. The number of clusters on the overlay network decreases because similar peers are connected together into groups. The peers are clustered into approximately five large clusters, one belonging to each type of peer, for the SelflessClustering algorithm. The decrease in the number of clusters after applying the SelfishClustering algorithm varies from 65 to 75 percent. For both the algorithms a low value like 10 can be used for the $PNSP_{desired}$ to achieve a significant decrease in the number of clusters.

2) Messaging Cost: In the SelflessClustering algorithm the messages exchanged to perform clustering will be proportional to the number of unsatisfied peers × $O(search)$. The search operation has been implemented using BFS in the simulations. In the worst-case scenario the $O(search)$ is $maxNbrs^h$ where $maxNbrs$ is the maximum number of neighbors that a peer can have and h the horizon of search. However in the simulations the average degree of the peers is much less than $maxNbrs$. A more realistic estimation of $O(search)$ is $avgNbrs^h$ where $avgNbrs$ is the average degree of the peers. In the simulations h is 5 and $maxNbrs$ is 4. Around 1000 messages will be exchanged by one unsatisfied peer to locate a similar peer. The messages exchanged to perform search is on a higher side for the SelflessClustering algorithm. Typically on an overlay network with 1,000 peers and $PNSP_{desired}$ value of 10, there are 200 to 300 peers that are unsatisfied before the algorithms are applied. If the number of messages exchanged to perform the search is an issue then alternate search algorithms like biased random walk and gossip based search discussed in IV may be used. We have done simulations using biased random walk that requires the exchange of a small number of messages and similar results to those described here were observed. We are not able to present the results in this paper because of space constraints.

The SelfishClustering algorithm is another alternative for clustering peers if the messages exchanged to perform clustering is an issue. This is because in the SelfishClustering algorithm the peers do not search for other peers on the overlay network. In SelfishClustering algorithm there is an overhead of messages exchanged to reconnect the peers that are disconnected from the overlay network because of selfish dropping. However this overhead is far less when compared to searching for a similar peer on the network. For example, on an overlay network with 1,000 peers and a $PNSP_{desired}$ value of 10, 8,000 messages where exchanged to reconnect peers.

3) Cluster Accuracy: Figure 2 shows the accuracy of the clusters against $PNSP_{desired}$ after applying the SelflessClustering and SelfishClustering algorithms. The accuracy of the clusters created by the Selfless clustering algorithm is extremely low. This means that the failure of a few connections could adversely affect the connectivity within the clusters. One way to improve the cluster accuracy could be a periodic exchange of routing table information between the peers in the cluster to create new connections within the cluster.

The percentage of the total population of the peers that are a part of the clusters with the highest number of peers for each category varies between 98 % to 100 % for the SelflessClustering algorithm and 10 % to 30 % for the SelfishClustering algorithm. In the SelflessClustering algorithm almost all the peers belong to the cluster with the highest number of peers for their category. Because the cluster sizes are very large and the SelflessClustering algorithm does not makes an attempt to increase the connection between the peers the clustering accuracy is very low.

4) Disconnected Topology: A critical value of $PNSP_{desired}$ (called $PNSP_{critical}$) was observed above which the overlay network's topology was disconnected. The value of $PNSP_{critical}$ is different for different networks. The authors were not able to find any correlation between the network and the $PNSP_{critical}$ value. A typical value of $PNSP_{critical}$ is 40 for SelflessClustering algorithm and 20 for SelfishClustering algorithm.

5) Discussion: For both the clustering algorithms a low value of $PNSP_{desired}$ (e.g., 10 to 20) is sufficient to achieve a substantial decrease in the number of clusters. The SelflessClustering algorithm rearranges the overlay network topology to have approximately five clusters, one for each type of peers. The SelfishClustering algorithm that provides approximately a 70 % decrease in the number of clusters. However the cluster accuracy is far less for SelflessClustering clustering algorithm. So if the cluster accuracy is a concern then SelfishClustering algorithm is a good choice whereas if the decrease in the number of clusters is a concern then SelfishClustering algorithm is a good choice.

In a P2P overlay network that utilizes the clustering algorithms presented in this article, a peer needs to spend some time and resources before it can efficiently utilize the capability of the P2P system. The time and resource expenses might act as a deterrent for peers to leave the system after they have consumed the resource of their interest.

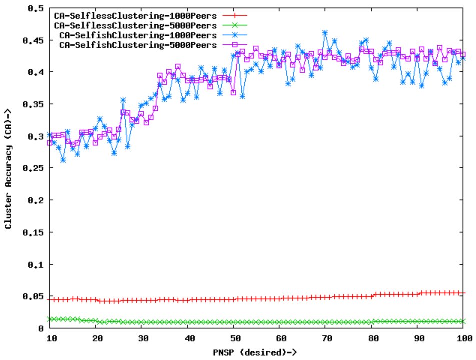

Fig. 2. A plot of cluster accuracy after applying the SelflessClustering and SelfishClustering algorithms against PNSP(desired), on overlay networks with 1,000 and 5,000 peers.

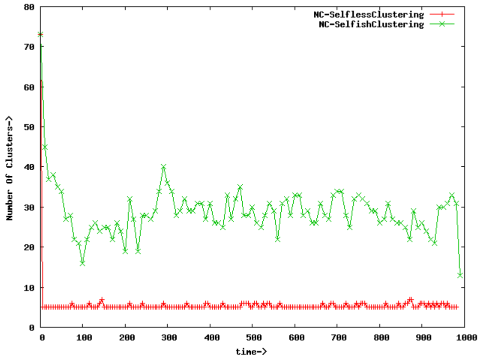

Fig. 3. The number of clusters in the overlay network against time, in a simulation in which new peers join the network every iteration.

C. Dynamic Simulations

Simulations have been done on a network that has continuous arrival of peers to show that the clustering algorithms work when there is a flux of peers on the overlay network. The network has 100 peers initially. Five peers are added to the network in every simulator iteration till there are 5,000 peers on the network. The simulations go on till all the peers are satisfied or 1,500 simulation iterations are reached. The simulations have been done using $PNSP_{desired}$ value of 10 because it is far below the $PNSP_{critical}$ value for both SelflessClustering and SelfishClustering algorithms. Figure 3 shows a plot of the number of clusters on the overlay network against time (simulator iteration) when SelflessClustering and SelfishClustering algorithms are applied on the networks described above. The number of clusters are calculated every fifth simulator iteration. The SelflessClustering algorithm maintains the value of the number of clusters between 5 and 7 even when there is an arrival of peers on the network. The number of clusters when using the SelfishClustering algorithm vary because of the reconnection of the disconnected peers. However the number of clusters remains between 10 and 40. In the SelflessClustering algorithm more than 99 % of the total number of peers are a part of the clusters with the maximum number of peers for each category. For the SelfishClustering algorithm this figure varies between 90 and 99 %.

VI. Related Work

Based on the criteria used for clustering, the existing decentralized algorithms for clustering peers can be divided into two major categories: content-based clustering and distance-based clustering. The content-based clustering algorithms (e.g., [26], [8], [9], [27] and [28]) are designed to create clusters of peers that have a similar property. For file-sharing applications this property is typically the type of content that a peer is sharing. The content-based clustering algorithms are accompanied by a routing algorithm that directs search requests to the most suitable cluster for handling the request. The distance based clustering algorithms (e.g., [5], [3] and [4]) are optimized to cluster peers that are close to each other on the underlying physical network.

Hang et al. proposed one of the earliest algorithms for content-based clustering [9]. In their algorithm, a peer P obtains information about other peers by flooding the network with a request for information about other peers. The peer P then uses this information to establish connection with peers that share similar content. In this algorithm, the peers will keep executing the expensive operation of flooding the network for topology adaptation even when they already have the neighbors required for the desired topology. In comparison, the peers using the clustering algorithms presented in this paper do not consume the network resources to search for new neighbors once the desired topology has been achieved. In the SelflessClustering algorithm the peers will only search for new neighbors if the topology desired by a peer is altered due to peers leaving or joining the overlay network.

In [28], a new peer which is in the process of joining the overlay network obtains information about other peers that share similar content and then establishes connections to them. The system is in a clustered state before the peer joins, and the choice of neighbors assures that it remains in a clustered state. While this is a conceptually elegant model, it is not without problems. Typically, a joining peer has only minimal knowledge of the network's topology and this knowledge is often limited to just one bootstrap node. Unless the bootstrap node happens to share similar content to the new peer, it is unlikely that its position in the clusters will be very useful for the new node, and an expensive search operation may be required to find suitable neighbors.

The distance-based clustering algorithms proposed in [5], [3] and [4] use a similar approach which involves using selected landmark peers. The peers measure their underlying network distance to the landmark peers and connect themselves to the landmark peer that is closest to them. This approach results in clusters that are based around the landmark peers. If the landmark peers are not evenly distributed across the overlay network's topology then this algorithm would result in unevenly sized clusters. The major drawback of these algorithms when compared to the clustering algorithms presented in this paper is their dependency on centralized components (landmark peers) which determine the clustering in the topology.

VII. Conclusion

The paper has demonstrated that the Schelling's model can be used effectively for adapting P2P network topology in a self-* manner. The paper presents two algorithms, SelfishClustering and SelfishClustering; based on the abstract Schelling's algorithm that can be used for clustering. Simulations are used to demonstrate that the algorithms can be used to bring together similar peers on the overlay network even when there is a continuous arrival of peers.

Acknowledgment

The work described in this article is funded by the Irish Research Council for Science, Engineering and Technology (IRCSET) under the Basic Research Grants Scheme.

References

[1] D. S. e. a. Milojicic, "Peer-to-peer computing," HP Labs, Tech. Rep., 2002.

[2] "Kazaa," http://www.kazaa.com/us/index.htm.

[3] R. S., H. M., K. R., and S. S, "Topologically-aware overlay construction and server selection," in *Proceedings of IEEE INFOCOM 2002*, 2002.

[4] A. Agrawal and H. Casanova, "Clustering hosts in p2p and global computing platforms," in *Proceedings of the 3rd IEEE/ACM International Symposium on Cluster Computing and the Grid (CCGRID'03)*, 2003.

[5] L. Ramaswamy, B. Gedik, and L. Liu, "Connectivity based node clustering in decentralized peer-to-peer networks," in *Proceedings of the Third International Conference on Peer-to-Peer Computing (P2P03)*, 2003.

[6] W. Zhang, S. Zhang, Y. Ouyang, F. Makedon, and J. Ford, "Node clustering based on link delay in p2p networks," in *ACM symposium on Applied Computing (SAC)*, 2005.

[7] F. K. Fessant, S. Handurukande, A. M. Kermarrec, and L. Massoulie, "Clustering in peer-to-peer file sharing workloads," *LNCS*, vol. 3279, pp. 217–226, 2004.

[8] A. Loser, F. Naumann, W. Siberski, W. Nejdl, and U. Thaden, "Semantic overlay clusters within super-peer networks," *Lecture Notes in Computer Science*, vol. 2944, pp. 33–47, 2004.

[9] C. H. Ng and K. C. Si, "Peer clustering and firework query model," in *The Eleventh International World Wide Web Conference*, 2002.

[10] A. Singh and M. Haahr, "Topology adaptation in p2p networks using schellings model," workshop on Emergent Behaviour and Distributed Computing, PPSN 2004.

[11] T. C. Schelling, "Models of segregation," *American Economic Review*, vol. 59, no. 2, pp. 488–93, May 1969.

[12] S. C. T., *Micromotives and Macrobehaviour*. Nortan and Company: W. W. Norton,, 1978.

[13] Schelling, "Dynamic models of segregation," *Journal of Mathematical Sociology*, 1971.

[14] U. Wilensky, "Netlogo segregation model." Center for Connected Learning and Computer-Based Modeling, Northwestern University, Evanston, IL. [Online]. Available: http://ccl.northwestern.edu/netlogo/models/Segregation

[15] J. Rauch, "Seeing around corners," *The Atlantic Monthly*, April 2002.

[16] A. Singh and M. Haahr, "Creating an adaptive network of hubs using schelling's model," *Communications of the ACM*, vol. 49, no. 3, pp. 69–73, 2006.

[17] J. Zhang, "A dynamic model of residential segregation," *Journal of Mathematical Sociology*, vol. 28, pp. 147–170, 2004.

[18] ——, "Residential segregation in an all-integrationist world," *Journal of Economic Behaviour and Organization*, vol. 54, pp. 533–550, 2004.

[19] H. P. Young, *Individual strategy and social structure*. Princeton University Press, 1998.

[20] M. Ripeanu, "Peer-to-peer architecture case study: Gnutella network," in *First International Conference on Peer-to-Peer Computing (P2P2001)*, 2001.

[21] P. Adamic, Lukose and Huberman, "Search in power-law networks," *Physical Review Vol 64*, 2003.

[22] Y. Chawathe, S. Ratnasamy, L. Breslau, N. Lanham, and S. Shenker, "Making gnutella-like p2p systems scalable," *SIGCOMM Germany*, August 2003.

[23] A. Montresor, "A robust protocol for building superpeer overlay topologies," Department of Computer Science, University of Bologna, Italy, Tech. Rep. UBLCS-2004-8, May 2004.

[24] A. Barbasi and Jeong, "Mean-field theory for scale-free random networks," *Physica*, 1999.

[25] S. van Dongen, "A cluster algorithm for graphs," CWI, Tech. Rep. INS-R0010, May 2000.

[26] S. Voulgaris, A. Kermarrec, L. Massoulié, and M. van Steen, "Exploring semantic proximity in peer-to-peer content searching," in *Proceedings of the 10th IEEE International Workshop on Future Trends of Distributed Computing Systems (FTDCS04)*, 2004.

[27] K. Sripanidkulchai, B. Maggs, and H. Zhang, "Efficient content location using interest-based locality in peer-to-peer systems," in *International Conference on Computer Communications (INFOCOM)*, vol. 3, 2003, pp. 2166–2176.

[28] M. Bawa, G. Manku, and P. Raghavan, "Sets: Search enhanced by topic segmentation," in *26th Annual International ACM SIGIR Conference*, 2003.