A Measure for Cluster Cohesion in Semantic Overlay Networks

Paraskevi Raftopoulou¹,² paraskevi@intelligence.tuc.gr
Euripides G.M. Petrakis² petrakis@intelligence.tuc.gr
¹ Max-Planck Institute for Informatics, Saarbruecken, 66123, Germany
² Technical University of Crete, Chania, Crete, 73100, Greece

ABSTRACT
Semantic overlay networks cluster peers that are semantically, thematically or socially close into groups by means of a rewiring procedure that is periodically executed by each peer. Rewiring proceeds by establishing new connections to similar peers, and by discarding connections that are outdated or pointing to dissimilar peers. This process aims at improving clustering quality (how well peers with similar content are clustered together) and by this, at improving the flow of information in the network by reducing the number of messages that are exchanged. Therefore, measuring the quality of clustering is an important issue by itself. This is exactly the issue this work is dealing with. In this paper, we introduce a new clustering measure that takes into account the whole neighborhood of a peer (rather than its direct neighbors) thus, providing better insight on the quality of the underlying clustered organisation. Our experimental evaluation with real-word data and queries confirms our assumption that the new measure is better suited for measuring clustering quality than other known measures, such as the (generalised) clustering coefficient.

Categories and Subject Descriptors
H.3.3 [Information Search and Retrieval]: [Clustering; Search process]

General Terms
Experimentation, Measurement

Keywords
Clustering Measure, Semantic Overlay Networks, Information Retrieval

1. INTRODUCTION
Over the last years unstructured overlays, i.e., networks where overlay links are established in an arbitrary way, have evolved as a natural decentralised way to share data and services over the Internet. Semantic Overlay Networks (SONs) [5, 14, 23, 20] are an instance of unstructured networks. In a SON, peers that are semantically, thematically or socially close (i.e., peers sharing similar interests or resources) are organised into groups to exploit these similarities at query time. SONs, while being highly flexible, improve query performance and guarantee high degree of peer autonomy. This technology has proven useful not only for information sharing in distributed environments, but also as a natural distributed alternative to Web 2.0 application domains. Contrary to structured overlays that focus on providing accurate location mechanisms, SONs are better suited for loose peer-to-peer (P2P) architectures due to better support of semantics and their natural emphasis on peer autonomy.

Peer organisation in a SON is achieved through a rewiring protocol, that is (periodically) executed by each peer. The purpose of this protocol is to update existing connections among similar peers. This is achieved by establishing new connections to similar peers, and by discarding connections that are outdated or pointing to dissimilar peers. The goal of a rewiring protocol is to create clusters of peers with similar interests. Queries can then be resolved by identifying which cluster is better suited to answer the query and by routing the query towards a peer in that cluster. Once reaching a cluster relevant to the query, the peer receiving it is responsible for forwarding it to other peers within the same cluster. Therefore, having peers organised into clusters is of paramount importance for improving system efficiency and retrieval effectiveness.

Several works on peer organisation using SONs (e.g., [11, 23, 20, 2, 19]) are based on the idea of small-world networks [28]. A small-world network is a type of graph in which nodes are not neighbors of one another, but can be reached from every other node by a small number of hops or steps. The main characteristics of small-world networks, as appear in [29], are (i) the small average shortest path length, and (ii) the high clustering coefficient. By the first characteristic, it follows that most pairs of peers will be connected by at least one short path. In this way, queries can be efficiently, in terms of network traffic, routed towards a relevant peer. On the other hand, clustering coefficient introduced by Watts and Strogatz [29] is related to the formation of cliques. It follows that in a P2P network high values of clustering coefficient will cause high representation of cliques, and subgraphs that are characterised by connections between almost any two peers within them. This high connectedness between the peers within a cluster can even-
tually result in a decreased number of peers reached by a query, and consequently in low retrieval effectiveness.

Some research proposals modify [3, 4], or generalise [7] the clustering coefficient. To the best of our knowledge, the work presented in this paper is the first to focus on information retrieval (IR) on top of SONs, and to introduce a measure that quantifies the underlying (dynamic) P2P structure (for directed and undirected networks) by focusing on the retrieval effectiveness of the network, i.e. the higher the value of this measure is the better the performance of retrievals.

In this paper, we introduce a new measure for assessing on the quality of clustering in SONs, referred to hereafter as clustering efficiency $\bar{\kappa}$ measure. Unlike clustering coefficient that takes into account only the immediate neighbors of a peer, the proposed measure considers information on how well all peers containing similar information are organised (clustered). The main idea behind clustering efficiency lies on the observation that rewiring may result into more than one clusters of highly-connected peers with similar interests [20]. This in turn, can result in high values of clustering coefficient, as this measure takes into account only the immediate neighbors of each peer, but in low retrieval effectiveness, as the query fails to identify all similar clusters. Thus, clustering coefficient measure is not desirable, since it fails to associate high (or low) values with the organisation quality of the network overall, and even more, with the anticipated performance of searches over the network: the clustering coefficient can be high (e.g., peers within a cluster are highly-connected), but at the same time recall can still be very low (e.g., peers with the same interests are organised in different clusters). It has been shown in [20] that the retrieval procedure can still address loosely-connected clusters of peers with the same interests and achieve good performance of retrievals. As will be shown in the experiments, clustering efficiency is more successful than clustering coefficient in associating network organisation quality with the performance of retrievals.

The remainder of the paper is organised as follows. Section 2 presents research proposals that implement SON-like structures to support IR functionality and research on clustering measures, while Section 3 discusses a generic SON architecture and the related protocols. Section 4 presents the proposed clustering measure that better reflects peer organisation in terms of IR. The experimental results are presented in Section 5, followed by conclusions in Section 6.

2. RELATED WORK

Recent research on IR approaches implementing SON-like structures for supporting content search in a distributed collection of peers includes the work of Lu et al. [15], where a two-tier architecture is proposed. In this architecture, a peer provides content-based information about neighboring peers and determines how to route queries in the network. Along the same lines, Klampanos et al. [9] propose an architecture for IR-based clustering of peers, where a representative peer (hub) maintains information about all other hubs and is responsible for query routing. The notion of peer clustering based on similar interests rather than similar documents is introduced in the work of Spripanidkulchai et al. [25]. In a similar spirit, Parreira et al. [18] introduce the notion of peer-to-peer dating that allows peers to decide which connections to create and which to avoid based on various usefulness estimators. Loser et al. [13] propose a three-layer organisation of peers (based both on peer content and usefulness estimators) and suggest combining information from all layers for routing queries. Koloniari et al. [10] model peer clustering as a game, where peers try to maximise the recall for their local query workload by joining the appropriate clusters. Other works in the area include the embedding of metric spaces in the SON paradigm, as in [12, 24], in an effort to broaden the applicability of SONs.

Another track of work on peer organisation using SONs is based on the idea of small-world networks. In [11], Li et al. propose creating a self-organising semantic small-world network based on the semantics of data objects stored locally to peers. Along the same lines, Schmitz [23] assumes that peers share concepts from a common ontology, and this information is used for organising peers into communities (small-worlds) with similar concepts. iCluster [20] extends this idea of peer organisation in small-world networks by allowing peers to have multiple and dynamic interests. In DESENT [2], SONs are organised as a hierarchy of clusters to support a digital library application. Each cluster is represented by a cluster gateway, while groups of clusters form super-clusters with their own gateways. To achieve efficient routing, each cluster gateway maintains information about all other cluster (and super-cluster) representatives.

Over the past few years, a wide range of concepts and measures quantifying network clustering have been proposed and investigated. Watts and Strogatz [29] introduce the clustering coefficient measure to determine whether a graph is a small-world network. The clustering coefficient of a vertex in a graph quantifies how close the vertex and its neighbors are from being a clique (complete graph). Hansen et al. [7], working on protein interactions, present a generalised version of the classical clustering coefficient measure. This work aims towards a measure that can be applied to all types of networks (e.g., networks with directed links) and can provide information about the underlying structure within the networks. In [4], Fronczak et al. assert that the standard clustering coefficient measure does not provide any useful insights of complex network structure and dynamics. This work extends the standard clustering coefficient by introducing higher order clustering coefficients that describe interrelations between vertices belonging to the nearest neighbourhood of a certain vertex in the complex network. Other works in the area, as for example [1, 21], discuss indices that measure the quality of a graph clustering mechanism. The works referred to above, mainly aim at clustering per se. In almost all studies, the authors consider a graph or a network and propose clustering techniques that maximise some clustering measure.

In [3], Forstner and Charaf propose a protocol for clustering P2P networks and evaluate this protocol using a modified clustering coefficient measure. They idea of the proposed protocol is to arrange peers in a topology so as to eliminate counterproductive links and minimise network traffic, while performing successful queries. A query is considered successful when a document matching the query is retrieved. This work applies to mobile environments, so network traffic is of major importance. Contrary, in our paper IR is treated as an important issue; we propose a peer rewiring process, and then identify a clustering measure that quantifies the underlying network structure, aiming at reflecting retrieval effectiveness.
3. OVERVIEW

In this section, we describe a generic SON architecture that aims at IR functionality, and summarise the related protocols. Notice that although our description is based on a home-brewed system called iCluster [20], all SON-based systems follow a similar architecture: a periodic rewiring protocol used to cluster peers with similar interests [23, 11, 27], and a firework-like technique [26, 17] used to route the issued queries. In iCluster, the interests of a peer are identified using its local document collection, and a single peer has a tunable and dynamic number of interests depending on its capabilities, collection size and content diversity. iCluster was the first system to overcome the specialisation assumption [16] common in SONs.

3.1 Architecture

We consider a P2P network, where peers are responsible for serving both users searching for information and users contributing information to the network. These peers represent the message routing layer of the network, run the peer rewiring protocol and form clusters based on their likelihood to contain similar content. Each peer is characterised by its information content. Peers’ documents are represented by a vector of terms in the spirit of Vector Space Model (VSM), which may be either automatically (by text analysis) or manually assigned to each document (e.g., tags or index terms). To identify its interests, a peer categorises its documents (each document may belong in multiple categories) using an external reference system, an ontology, or unsupervised clustering methods [6]. Each peer maintains a routing index holding information for short- and long-range links to other peers. Short-range links correspond to intra-cluster information (i.e., links to peers with similar interests), while long-range links correspond to inter-cluster information (i.e., links to peers having different interests).

The role of the P2P network is two-fold: it acts as the glue between information producers and consumers of information through a distributed self-organising repository that can be queried efficiently, and serves as a fault-tolerant and scalable routing infrastructure.

3.2 Basic protocols

The main idea behind SONs is to let peers self-organise into clusters of similar content. Then, query execution is performed by addressing the cluster of peers with content similar to the issued query. In this section, we discuss the basic protocols that specify how peers join or leave a SON, how peers self-organise into clusters, and how query processing is carried out.

3.2.1 Joining protocol

The first time a peer \( p_i \) connects to the network, it has to follow the join protocol. Initially, \( p_i \) categorises its documents, which may belong to more than one categories. Consequently, \( p_i \) may have more than one interests stored in its interest list \( I(p_i) \). Since documents are represented by term vectors, naturally, each document category is also represented by its centroid vector (i.e., the mean vector of the vector representations of the documents it contains).

For each distinct interest \( I_{jk} \), peer \( p_i \) maintains a separate routing index \( RI_{jk} \), which contains short-range links and long-range links. Entries in the routing index are of the form \((ip(p_j), I_{jk})\), where \( ip(p_j) \) is the IP address of peer \( p_j \) and \( I_{jk} \) is the \( k \)-th interest of \( p_j \). The number of routing indexes maintained by a peer equals the number of its interests. Peers may merge or split their routing indexes by merging or splitting their corresponding interests. A routing index is initialised as follows: peer \( p_i \) collects in \( RI_{jk} \) the IP addresses of \( s \) randomly selected peers. These links will be refined according to the interest \( I_{jk} \) of \( p_i \) using the peer rewiring protocol described in the next section.

In the following, for simplicity of the presentation, we assume that each peer \( p_i \) has only one interest \( I_i \). However, our results on rewiring can be extended to the case of multiple interests, since peers maintain one routing index per interest in the spirit of [20]. Notice that by assuming one interest per peer, our protocol description follows the assumptions of most existing SON-based systems [23, 26, 17].

3.2.2 Rewiring protocol

Peer organisation proceeds by establishing new connections and by discarding old ones, producing clusters of peers with similar interests. Each peer \( p_i \) periodically initiates a rewiring procedure by computing the intra-cluster similarity (or neighborhood similarity) \( NS_i = \frac{1}{s} \sum_{p_j \in RI_i} \text{sim}(I_i, I_j) \), where \( s \) is the number of short-range links of \( p_i \) (i.e., peers in the neighborhood of \( p_i \)), \( I_i \) is the interest of peer \( p_i \) contained in the \( RI_i \), and \( \text{sim}() \) is the cosine similarity of the VSM. The neighborhood similarity is used here as a measure of cluster cohesion. If \( NS_i \) is greater than a threshold \( \theta \), then \( p_i \) does not need to take any further action, since it is surrounded by peers with similar interests. Otherwise, \( p_i \) creates and issues a \text{FindPeers}() message to all its neighbours, containing the IP address and interests of peers discovered while the message traverses the network. System parameters \( \theta \) and \( \tau_R \) need to be known upon bootstrapping. Notice that a similar notion of cluster cohesion has been utilised in many of the existing SON-based systems [17, 23, 20] to trigger the rewiring procedure.

A peer \( p_j \) receiving the \text{FindPeers}() message appends \( (ip(p_j), I_j) \) to \( L \), reduces \( \tau_R \) by one and forwards the message to \( m \) neighbor peers \((m \leq s)\). The \text{FindPeers}() message can be forwarded using one of the following strategies:

1. The message is forwarded to the \( m \) neighbor peers \((m \leq s)\) with interests most similar to \( I_i \). This message forwarding technique is referred to in the literature as gradient walk (GW) [22, 23]. The idea behind the GW strategy is to forward the rewiring message to regions of the network that contain clusters of peers with interests similar to \( I_i \).

2. The message is forwarded to \( m \) randomly chosen peers stored in \( p_j \)'s routing index \( RI_j \). The idea behind this message forwarding technique, called random walk (RW) strategy, is to explore the network for peers with interests similar to \( I_i \) (the interest of the message initiator \( p_i \)) by making no assumption on the clustering of the network.

3. The message is forwarded with equal probability either to (i) a set of \( m \) randomly chosen peers, or (ii) the set of \( m \) peers with interests most similar to the interest \( I_i \) of the initiator peer \( p_i \). This strategy, called GW+RW strategy, aims at combining the benefits from the GW
Procedure Rewiring($p_i, I_i, \tau_R, \theta, m, \varrho$)

Initiated by $p_i$ when neighborhood similarity $NS_i$ drops below $\theta$.

input: peer $p_i$ with interest $I_i$ and routing index $RI_i$
output: updated routing index $RI_i$

1: compute $NS_i = \frac{1}{\varrho} \sum_{j:p_j \in RI_i} \text{sim}(I_i, I_j)$
2: if $NS_i < \theta$ then
3: $L \leftarrow \emptyset$
4: create $\text{FindPeers}()$
5: forward $\text{FindPeers}()$ message using //strategy $S = (GW,RW,GR+RW)$
6: forward($\text{FindPeers}(), S$)
7: let $p_j$ be a neighbor of $p_i$ receiving $\text{FindPeers}()$
8: generate a random number $x$
9: if $x \leq \varrho$ then
10: update $RI_j$, using $L$
11: $L \leftarrow L \cup (ip(p_j), I_j)$
12: $\tau_R \leftarrow \tau_R - 1$
13: repeat the above procedure for $p_j$’s neighbors
14: until $\tau_R = 0$
15: return list $L$ to $p_i$
16: update $RI_i$ with information from $L$

Figure 1: The rewiring protocol.

and RW strategies by providing a non-deterministic choice between the two methods.

Finally, when $\tau_R = 0$, the $\text{FindPeers}()$ message is sent back to the message initiator $p_i$.

A peer receiving a $\text{FindPeers}()$ message may exploit the information contained in the message according to refinement probability parameter $\varrho$. This parameter takes values in the interval $[0, 1]$: when $\varrho = 0$, no peer apart from the message initiator may use the contents of $\text{FindPeers}()$ message, while when $\varrho = 1$, all peers may exploit the information contained in the $\text{FindPeers}()$ message to update their routing indexes. A peer $p_j$, receiving a $\text{FindPeers}()$ message, updates its routing index $RI_j$ by replacing short-range links that are outdated or pointing to peers with dissimilar interests with links found in the message. System parameter $\varrho$ needs to be known upon bootstrapping. Figure 1 illustrates the above rewiring protocol in algorithmic steps.

3.2.3 Query processing protocol

Let us assume that a user issues a query $q$ through peer $p_i$, where $q$ is a term vector. Peer $p_i$ compares $q$ against its interest $I_i$. If $\text{sim}(q, I_i) \geq \theta$, then $p_i$ creates a $\text{Query}(ip(p_i), q, \tau_h)$ message, where $\tau_h$ is the query TTL, and forwards it to all its neighbors using the short-range links in $RI_i$. This forwarding technique is referred to as fixed forwarding [23], since forwarding proceeds until the query reaches a cluster of peers similar to $q$. All forwarding peers execute the aforementioned protocol and reduce $\tau_h$ by one at each step of the forwarding procedure. The combination of the two parts of query routing described above is collectively mentioned in the literature as the fireworks technique [17, 26].

Notice that the value for the broadcasting TTL $\tau_b$ is different from the value of fixed forwarding TTL $\tau_f$. Typically, $\tau_b$ is smaller than $\tau_f$, since in broadcasting the message needs to reach peers only a few hops away (i.e., in the same cluster of the message recipient). In the case of fixed forwarding the message needs to explore regions of the network that are possibly far away from the query initiator. Figure 2 presents the pseudocode for the query processing protocol.

Procedure QueryProcessing($q, p_i, \tau_f, \tau_b, \theta, m$)

Compares query $q$ against the document collection of $p_j$, retrieves matching documents, and forwards $q$ to the network.

input: query $q$ issued by peer $p_i$ and threshold $\theta$
output: list $R$ of documents similar to $q$

1: if $\text{sim}(q, I_j) \geq \theta$ then
2: compare $q$ against $p_j$’s local document collection
3: if $\text{sim}(q, d) \geq \theta$ then
4: $R \leftarrow R \cup (ip(d), m(d), \text{sim}(q, d))$
5: send message $\text{RetRes}(ip(p_j), R)$ to $p_i$
6: $\tau_h \leftarrow \tau_h - 1$
7: forward $\text{QUERY}()$ to all short-range links in $RI_j$
8: else
9: forward $\text{QUERY}()$ to $m$ neighbors of $p_i$, with interests most similar to $q$
10: $\tau_f \leftarrow \tau_f - 1$
11: repeat the above procedure for $p_j$’s neighbors
12: until $\tau_f = 0$ or $\tau_b = 0$

Figure 2: The query processing protocol.

3.2.4 Document retrieval protocol

Let us assume a peer $p_j$, receiving a query $q$, for which $\text{sim}(q, p_j) \geq \theta$ holds. Apart from executing the forwarding protocol described in the previous section, $p_j$ also applies a procedure for retrieving documents similar to $q$. Query $q$ is matched against $p_j$’s local document collection, and all documents $d$ with $\text{sim}(q, d) \geq \theta$ are retrieved and ordered by similarity to the query. Subsequently, $p_j$ creates a result list $R$ containing tuples of the form $(ip(d), m(d), \text{sim}(q, d))$ for each relevant document $d$, where $ip(d)$ is a pointer to a document and $m(d)$ are metadata describing $d$ (e.g., document title, author and an excerpt of the document’s text in the style of search engine result presentation). The resulting list is placed in a message of the form $\text{RetRes}=(ip(p_i), R)$ and is returned to the peer that initiated the query using the contact information contained in the $\text{QUERY}()$ message. In this way, query initiator $p_i$ accumulates the results obtained by different peers, merges the different lists in a single list that contains unique entries sorted by descending similarity, and presents the results to the user.

4. MEASURING CLUSTERING QUALITY

In what follows, we introduce clustering efficiency $\kappa$ as a measure that quantifies the network organisation by exploiting the underlying network structure. We also present two
other measures appearing in the literature (to compare them against our measure): (i) clustering coefficient [29], which is the measure typically used to evaluate network organisation, and (ii) generalised clustering coefficient [7], which tries to broaden the clustering concept and to take into account the whole neighborhood around a peer.

### 4.1 (Generalised) Clustering coefficient

Clustering coefficient [29] is a measure widely used [23, 8, 20] to describe the effect of peer clustering. The clustering coefficient \( c_i \) for peer \( p_i \) is formally defined as the ratio of links between the peers within \( p_i \)'s neighborhood (i.e., peers contained in routing index \( RI_i \)) with the number of links that could possibly exist between them. If \( s \) is the number of peers in the routing index of \( p_i \), then \( p_i \) can connect to \( s(s-1) \) other peers in its neighborhood. Then, the clustering coefficient is given as:

\[
c_i = \frac{\left| \{p_j, p_k\} \right|}{s(s-1)}, \quad p_j, p_k \in RI_i, \quad p_k \in RI_j
\]  

(1)

The clustering coefficient measure is equal to 1 if every neighbour connected to \( p_i \) is also connected to every other peer within the neighborhood, and equal to 0 if no peer that is connected to \( p_i \) connects to any other peer that is connected to \( p_i \).

The clustering coefficient for the whole network is defined as the average of the clustering coefficient of all peers in the network:

\[
\bar{c} = \frac{1}{N} \sum_{i=1}^{N} c_i
\]  

(2)

As can be seen from the above definition, the clustering coefficient measure takes into account only the immediate nearest neighbors of a peer and the links between these peers. However, this may not always correspond to the underlying network structure. Figure 3 presents three small networks. All have the same number of peers and links but different structure. The clustering coefficient of the central peer is the same for all cases (equals 1/3).

The so-called generalised clustering coefficient [7] tries to broaden the clustering concept and takes into account a local sub-network characterised by a radius \( r \) around a peer \( p_i \). The generalised clustering coefficient \( gc_i \) of a peer \( p_i \) is formally defined as the number of paths of length \( n \) to peers that can also be reached by a path of length \( m \) from peer \( p_i \), divided with the number of all possible paths that could exist in \( p_i \)'s sub-network:

\[
gc_i = \frac{p_i(m,n)}{\prod_{j=0}^{n-1} (N^* - j)}
\]  

(3)

where \( N^* \) is the number of peers around \( p_i \). The generalised clustering coefficient takes values in the interval \([0,1]\), and is reproduced to classical clustering coefficient when \( r = 1 \). The generalised clustering coefficient for the whole network \( gc \) is defined as the average of the generalised clustering coefficient of all peers in the network.

### 4.2 Clustering efficiency

To describe the connectivity of a network in a formal way, we introduce a new network clustering measure, the so-called clustering efficiency measure \( \kappa_i \), characterised by radius \( \tau \) around a peer \( p_i \). For the definition of the clustering efficiency measure we are based on the way the network is organised into cohesive peer clusters and on the query routing protocol: when a query \( q \) reaches a similar peer \( p_i \), \( \text{sim}(q, I_i) \geq \theta \), then \( q \) is forwarded with TTL \( \tau \) to all \( p_i \)'s neighbours using the short-range links of \( p_i \). Notice that it is implicitly considered that \( p_i \)'s neighborhood consists of all peers by radius \( \tau \) around \( p_i \). Since the network is organised into clusters of similar peers and queries address these clusters, it is of great importance all similar peers to be gathered in the same neighborhood.

Formally, the clustering efficiency \( \kappa_i \) for a peer \( p_i \) is defined as the number of peers \( p_j \) similar to \( p_i \), \( \text{sim}(I_i, I_j) \geq \theta \), that can be reached from \( p_i \) within \( \tau \) hops following short-range links, divided by the total number of peers in the network similar to \( p_i \):

\[
\kappa_i = \frac{\sum_{j=1}^{N} p_j : \{d_G(p_i, p_j) \leq \tau, \text{sim}(I_i, I_j) \geq \theta\}}{\sum_{k=1}^{N} p_k : \{\text{sim}(I_i, I_k) \geq \theta\}}
\]  

(4)

where \( d_G() \) is the distance (measured in number of hops) of the peers in the graph. The clustering efficiency measure equals 1 if the neighborhood of \( p_i \) contains all peers similar to \( p_i \). Conversely, the clustering efficiency equals 0 if no peer similar to \( p_i \) is contained in the neighborhood of \( p_i \).

The clustering efficiency for the network as a whole is defined as the average (over all peers in the network) of the clustering efficiency of each peer:

\[
\bar{\kappa} = \frac{1}{N} \sum_{i=1}^{N} \kappa_i
\]  

(5)

The way clustering efficiency is defined aims to reflect retrieval effectiveness. As it will be proven in the experiments, clustering efficiency is directly associated with the performance of retrievals (the higher \( \bar{\kappa} \) is, the better the performance of retrievals will be). Clustering efficiency is a measure that gives information about the underlying network involving more than just the immediate neighbours of the peers in the clustering concept, and looks at how the network is structured at a larger scale. Figure 4 elaborates more on this. The clustering coefficient of the central peer equals 0, the generalised clustering coefficient equals 1/6, and the corresponding clustering efficiency equals 1. In what follows, we experimentally evaluate the applicability of clus-
5. EXPERIMENTAL EVALUATION

In this section, we present our evaluation of the proposed clustering measure using a real-world dataset with web documents.

Dataset. The dataset contains over 556,000 documents from the TREC-6\(^1\) collection, categorised in 100 categories, and has been previously used to evaluate IR algorithms over distributed document collections (e.g., [30]). The queries employed in the evaluation of the corpus are strong representatives of document categories, and are issued from random peers in the network.

Setup. The base unit for time used in the experiments is the period \(t\). The start of the rewiring procedure for each peer is randomly chosen from the interval \([0, 4K \cdot \bar{t}]\) and its periodicity is randomly selected from a normal distribution of \(2K \cdot \bar{t}\), in the spirit of [23, 20]. We start recording the network activity at time \(4K \cdot \bar{t}\), when all peers have initiated the rewiring procedure at least once. We used a network size of 2,000 peers, and our results were averaged over 25 runs (5 random initial network topologies, and 5 runs for each topology). The average number of peers per class for runs (5 random initial network topologies, and 5 runs for each topology). We used a network size of 2,000 peers, and our results were averaged over 25 runs (5 random initial network topologies, and 5 runs for each topology). The average number of peers per class for runs (5 random initial network topologies, and 5 runs for each topology).

The baseline parameter values used for the experiments are summarised in Table 1. The discussion about determining the right parameter values in a SON is omitted due to space constraints, and the interested reader is referred to [20].

**Table 1: Baseline parameter values.**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Symbol</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>peers</td>
<td>(N)</td>
<td>2,000</td>
</tr>
<tr>
<td>short-range links</td>
<td>(s)</td>
<td>8</td>
</tr>
<tr>
<td>long-range links</td>
<td>(l)</td>
<td>4</td>
</tr>
<tr>
<td>similarity threshold</td>
<td>(\theta)</td>
<td>0.9</td>
</tr>
<tr>
<td>rewiring probability</td>
<td>(\varphi)</td>
<td>0.5</td>
</tr>
<tr>
<td>rewiring TTL</td>
<td>(\tau_R)</td>
<td>4</td>
</tr>
<tr>
<td>fixed forwarding TTL</td>
<td>(\tau_f)</td>
<td>6</td>
</tr>
<tr>
<td>broadcast TTL</td>
<td>(\tau_b)</td>
<td>2</td>
</tr>
</tbody>
</table>

**Performance measures.** We use the clustering quality measures presented in Section 4, i.e., clustering coefficient \(\bar{c}\), generalised clustering coefficient \(\bar{gc}\) and clustering efficiency \(\bar{k}\), to directly evaluate peer organisation. The IR effectiveness (and indirectly the clustering quality) is evaluated using recall, i.e., the number of relevant documents retrieved at a specific point of time over the total number of relevant documents in the network. Notice that precision is always 100% in our approach, since only relevant documents are retrieved.

5.1 Using different forwarding strategies

Figures 5(a) and (b) illustrate peer organisation as a function of time for the different forwarding strategies discussed in conjunction with the rewiring protocol of Section 3.2.2. Figure 5(a) presents the clustering coefficient measure \(\bar{c}\), while Figure 5(b) presents results for the clustering efficiency \(\bar{k}\). At the beginning of the rewiring procedure \((t = 4K)\) both measures are almost 0 (with all rewiring strategies), which can be attributed to poor network organisation. When all peers have executed the rewiring protocol at least once \((t = 8K)\), the effect of the different forwarding strategies to peer organisation can be observed. When the GW strategy is used, both clustering measures reach low values \((\bar{c} = 0.33 \text{ and } \bar{k} = 0.21)\) and by this we can apparently allege that the network does not manage to reach an effective peer organisation. On the other hand, when the RW, or the GW+RW strategy is used both clustering measures achieve relatively high values; we can thus, turn out that the RW and GW+RW strategies manage to efficiently and quickly organise the network and maintain an effective peer organisation. Figure 5(a) shows that the clustering coefficient reaches its highest value \((\bar{c} = 0.69)\) when the GW+RW strategy is used, driving us to the conclusion that the GW+RW strategy is the best choice for organising the network. However, the results presented in Figure 5(b) point out that the RW strategy is the best strategy for organising the peers into clusters, since by using the RW strategy clustering efficiency reaches its highest value \((\bar{k} = 0.6)\). We have also used the generalised clustering coefficient \(\bar{gc}\) to evaluate the network organisation. Table 2 presents the highest values achieved for each one of the clustering measures (i.e., when the network is organised) when using different forwarding strategies.

The purpose of the next figure is to associate the performance of retrievals with the quality of clustering and by this, recommend the clustering measure that best represents the association between the two. Intuitively, the higher the clustering quality of the network is, the better the performance of retrieval should be.

Figure 5(c) illustrates the retrieval effectiveness of the network as a function of time for the different forwarding strategies. At the beginning of the rewiring procedure \((t = 4K)\) the values of recall are low (around 35%). We can observe the effect of the different forwarding strategies to peer organisation and consequently to retrieval effectiveness. The GW strategy improves recall only by 3%, which can be attributed to poor network organisation. On the other hand,
the RW and GW+RW strategies present much higher values of recall since they achieve better peer clustering. Although the GW+RW strategy improves recall by 70%, it has worse retrieval performance than the RW strategy, which improves recall by more than 92%.

The results obtained when using clustering efficiency $\bar{\kappa}$ (Figure 5(b)) pointed out RW strategy as the best strategy for organising the network, a conclusion that corresponds to the results obtained for recall (Figure 5(c)). We can thus, make out that clustering efficiency is a good measure to evaluate network organisation, providing us with straightforward results concerning the underlying network organisation: the highest the value of the clustering efficiency is, the better the underlying network organisation is.

### 5.2 Varying the refinement probability

Figure 6 illustrates an evaluation of network organisation over time when varying $\rho$. In the set of experiments, we used the RW strategy for message forwarding, varied the values of $\rho$ and observed the way $\rho$ affects network organisation and retrieval performance. Figure 6(a) presents clustering coefficient $\bar{c}$ for $\rho = 0$ (i.e., none of the forwarding peers use `FindPeers()` message to update their short-range links) and $\rho = 1$ (i.e., all of the forwarding peers use `FindPeers()` message to update their short-range links). Initially (left-most points in x-axis), clustering coefficient is almost 0 for both values of $\rho$, which means that the network is not yet organised. However, clustering coefficient increases as peers get organised ($t = 8K$). The results indicate that when $\rho = 1$ the clustering coefficient is much higher ($\bar{c} = 0.58$) than in the case that $\rho = 0$ ($\bar{c} = 0.22$), implying that a much better network organisation is achieved in this case.

Figure 6(b) shows clustering efficiency $\bar{\kappa}$ as a function of time for $\rho = 0$ and 1 for the RW strategy. At the beginning ($t = 4K$) clustering efficiency is almost 0 for both values of $\rho$, which is attributed to poor network organisation. Clustering efficiency increases as peers get organised ($t = 8K$). The results presented in this figure illustrate that clustering efficiency is higher ($\bar{\kappa} = 0.62$) in the case that $\rho = 0$, even though the rate of increase is slower, than in the case that $\rho = 1$ ($\bar{\kappa} = 0.59$). Notice that when $\rho = 0$ the values of $\bar{\kappa}$ achieved in an organised network ($t = 16K$) have not great difference compared to the values of $\bar{\kappa}$ when $\rho = 1$.

Figures 6(a) and (b) present contradictory results concerning the values of the rewiring probability parameter $\rho$. Clustering coefficient indicates that the network is better organised when $\rho = 1$, while clustering efficiency indicates $\rho = 0$ as the best value and similar network organisations for both values of $\rho$. In what follows, we further evaluate network organisation by using recall and perceiving which network organisation improves retrieval effectiveness.

Figure 6(c) illustrates retrieval effectiveness (and thus network organisation) for the two values of $\rho$ as a function of time. When the network is unorganised ($t = 4K$) the queries cannot be routed efficiently, thus resulting in low recall (around 35%). When the network starts to organise into cohesive clusters ($t = 6K$), higher values of recall are achieved for both values of $\rho$. However, when $\rho = 0$ the recall achieved (69%) is better than in the case that $\rho = 1$ (65%). The results obtained for the retrieval effectiveness of the network object to the results obtained when using the clustering coefficient measure to evaluate the network organisation (Figure 6(a)), though coincide to the clustering efficiency results (Figure 6(b)). Figure 6(c) points out
6. CONCLUSION

We introduce a new concept for network clustering, coined clustering efficiency \( \kappa \) measure. We focused on IR on top of SONs, and presented a measure that quantifies the underlying (dynamic) P2P structure aiming in reflecting retrieval effectiveness: the higher the values of the clustering efficiency measure are, the better the underlying network organisation is in terms of retrieval effectiveness. The measure presented in this paper gives information about the underlying network by involving more than just the immediate neighbours of the peers in the clustering concept. Clustering efficiency measure gives information about whole peer neighborhoods, and looks at how the network is structured at a larger scale.

To evaluate the introduced clustering concept, we used real world data and queries and a self-organising P2P network, and measured the clustering quality both directly by using three clustering measures: (i) the classic clustering coefficient, (ii) the generalised clustering coefficient and (ii) the proposed clustering efficiency measure, and indirectly by recall. Our results indicate that clustering efficiency measure is better modelling network clustering quality.

7. REFERENCES