Content-Based Music Information Retrieval in Wireless Ad-hoc Networks

Ioannis Karydis **Alexandros Nanopoulos** **Apostolos Papadopoulos**
Dimitrios Katsaros **Yannis Manolopoulos**

Aristotle University, Thessaloniki 54124, Greece
karydis,alex,apostol,dimitris,manolopo@delab.csd.auth.gr

ABSTRACT

This paper, introduces the application of Content-Based Music Information Retrieval (CBMIR) in wireless ad-hoc networks. We investigate for the first time the challenges posed by the wireless medium and recognise the factors that require optimisation. We propose novel techniques, which attain a significant reduction in both response times and traffic, compared to naive approaches. Extensive experimental results illustrate the appropriateness and efficiency of the proposed method in this bandwidth-starving and volatile, due to mobility, environment.
Keywords: music information retrieval, content-based similarity, wireless ad-hoc, P2P.

1 INTRODUCTION

1.1 An emerging way of music distribution

Imagine jogging or resting in London's Hyde Park while listening to music through your enhanced pocket-sized ultralight device. A device that, apart from the ability to play pre-stored music like *Apple's iPod* in a area which is not covered by wireless local area networks, can also search for and acquire music songs from other people's similar musical prodigies. This is feasible, as the device is equipped with wireless connectivity and can participate in an ad-hoc network formed with devices being in its close proximity.

Being already at the end of an era for the traditional music distribution (Premkumar, 2003), the development of technologies like MP3 (and supporting applications for their distribution, e.g., *Apple's iTunes*, *MS iMusic* online music services) and the penetration of the World Wide Web, have reshaped the market model and changed consumer's buying behavior. The maturing distributed file sharing technology, implemented by peer-to-peer networks, enables the dissemination of musical content in digital forms, permitting customers an ubiquitous reach to stored music files.

The widespread penetration of the wireless networks, e.g., wireless LANs, GPRS, UMTS, creates brand new opportunities for music delivery, e.g., pioneering applications (Roccetti et al., 2005) supporting the distribution of MP3-based songs to 3G UMTS devices. Such applications are supported by the existence of a central server, which receives requests for and delivers audio files to the mobile clients. These single-hop infrastructure wireless networks are not the only possible alternative for music delivery. The emergence of wireless ad-hoc networks will probably give rise to scenario like the one previously mentioned. The salient characteristics of these networks, i.e., dynamic topology, bandwidth-constrained communication links and energy-constraint operation, introduce significant design challenges.

Limited research work dealing with the issue of delivering streaming media (audio and video) in wireless mobile ad-hoc networks (Baochun and Wang, 2003) does exist. Though, there are no prior efforts that deal with content-based music information retrieval over ad-hoc networks, where mobile clients place queries (through humming or small samples) that search for music pieces containing parts similar to the query excerpt.

1.2 Requirements posed by the wireless medium

The focal point of this work is the development of methods for searching by content in wireless ad-hoc networks, where the querier receives music excerpts matching to a posed query. Note that for the legal issues concerning the transferring and reproduction of pieces found, CBMIR applications in wireless networks, can adopt the solutions proposed in the context of online music distribution over wired P2P networks (Kalker et al., 2004).

Despite the relationship of the searching procedure with the latest approaches for CBMIR in wired P2P networks (see Section 2.1), the wireless medium poses new and challenging requirements, which call for new solutions. These requirements are summarised as follows:

1. CBMIR methods for wired P2P networks do not consider the continuous alteration of the network topology, which is inherent in wireless ad-hoc networks. One impact of this mobility is that selective propagation of the query among peers, e.g., by using data indexing like DHT (Tzanetakis et al., 2004) or caching

past queries (Kalogeraki et al. (2002) for text documents and Karydis et al. (2005b) for music), is not feasible. Additionally, the recall of the searching procedure is affected by the possibility of unsuccessful routing of the query, as well as the answers, over the changing network topology.
2. The need to reduce traffic by replacing the original query with a newly developed representation that utilises novel, appropriate transcoding schemes. Although traffic concerns CBMIR in wired P2P networks too, the requirement of traffic reduction is much more compelling in wireless ad-hoc networks due to node constraints in processing power and autonomy.
3. In CBMIR over wired P2P networks, should a matching music excerpt is found, it can immediately be returned to the querying node, since the querier is directly accessible (through its IP address). In contrast, in wireless ad-hoc networks the answers to the query have to be propagated back to the querier via the network (the querier is not directly accessible). This requirement further burdens traffic.

Although some aspects of the aforementioned issues are being considered by algorithms proposed for the problem of routing in wireless ad-hoc networks, they consider neither the peculiarities of searching for CBMIR purposes nor the size of the transferred data, since music data are considerably larger than routing packets.

1.3 Contribution and paper organisation

To address the requirements posed by the wireless medium, we propose the following techniques:

1. To fulfill the first requirement, we perform breadth-first searching over the wireless ad-hoc network using knowledge about neighbouring peers (obtained by probing neighbourhood at specific time points). This approach can cope with mobility, maintain increased final recall, and constraint the drawbacks of flooding, e.g., excessive traffic due to multiple broadcastings (explained in Section 2.3).
2. The second requirement is addressed by a technique that uses a concise, feature-based representation of the query with reducing length. The reducing-length representation (a.k.a transcoding) that we propose drastically degrades traffic, while reducing the computation performed at each peer as well.
3. For the additional traffic produced by the third requirement we propose: (i) to constraint the number of peers involved for the propagation of the answers, by exploiting any peers that were involved during the propagation of the query, (ii) to allow such peers to prune the propagation of answers, based on a property of the previously described representation.

To our best knowledge this work is the first to examine the issue of CBMIR in ad-hoc wireless networks. The contributions are: the introduction of the problem and the identification of the resulting requirements, a novel algorithm that combines the aforesaid techniques and addresses the posed requirements, and extensive experimental results, which illustrate the efficiency of the proposed algorithm.

The rest of the paper is organised as follows. Section 2 describes background and related work. Section 3 provides a complete account of the proposed method as well as two baseline algorithms. Subsequently, Section 4 presents and discusses the experimentation and results obtained. Finally, the paper is concluded in Section 5.

2 BACKGROUND & RELATED WORK

2.1 CBMIR in P2P networks

Research related to the application of CBMIR in wired P2P networks is recent. One of the first attempts (Wang et al., 2002) presents four P2P models for CBMIR, which include centralised, decentralised and hybrid categories. In another research based on a hybrid configuration (Tzanetakis et al., 2004), the authors propose a DHT-based system utilising both manually specified attributes (artist, album, title, etc.) and extracted features in order to describe the musical content of a piece. Yang (2003), proposed the utilisation of the feature selection and extraction process for CBMIR in a decentralised unstructured P2P system. Finally, Karydis et al. (2005b) investigated CBMIR in P2P networks under the time-warping distance.

This work deals with a wireless ad-hoc network, where two nodes can communicate only if in close proximity (in-range). In this type of network peers participate randomly and for short term, and when they do, they change frequently their location. These factors cause existing approaches, e.g., indexing, to become inapplicable.

2.2 Features for CBMIR

The selection of appropriate features is very important in music information retrieval. Meaningful features help in the effective representation of the objects and enable the use of indexing schemes for efficient query processing. In this work, we do not concentrate on devising new features. Instead, we are interested in a methodology for the searching procedure. Thus, we apply a feature extraction process based on the wavelet transform. Wavelet transforms provide a simple but yet efficient representation of audio by taking into consideration both non-uniform frequency resolution and impulsive characteristics (C. Roads and Poli, 1997). In detail, we consider the Haar wavelet transformation, as it is easy to compute incrementally, and for its capability concerning the capture of time dependant properties of data and overall multiresolution representation of signals (Kin-Pong Chan and Yu, 2003).

2.3 Information discovery in mobile ad-hoc networks

A wireless mobile ad-hoc network (MANET) is a collection of wireless mobile hosts forming a temporary network without the aid of any centralised administration or standard support services regularly available on the wide area network to which the hosts may normally be connected. When a source node desires to send a message to some destination node and does not already have a valid route to that node, it initiates a path discovery process to locate the destination. It broadcasts a route request to its neighbours, which then forward the request to their neighbours and so on, until the destination or an intermediate node with a route to the destination is located. Nodes are identified by their IP address and maintain a broadcast ID, which is incremented after every route request they initiate. The broadcast ID together with the node's IP address, uniquely identify a route request. In the same manner, the transmitted data requests can be identified.

There is no prior relevant work on performing content-based information retrieval in MANETs, though there is a wealth of routing algorithms. For wireless ad-hoc networks there have been proposed various routing/discovery protocols, which roughly fall into the following categories: a) table-driven routing protocols, b) source-initiated on-demand routing protocols, and c) hybrid routing protocols. Apart from the former, the remaining two families of information (node) discovery protocols rely on some form of *broadcasting*. *Flooding* is the simplest broadcasting approach, where every node in the network forwards the packet exactly once. Flooding, though, generates too many redundant transmissions, causing the *broadcast storm problem* (Ni et al., 1999).

Various algorithms have been proposed to address this problem (Lou and Wu, 2004). They can be classified as follows: a) *probabilistic approaches* (counter-based, distance-based, location-based), and b) *deterministic approaches* (global, quasi-global, quasi-local, local). The *deterministic approaches* provide full coverage of the network for a broadcast operation, by selecting only a subset of nodes to forward the broadcast packet (*forward nodes*), and the remaining nodes are adjacent to the nodes that forward the packet. The selection of nodes is done by exploiting "state" information, i.e., network topology and broadcast state (e.g., next selected node to forward the packet, recently visited nodes and their neighbour sets).

3 SEARCHING ALGORITHMS FOR CBMIR IN MANETs

3.1 Outline

The searching process is initiated at the querying peer, aiming at detecting sequences in other peers containing excerpts whose distance (defined in Section 3.2) from the query sequence Q is less than a user-defined threshold ϵ. The length of detected excerpts is equal to the length of Q.

Due to the described requirements of the wireless framework, the examined problem is formulated as follows: (i) Peers that have qualifying sequences have to be reached in a way that addresses their mobility and minimises the traffic. Because of their relative positions and of the preferred tolerance to traffic (see below), all such nodes may not be possible to reach. (ii) At each reached peer, the qualifying sequences have to be detected by detaining the peers as little as possible. (iii) Each qualifying sequence has to reach the querier in a way that reduces traffic. Notice that the answers may have to be routed back to the querier by following paths different from those through which the peers with qualifying sequences were reached, since intermediate peers may have changed their position, and therefore may be out of range. Due to this effect, every detected answer may not be possible to reach the querier. An example is illustrated in Figure 1.

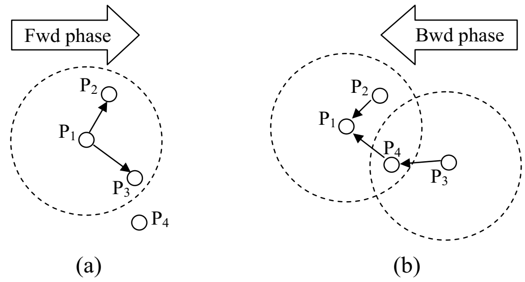

Figure 1: The querier is peer P_1. During the forward phase (a), R is received by peers P_2 and P_3. During the backward phase (b), answers can be directly returned by P_2 (still in range of P_1). Due to relative movement, P_3 is, now, out of range. Thus its answers are routed through P_4 (which was previously out the range of P_1).

To address traffic minimisation, Q is transformed to a representation form, denoted as R, through which qualifying sequences are detected. The impact of the selection of R will be explained in the following. Evidently, R must present no false-negatives. However, its particular implementation determines whether false-positives may be produced or if they will be definitely avoided. Based on all the aforementioned issues, an abstract schema to describe the entire searching process consists of the following steps:

1. User poses a query Q.
2. Q is transformed to a representation form R.
3. R is broadcast to all peers in range.
4. Qualifying sequences (true- and false- positives) detected at each peer comprise an answer-set.
5. Each answer-set is broadcast back to the querier.
6. Resolution of false-positives (possible places are: at peers that provide answers, at the querier, at intermediate peers).
7. Return of actual matches to the user/application.

In the following, the seven aforementioned steps are summarised according to four resulting events:query initialisation (involves steps 1, 2, 3), reception of R (involves steps 4, 5), reception of an answer-set (involves steps 5, 6) and answer-set reaching the querier (involves step 7).

During the propagation of R resulting from step 3, by appropriately tagging it with an ID (see Section 2.3), peers that have already received it will perform no further action, in order to avoid duplicate effort. Additionally, the propagation of R to the neighbouring peers is controlled by a parameter called *maxhop*, which is a counter that is decreased at each receiving peer. Its initial value, at the querier, is equal to $MaxHop$. This value corresponds to the preferred tolerance to traffic. The propagation of answer sets (resulting from step 5) is handled analogously, by employing again a *maxhop* parameter (the same initial value, *MaxHop*, is used).

Evidently, the searching process consists of two phases: (i) a forward phase, during which R is propagated and (ii) a backward phase, during which answers are routed back to the querier. The two phases are interleaved, since during the propagation of R by some peers, other peers are returning answers to the querier. In general, the volume of information transferred during the backward phase is larger than that of the forward phase.

In the sequel, we first present some details on the acceleration of similarity searching within each peer by using indexing. Next, we describe two algorithms (provided for comparison purposes), which follow simplistic decisions. Finally, the proposed method is studied.

3.2 Indexing within peers

In each peer, the original audio sequences are transformed to a number of multidimensional points by applying a sliding window to the audio data and by applying the Discrete Wavelet Transform (DWT) to each window. Therefore, each audio sequence produces a set of multi-dimensional points in the feature space. The dimensionality of the transformed space depends on the number of DWT coefficients that will be used for the representation. By keeping the first few DWT coefficients the size of the original audio sequence is reduced significantly.

Chan and Fu (1999) proved that no false dismissals are introduced when using this transformation technique. However, false-positives are a possibility and thus require resolution. They also proved the property of preservation for the Euclidian distance. Although this distance measure is simple, it is known to have several advantages (Keogh and Kasetty, 2002). Nevertheless, the proposed methodology does not decisively depend on the particular features and distance measure, which are used herein following simplicity as well as computation efficiency reasons.

To achieve efficient retrieval of sequences containing parts similar to the query sequence Q, the transformed audio sequences are organised by means of an indexing scheme. However, the direct indexing of multidimensional points in feature space, would lead to large storage consumption, since each audio sequence can generate a large number of multi-dimensional points. To overcome this problem we perform a grouping of points from consecutive windows, which are also close enough, with Minimum Bounding Rectangles (MBRs). MBRs are organised by means of an R^*-tree (Beckmann et al., 1990) or any other multi-dimensional access method. An analogous type of indexing for audio sequences for the purpose of similarity searching has been described by Karydis et al. (2005a).

3.3 Algorithm based on minimal query representation

Due to the previously described indexing scheme, it is evident that the minimal representation R of the query sequence Q consists of a number of DWT coefficients, which allow index probing in the examined peers. The number of coefficients used is pre-specified, since all indexes have the same dimensionality. Opting for low traffic during the forward phase, an algorithm, denoted as *CQ* (querying by *C*oefficients, resolution at *Q*uerier), selects for R this minimal representation.

The steps of CQ are summarised according to the actions performed for each occurring event (see Section 3.1), as follows:

Query initialisation: The querier sets R as the query coefficients and propagates (broadcasts) it to all its neighbours.

Reception of R: Upon the reception of R, each peer P probes its indexes and produces a list of results. Since R consists only of DWT coefficients, this step produces both true and false-positives. The answer-set (i.e., all found positives from this node) is propagated back to the querier, by broadcasting it to all neighbours of P (backward phase). Accordingly, should there be available maxhop, R is conveyed to all P's neighbouring peers (forward phase).

Reception of an answer-set: Each peer P, that is not the querier, receiving an answer-set, continues the propagation (backward phase) to all its neighbouring peers, while $MaxHop$ has not been reached.

An answer-set reaches the querier: When an answer-set reaches the querier, then the result sequences are resolved against the query sequence and all true positives are presented to the user.

By choosing R to contain only the DWT coefficients, CQ is expected to produce very low traffic for the forward phase. However, by using only the DWT coefficients, it may result to a large number of false-positives. This is due to the fact that the coefficients may not be able to perform effective pruning, especially for query sequences of larger length and for larger values of ϵ. Hence, the traffic during the backward phase is going to be significantly affected, since answer-sets will be large in number and size (recall that the backward phase is much more demanding). This is further aggravated, since CQ does not use any devisable policy for the routing of answer-sets (it performs plain propagation by broadcasting to all neighbouring peers). Accordingly, it is expected that the traffic of the backward phase will be prohibitive for CQ.

3.4 Algorithm based on maximal query representation

Considering the antipodal of the previous case, one can opt for eliminating the burden to the backward phase caused by false-positives. This can be achieved by choosing R equal to the maximum possible representation, that is, the entire query sequence Q. In order to be able to perform index probing (i.e., to avoid sequential searching at each peer), the DWT coefficients are also included in R, however its size is dominated by the size of the query sequence. The advantage of this choice is that at each peer, after some matches have been found by index probing, these results can be immediately tested against the query itself (refinement). Thus, no false-positives will be included in the answer-sets. The resulting algorithm is denoted as *QL* (full *Q*uery, *L*ocal resolution at peers). As previously, the steps of QL can be summarised according to the actions performed for each occurring event:

Query initialisation: The querier assigns to R the entire query sequence (plus the query coefficients) and propagates (broadcasts) it to all its neighbours.

Reception of R: Upon the reception of R, each peer P probes its indexes, resolves the false-positives, and produces a list of results (only true-positives). The answer-set is propagated back to the querier, by broadcasting it to all the neighbours of P (backward phase). Accordingly should there be available $maxhop$, R is conveyed to all P's neighbouring peers (forward phase).

Reception of an answer-set: Each peer P, that is not the querier, receiving an answer-set, continues to propagate it (backward phase) to all its neighbouring peers as long as there is available maxhop.

An answer-set reaches the querier: When an answer-set reaches the querier, then the result are immediately presented to the user.

In contrast to CQ, QL manages to decrease the excessive traffic of the backward phase. However, this comes at the cost of increasing the traffic of the forward phase, since the entire query is propagated. Although the backward phase is more crucial, the forward phase requires optimisation too, especially in the case of large query sequences. Moreover, like CQ, QL does not use any devisable policy for the routing of answer-sets (performs propagation by broadcasting to all neighbouring peers). Thus, the improvement of backward traffic is expected to be moderate.

3.5 Proposed algorithm

From the description of CQ and QL algorithms, it is clear that they refer to the two extremes with respect to the size of the query representation R. The result of each decision is that the backward and the forward traffic is affected, respectively. We devise a hybrid representation scheme aiming towards high performance, which is based on a sample of the query, plus the DWT coefficients (commonly used by all algorithms). A sample of the query does not burden forward traffic as much as in the case where R is equal to the entire query sequence, while at the same time, it can manage to eliminate a significant number of false-positives. A sample discards a false positive when the distance between it and the corresponding elements in the examined sequence is greater than ϵ. Evidently, the length of the sample presents a trade-off: a large sample can prune more false positives, but incurs higher traffic.

The sample used for the representation initially takes an adequate value (see the section with experimental results for its tuning) and this size is monotonically reduced during the propagation of R in the forward phase. By this transcoding scheme we attain: (i) to keep forward traffic low, (ii) the reduction in the processing time at each peer (the cost of eliminating false drops at each peer depends on the size of the representation), (iii) we can still develop a policy for pruning during the backward phase (to be explained next), thus to reduce backward traffic.

The reduction of R's size can be achieved in many ways. The reduction of the initial value is based on a sigmoid function. Due to the shape of the function, the immediate neighbourhood of the querier, which can provide results faster, receives a larger R, whereas the burden posed on peers that are far is appreciably smaller.

As mentioned, the reduced representation size does not inhibit the development of a policy to optimise backward traffic, compared to the plain propagation of answer-sets to all neighbours, which is used by QL and CQ. The optimisation is as follows. During the forward phase, each peer that receives R, additionally receives the ID of all peers in the path that was followed from the querier to it. In the example depicted in Figure 2a, for peer P_4 this path consists of the IDs of peers P_1, P_2, and P_3. These IDs can be maintained along with R with minimal cost (only some bytes). When any of such peers that were involved during the forward phase starts propagating answer-sets for the backward phase, it has to select which of its current neighbouring peers will propagate these answer-set.

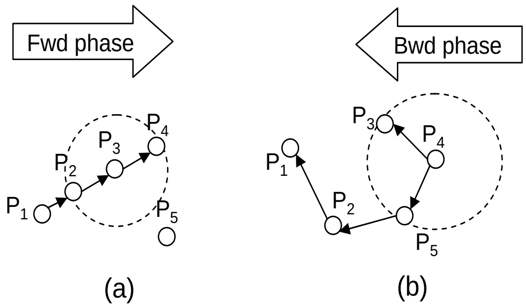

Figure 2: Example of relative locations of peers in forward and backward phase.

The simplistic decision followed by QL and CQ, is to propagate to all neighbours. In contrast, we exploit any peers that were in the path up to this peer during the forward phase, which are still in range (i.e., they are currently neighbours). Therefore, if at least a factor, denoted as *neighbour factor* (*NF*), of the current neighbours was in the path, we select them as the only ones to propagate the answer-set. If their number does not suffice, then we select randomly some of the current neighbours (not from the path) in order to achieve at least *NF* neighbours to propagate the answer-set. In contrast, if their number is larger than *NF*, then they are all selected. For instance, in the occasion depicted in Figure 2b (which is after some peers have relocated compared to Figure 2a), and by assuming that *NF* is equal to two, P_4 chooses P_3 from the path, whereas P_5 is chosen randomly.

The emphasis on exploiting peers that appear in the path of the forward phase is based on that: (i) they may be more promising to reach the querier, and (ii) they may have received a representation of the initial query with larger length than that received by the current peer. In the latter case, we can reduce backward traffic by employing the selected neighbours for potential pruning of false-positives. Now it becomes easy to see why we chose a reduced-size representation, since it is possible to test against a sample with larger size and reduce the backward traffic. Notice that the aforementioned conditions are not always met due to mobility, thus this pruning may not always be possible.

For the nodes that were selected at random in order to fulfill the *NF* parameter, we still provide to them the knowledge of the path of the peer that initiated the propagation of the answer-set (for the previous example, P_5, which is selected by P_4, will know the path from P_1 to P_4). This way, due to mobility, it is possible for such nodes during the backward phase to find neighbours that appear in the carried path (again in the same example, P_5 finds P_2 that was in the path). Therefore, the impact of such randomly selected peers on the proposed policy may be kept at a moderate level.

The algorithm that combines all the described techniques is denoted as ST (querying by *S*ample, probable resolution at peers due to query *T*ranscoding), and is summarised as follows:

Query initialisation: The querier sets R equal to a sample with an initial size (parameter) plus the query coefficients, and broadcasts it to all its neighbours.

Reception of R: Upon the reception of R, each peer P probes its indexes, resolves as many false-positives as possible based on the received query sample of R, and produces a list of results. The answer-set is propagated back to the querier, by following the described policy for the backward phase. Accordingly, should there be available $maxhop$, R's size is reduced, and the reduced R is conveyed to all P's neighbouring peers (forward phase).

Reception of an answer-set: When a peer receives a reply, we check if it can resolve any false-positives. This is true should it had received (if any) a representation that was larger than the one that the sequences in the answer set were examined previously (i.e., at the sending peer). After any possible pruning, as long as there is available $maxhop$, the answer-set is routed backwards following the proposed policy.

An answer-set reaches the querier: When an answer-set reaches the querier, initially any remaining false-positives requiring resolution are checked, and then the results are presented to the user.

4 PERFORMANCE EVALUATION

4.1 Simulation configuration

The performance of the algorithms was compared through simulation. The settings of the simulation were as follows. The P2P network had 100 nodes. We used 300 real acoustic sequences, which correspond to various pop songs. The average duration was about 5 minutes. To account for the fact that songs (especially the popular ones) are common in several nodes, we replicated each sequence to a number of peers (default value equals to four).

Regarding the simulation of mobility, we used the GSTD simulator (Theodoridis et al., 1999), which considers points moving freely in a 2-D area. We used a squared area with size equal to 4 Km^2, whereas the transmission/reception range of each peer was set to 500 m radius. Different degrees of velocity were selected for the moving peers, adjusted by parameters of the GSTD, but due to lack of space we present results only for the average walking speed of a human (5 Km/h). Regarding ST, the default value for *NF* was 10% of the number of neighbours at each peer, whereas the default initial sample size was 10% of the query sequence's size. For all algorithms, the default value of ϵ was 0.3, and the default size (in number of elements) of the query sequence was about 64 K. Additionally, the number of DWT coefficients retained was set to eight. Finally, the default MaxHop was set to 5. When parameter values are not specified, we assume the default values.

The evaluation metrics are the average traffic (measured in MB) that each query incurs and the time the first and last result were discovered (the time of the first result is an useful measure, since users may terminate searching early). The results on time reflect the perceived latency required for the response to the querier. In contrast, total traffic reflects the load posed to the network in order to provide responses. Thus, the two factors require separate consideration. We have to notice that in all presented results, all algorithms find approximately the same number of result, i.e., they have about the same average recall (difference only in few decimal positions, due to the randomness in mobility), which guarantees a fair comparison.

We have experimented with the impact of Maxhop, query range and size, the degree of replication of music pieces, the number of peers in the network, the transmission/reception range, the sensitivity against the shape of the sigmoid function (used by ST), *NF* and initial sample factor.

4.2 Experimental results

In our first experiment, illustrated in Figure 3a, we examined the time of the first and last results against MaxHop (the time of first result is depicted as a fraction of the total time for all results, while vertical axis is logarithmic). Increase in available MaxHop produces longer times, since more peers are examined. Though, beyond a value (in this case, five) few additional results are found, thus the increase diminishes. In all cases, the increase is far more steep for CQ. ST clearly outperforms the other algorithms.

This result can be further clarified by the results on traffic, which are depicted in Figure 3b (the addition of forward and backward traffic gives the total traffic). As expected, CQ produces the lower forward traffic in all cases, QL the highest, whereas the forward traffic of ST is close enough to that of CQ (the vertical axis is logarithmic). With regard to the backward traffic, as anticipated, CQ results to prohibitive values. (For large MaxHop, CQ did not manage to finish in reasonable time, and the corresponding results are omitted.) QL considerably improves backward traffic, though ST performs significantly better (the vertical axis is logarithmic), due to the proposed routing policy for the backward phase. From this result it becomes obvious that, although the backward phase is in general more demanding for all algorithms, due to the reduction of backward traffic attained by ST, the requirement of optimising the forward phase, is fair.

Next, we examine the impact of query range. Figure 3c shows the time of the first and last result against different ranges (vertical axis is logarithmic). As expected, the performance of CQ is by far worse than the others. (For higher ϵ values, CQ could not manage to finish, thus its results for these values are omitted.) Again, ST presents the best performance. Analogous conclusions can be drawn for the traffic (see Figure 3d). As of this point on, we do not examine CQ any further, since it incessantly presented the worse performance.

We move on to examine the size of the query (measured by the number of elements in the query sequence). The results are given in Figure 3e, which presents the time of the first and last result against varying query size. The

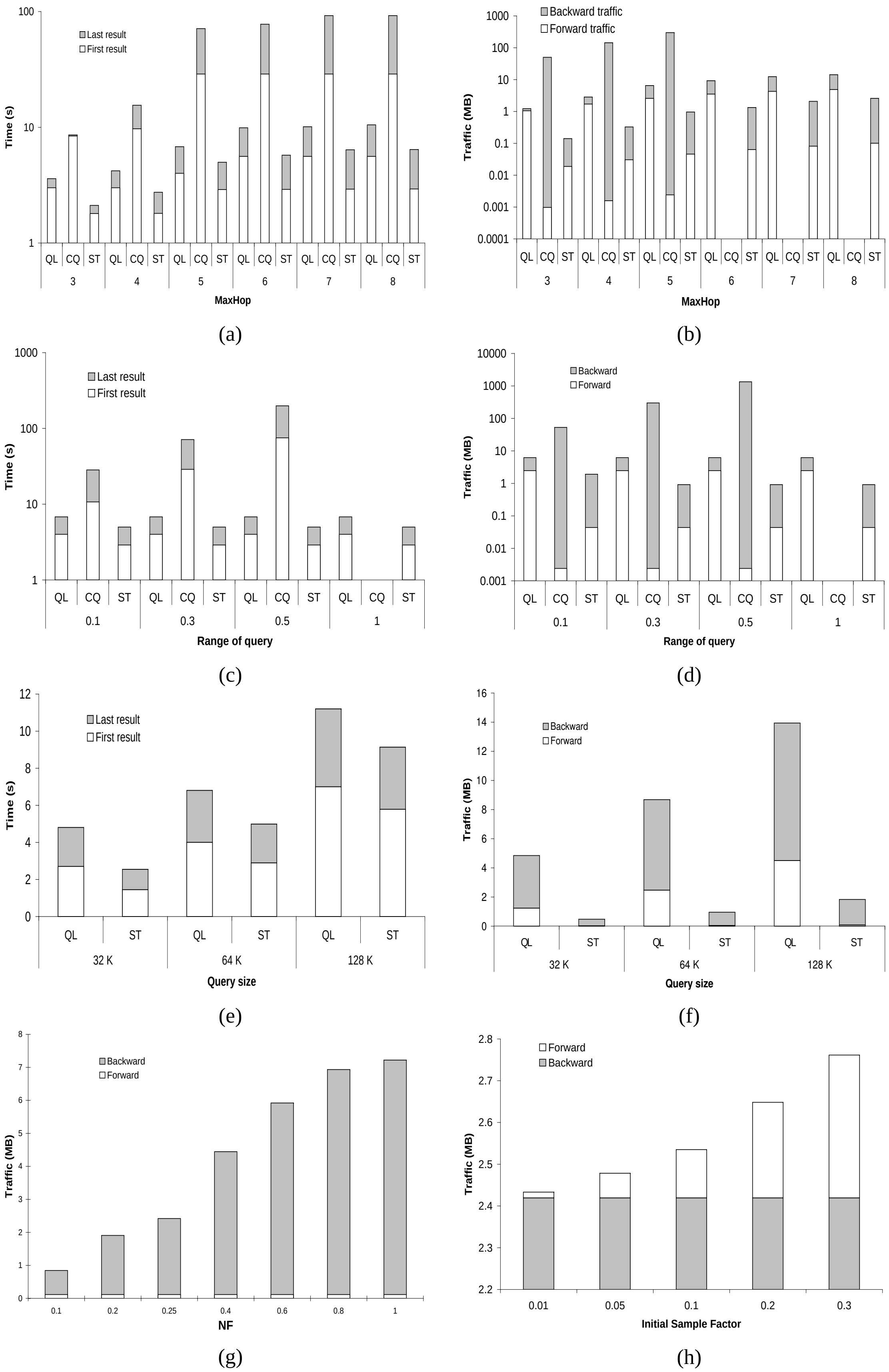

Figure 3: Experiments.

increase in query size bears an increase in response time for both QL and ST. The reason for that is the increased processing required for the determination of matching excerpts of longer queries, and the increase in transmission times and traffic due to the propagation of larger representations. The latter is illustrated in Figure 3f. For larger queries the technique used by ST in the forward phase pays-off significantly. Moreover, the policy for the backward phase manages to keep the corresponding traffic quite lower than that of QL.

Finally, we tested the sensitivity of ST against the *NF* parameter and the size of the initial sample. The results for the traffic for varying *NF* are illustrated in Figure 3g. When *NF* is high, the effectiveness of the policy for the backward phase is limited, since most peers are selected at random by this policy. Thus, the resulting backward traffic is high (forward traffic is not affected). Notice that, for the examined range of *NF* values, the reduction in traffic does affect the number of found matches (the difference between the results for the extreme values of *NF* are only in the order of decimal values). Conclusively, small *NF* values are sufficient. The resulting traffic of ST for varying initial-sample sizes are depicted in Figure 3h (for better illustration, forward traffic is depicted on the top). In this case, backward traffic is unaffected. As expected, forward traffic increases with increasing sample size. Again, for the examined range of values, the reduction in traffic is not combined with a decrease in the number of found matches. Hence, small initial sample size suffices.

5 CONCLUSIONS

In this paper, we introduce the application of CBMIR application in wireless ad-hoc networks. We recognise the new challenges posed by this type of networks. To address them, we propose a novel algorithm, which is based on a twofold optimisation: (i) the use of query representation with reducing length, (ii) a selective policy for the routing of answers, which performs additional pruning of traffic. The combination of these factors attains significant reduction in both response times and traffic. This is verified through extensive experimental results, which illustrate the superiority of the proposed method against two baseline algorithms. Concluding, we have to mention that the examined context does not depend on the specific features and distance measure, since it can be used in combination of several other ones, as long as they allow for a reducing-length representation.

ACKNOWLEDGEMENTS

This research is supported by the *ΗΡΑΚΛΕΙΤΟΣ* and *ΠΥΘΑΓΟΡΑΣ ΙΙ* national programs and funded by the *ΕΠΕΑΕΚ* framework.

References

L. Baochun and K. H. Wang. NonStop: Continuous multimedia streaming in wireless ad hoc networks with node mobility. *IEEE Journal on Selected Areas in Communications*, 21(10):1627–1641, 2003.

N. Beckmann, H. P. Kriegel, and B. Seeger. The R*-tree: An efficient and robust method for points and rectangles. In *Proc. ACM SIGMOD Conf.*, pages 322–331, 1990.

A. Piccialli C. Roads, S. Pope and G. De Poli, editors. *Musical Signal Processing*. Royal Swets & Zeitlinger, 1997.

K. Chan and A. W.-C. Fu. Efficient time series matching by wavelets. In *Proc. ICDE Conf.*, pages 126–133, 1999.

T. Kalker, D. H. J. Epema, P. H. Hartel, R. L. Lagendijk, and M. van Steen. Music2share - copyright-compliant music sharing in p2p systems. *Proceedings of the IEEE*, pages 961–970, 2004.

V. Kalogeraki, D. Gunopulos, and D. Zeinalipour-Yazti. A local search mechanism for peer-to-peer networks. In *Proc. CIKM Conf.*, pages 300–307, 2002.

I. Karydis, A. Nanopoulos, A. Papadopoulos, and Y. Manolopoulos. Audio indexing for efficient music information retrieval. In *Proc. MMM Conf.*, pages 22–29, 2005a.

I. Karydis, A. Nanopoulos, A. N. Papadopoulos, and Y. Manolopoulos. Music retrieval in p2p networks under the warping distance. In *Proc. ICEIS Conf.*, 2005b.

E. Keogh and S. Kasetty. On the need for time series data mining benchmarks: A survey and empirical demonstration. In *Proc. ACM SIGKDD Conf.*, pages 102–111, 2002.

A. W.-C. Fu Kin-Pong Chan and C. Yu. Haar wavelets for efficient similarity search of time-series: With and without time warping. *IEEE Transactions on Knowledge and Data Engineering*, 15(3), 2003.

W. Lou and J. Wu. Broadcasting in ad hoc networks using neighbor designating. In I. Maghoub and M. Ilyas, editors, *Handbook of Mobile Computing*. CRC Press, 2004.

S. Y. Ni, Y. C. Tseng, Y. S. Chen, and J.P. Sheu. The broadcast storm problem in a mobile ad hoc networks. In *Proc. ACM/IEEE MOBICOM*, pages 151–162, 1999.

G. P. Premkumar. Alternative distribution strategies for digital music. *Communications of the ACM*, 46(9):89–95, 2003.

M. Roccetti, P. Salomoni, V. Ghini, and S. Ferretti. Bringing the wireless Internet to UMTS devices: A case study with music distribution. *Multimedia Tools and Applications*, 25(2):217–251, 2005.

Y. Theodoridis, J.R.O. Silva, and M.A. Nascimento. On the generation of spatiotemporal datasets. In *Proc. SSD Symp.*, 1999.

G. Tzanetakis, J. Gao, and P. Steenkiste. A scalable peer-to-peer system for music information retrieval. *Computer Music Journal*, 28(2):24–33, 2004.

C. Wang, J. Li, and S. Shi. A kind of content-based music information retrieval method in a peer-to-peer environment. In *Proc. ISMIR Symp.*, pages 178–186, 2002.

C. Yang. Peer-to-peer architecture for content-based music retrieval on acoustic data. In *Proc. WWW Conf.*, pages 376–383, 2003.