# Music Retrieval over Wireless Ad-hoc Networks*


Ioannis Karydis[1] Alexandros Nanopoulos[1] Apostolos Papadopoulos[1]
Dimitrios Katsaros[1,2] Yannis Manolopoulos[1]

[1]Aristotle University of Thessaloniki
Thessaloniki 54124, Greece

[1,2]University of Thessaly
Volos 38221, Greece

{karydis,alex,apostol,dimitris,manolopo}@delab.csd.auth.gr



***Abstract*— Wireless networks introduce new opportunities for music delivery. The trend of using mobile devices on wireless networks, can significantly extent the recent change of paradigm in the model of music distribution, by allowing mobile clients to search for audio music in a network of wireless mobile hosts. This work, introduces the application of Content-Based Music Information Retrieval (CBMIR) in wireless ad-hoc networks. We investigate, for the first time in the literature, the challenges posed by the wireless medium and recognize the factors that require optimization. We propose novel techniques, which attain a significant reduction in both response time and network traffic, compared to naive approaches. Extensive experimental results illustrate the appropriateness, effectiveness and efficiency of the proposed method to this bandwidth-starving and volatile, due to mobility, environment.**


## I. Introduction

### A. Music distribution adopts a new paradigm

Imagine listening to music through your enhanced pocket-sized ultralight device while jogging or resting in a park. A device that, apart from the ability to play pre-stored music like any MP3 player in a area that is not covered by wireless local area networks, can also search for and acquire music songs from other people's similar musical devices. This data exchange is attainable through the device's wireless connectivity equipment allowing for participation in ad-hoc networks, formed with similar devices being in close proximity. Although such a scenario may seem futuristic, it is not so distant.

Having already reached the end of an era for the traditional music distribution [45], the market model as well as the buying behavior of consumers have been reformed by the development of technologies like MP3 (and the supporting applications for their distribution, e.g., *Apple's iTunes*, *iMusic* online music services) and the penetration of the World Wide Web. Peer-to-peer networks and the maturing distributed file sharing technology, enable the dissemination of musical content in digital forms, permitting customers an ubiquitous reach to stored music files.

New opportunities for music delivery are additionally introduced by the widespread penetration of the wireless networks (wireless LANs, GPRS, UMTS [15]) such as the pioneering applications [47] supporting the distribution of MP3-based songs to 3G UMTS devices. These applications rely on the existence of a central server, which receives requests from and delivers audio files to the mobile clients. Though, aside from these single-hop infrastructured wireless networks, music delivery can also unfold over the emerging Mobile Ad-hoc NETworks (MANETs). The wireless ad-hoc networks are peer-to-peer, multi-hop, mobile wireless networks, where information packets are transmitted in a store-and-forward fashion from source to destination, via intermediate nodes. Such networks are expected to give rise to scenarios 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.

In this paper, we focus on the following problem. We consider a number of mobile hosts that participate in a wireless ad-hoc network, where each host may store several audio musical pieces. Assume a user that wants to search in the wireless network, to find audio pieces that are similar to a given one. For instance, the user can provide an audio snippet (e.g., a musical piece excerpt) and query the network to find the peers that store similar pieces. As will be described in the following, the definition of similarity can be based on several features that have been developed (see Section IV-A) for Content-Based Music Information Retrieval (CBMIR). It is important to note that the querying host does not have any prior knowledge of neither the qualifying music pieces nor the hosts' locations that contain them. This differentiates the current problem from existing ones that are interested just in identifying the hosts in a wireless ad-hoc network that contain a known musical piece. Moreover, the examined problem is complementary to the one of delivering streaming media (audio and video) [3] in wireless ad-hoc networks, since the latter does not involve any searching for similar musical pieces, and just focuses on transferring data from one host to another.

### B. Requirements set by the wireless medium

This research focuses on the development of methods for searching audio music by content in wireless ad-hoc networks, where the querier receives music excerpts matching to a posed query. As for the legal issues of transferring and reproducing the musical pieces found are concerned, analogous issues are being confronted in online music distribution over wired P2P networks, where ways to protect intellectual property


*This research is supported by the *ΗΡΑΚΛΕΙΤΟΣ* and by a *ΓΓΕΤ* grant in the context of the project "Data Management in Mobile Ad Hoc Networks" funded by *ΠΥΘΑΓΟΡΑΣ ΙΙ* national research program. The second author is supported by IKY postdoctoral scholarship. Preliminary results of this research appeared in [26].

are now maturing. CBMIR applications in wireless networks can, and must, adopt any such developments. Additionally, for the issue of reproduction, adequate techniques for preserving intellectual property do exist [23].[1]

The searching procedure can benefit from the latest approaches for CBMIR in wired P2P networks (see Section II-B). Nevertheless, the combination of the characteristics of the wireless medium and of the audio-music data pose new and challenging requirements, which call for new solutions:

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, since Mobile Hosts (MHs, the terms MH and peer are similar in this context and thus interchangeable) are moving and become in and out of range of the others continuously. One impact of this mobility is that selective propagation of the query among MHs, e.g., by using data indexing like DHT[2] as proposed by [55] or caching past queries ([24] for text documents and [28] 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. Thus, new query propagation methods need be developed for wireless ad-hoc networks.
2) The need to reduce traffic, which results from the size of audio-music data (approx. 8MBytes for a 3 minute query). This is done by replacing the original query with a newly developed representation that utilizes 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, where the communication ability is usually assumed to be around 1 MBps for relatively long distances (see also Section II-C). It is worth noticing, that the reduction of traffic also reduces the involvement of other MHs, due to constraints in their processing power and autonomy.
3) In CBMIR over wired P2P networks, should a matching music excerpt be 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, thus requiring optimization.

Existing methodologies in MANETs, address the aforementioned issues in a limited extent. In particular, algorithms proposed for the problem of routing in MANETs 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. These peculiarities include the need to search for similarity in a search space where similar content location is not known beforehand. As for the data, their size prohibits the dissemination for direct local retrieval. To our best knowledge, no existing approach has addressed all the aforementioned issues, collectively, in the field of MANETs.

[1]The searching procedure, i.e., the subject of this work, does not reveal concerns about legal issues, since it only involves excerpts.

[2]Distributed Hash Tables. In such systems each node is assigned with a region in a virtual address space, while each shared document is associated with a value (id) of this address space.

### C. *Contribution and paper organization*

This work introduces the application of CBMIR in wireless ad-hoc networks and investigates the challenges posed by the wireless medium in order to perform content-based MIR, as described in the paradigm of Section I-A. Accordingly, we propose novel techniques, which attain a significant reduction in both response time and network traffic, compared to naive approaches.

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 neighboring MHs (obtained by probing neighborhood 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 broadcasts (explained in Section II-C).
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 MH as well.
3) The additional traffic produced by the third requirement is addressed by a twofold proposal: (i) We propose policies to constraint the number of MHs involved for the propagation of the answers, by exploiting any MHs that were involved during the propagation of the query. (ii) We allow such MHs 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: i) the introduction of the problem and the identification of the inherent requirements, ii) a novel algorithm that combines the aforesaid techniques and addresses the posed requirements, and iii) an extensive experimental evaluation, which illustrate the efficiency of the proposed methodology.

The rest of the paper is organized as follows. Section II describes background and related work. In Section III we outline the proposed method, whereas Section IV describes features selection and the indexing method that we use. Section V provides a complete account of the searching algorithms and, subsequently, Section VI describes the proposed routing policies. Section VII presents and discusses the experimentation and results obtained. Finally, the paper is concluded in Section VIII.

## II. Background and related work

In this section, CBMIR is considered as a discrete procedure as well as in terms of a process deployed in P2P networks.

An overview of content-based music information retrieval systems, both for audio and for symbolic music notation can be found in [53]. Moreover, a brief introduction on information discovery and resource in wireless mobile ad-hoc networks is provided.

### *A. CBMIR in audio*

Music exists in two representations: the symbolic representation (MIDI, Humdrum, common music notation) and the acoustic representation (audio format - wav, mp3, etc). Their key difference lies in the fact that the family of symbolic representations contains information of what a musical player should perform, whereas the acoustic representations comprise a specific recorded performance of a music piece. Acoustic representations contain the sampled waveforms of a sound [46] while symbolic contain various degrees of structured descriptive music. In this work our focus is on acoustic musical data.

Our focus on acoustic music is motivated by the popularity of music in acoustic format. A reason of that popularity is the ease of quality performance reproduction that acoustic formats offer. In addition, this trend gives as well as receives further impulse by the transition of the music distribution model, which nowadays offers music to download as well. Thus, the obvious result is the formation of digitized music databases the size of which is rapidly augmenting. As users attempt information retrieval in these collections, methods for Music Information Retrieval (MIR) are necessary. Although abundantly used, even nowadays, the traditional metadata [49] (title, composer, performer, genre, date, etc.) of a music piece give rather minimal information about the actual content of the music object itself. Their use aims solely in performing MIR using textual information of the music pieces. On the other hand, MIR can be performed based on humming [22] (using a microphone) or on a small piece of musical file. This type of queries lies within the Content-Based MIR. In CBMIR, an actual music piece is required in order to compare its content with the content of the music pieces already available in the database.

Though, acoustic sequences tend to be very large in size as a three minute CD-quality recording can be about 30 MBytes.[3] Thus, for an approach to be efficient, characteristic features need be extracted from the music file in order to perform similarity search on them. As a first approach for feature extraction, one can transform the acoustic data into symbolic, leading to a complete account of the datum (transcription), and accordingly extract features. Although skilled musicians are able to perform music transcription with high success [32], computer music transcription is generally admitted to be very hard and poor performing [44], [60]. For this reason, a second approach for feature extraction is to compute approximations of some of the four most semantically important features such as pitch, rhythm, timbre and dynamics.

Common alternatives, in the direction of approximation of some of the four most semantically important features include pitch detection [6], [13] and rhythm detection [40], [42], [51]. Pitch detection is dealt with time-domain fundamental period pitch detection, autocorrelation pitch detection, adaptive filter pitch detection, cepstrum analysis and frequency-domain pitch detection. It should be noted that no pitch detection algorithm is totally accurate and some that appear to be, utilize music inputs that follow specific constraints or show increased computational requirements (non real-time). Key difficulties in pitch detection include attack transients, low and high frequencies identification, myopic pitch tracking and acoustical ambience. Rhythm detection, on the other hand, can be divided into three levels: low-level (event detection), mid-level (transcription into notation) and high-level (style analysis). As with pitch detection, rhythm detection is also inherently difficult due to non accurate human performance of musical scores as well as the ambiguity of the music notation.

### *B. CBMIR in P2P networks*

Research related to the application of CBMIR in wired P2P networks is recent. In one of the first attempts, [56] presents four P2P models for CBMIR, which include centralized, decentralized and hybrid categories. Another research based on a hybrid configuration is presented in [55], in which the authors propose a DHT-based system utilizing both manually specified attributes (artist, album, title, etc.) and extracted features in order to describe the musical content of a piece. The authors of [61] propose the utilization of the feature selection and extraction process that is described in [60] for CBMIR in a decentralized unstructured P2P system. Moreover, although oriented towards a differentiated discipline, the work of [50] refers to audio retrieval in P2P networks. The principal target of this research is combating of unauthorized music file sharing in P2P networks. Finally, [28] investigated the problem of content-based searching for similar acoustic data over unstructured decentralised P2P networks, under the time-warping distance[4].

In the present work we deal with a wireless ad-hoc network, where two nodes can communicate only if in close proximity (in-range). As described, in this kind 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.

### *C. Information discovery/provision in wireless mobile ad-hoc networks*

As was previously mentioned, a MANET is a collection of wireless MHs forming a temporary network without the aid of any centralized administration or standard support services regularly available on the wide area network to which the hosts may normally be connected.

Ad-hoc networks are significantly different than Wireless Local Area Networks (WLANs), which are infrastructured, and Wireless Personal Area Networks (WPANs), e.g., Bluetooth, which are very short range wireless networks (with a

[3] [18] reports that MP3 compression of bitrates above 128Kbps is “near CD quality”. In this case, the recording requires 5-8 MBytes, which is still large.

[4] Dynamic Time Warping (DTW) has been proposed as a more robust similarity measure to Euclidean distance, as it can express similarity between two time series even if they are out of phase in the time axis or they do not have the same length.

range around to 10 meters. Although, different than WLANs and WPANs, MANETs are often implemented using WLANs or WPANs [62]. Thus, the medium access control layer of the ad-hoc networks is commonly assumed to be than of WLANs or WPANs, providing, for instance, symbol rates at the range of 11 and up to 50 Mbps. Though, these rates are achievable for ranges less than 70 meters; for ranges between 110–130 meters the rate is 1 Mbps, whereas for distances longer than 100 meters, the rates drop below 1 Mbps (Figure 2.2 in [62] and Table 3.6 in [2]). For this reason, almost all the studies involving transmissions at a range of 250 meters or longer, assume a symbol rate between 500 Kbps and 1 Mbps.

In an ad-hoc network, 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. 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. Routing algorithms for MANETs are radically different from the traditional routing (e.g., Open Shortest Path First) and information search protocols (e.g., Distributed Hash Table) used in hardwired networks, due to the absence of "fixed" infrastructure (servers, access points, routers and cables) in a MANET as well as the mobility of the nodes. For wireless ad-hoc networks there have been proposed various routing/discovery protocols, which roughly fall into the following categories ([1]): a) table-driven or proactive routing protocols, b) source-initiated on-demand or reactive routing protocols, and c) hybrid routing protocols.

Proactive protocols maintain unicast routes between all pairs of nodes regardless of whether all routes are actually used. Therefore, they require consistent, up-to-date routing information from each node to every other node in the network and thus are practically unfeasible for large-scale and dynamic MANETs.

On the other hand, the main idea in on-demand (reactive) routing is to find and maintain only needed routes. Recall that proactive routing protocols maintain all routes without regard to their ultimate use. The obvious advantage with discovering routes on-demand is to avoid incurring the cost of maintaining routes that are not used. This approach is attractive when the network traffic is bursty and directed mostly toward a small subset of nodes. The most popular on-demand routing protocols are the Dynamic Source Routing (DSR) and ad-hoc On-demand Distance Vector (AODV).

DSR [20] is characterized by the use of source routing. That is, the sender knows the complete hop-by-hop route to the destination. These routes are stored in a route cache. The data packets carry the source route in the packet header. When a node in the ad-hoc network attempts to send a data packet to a destination for which it does not already know the route, it uses a route discovery process to dynamically determine such a route. Route discovery works by flooding the network with route request (also called query) packets.

AODV [43] shares DSR's on-demand characteristics in that it also discovers routes on an "as needed" basis via a similar route discovery process. However, AODV adopts a very different mechanism to maintain routing information. It uses traditional routing tables, one entry per destination. This is in contrast to DSR, which can maintain multiple route cache entries for each destination. Without source routing, AODV relies on routing table entries to propagate a RREP back to the source and, subsequently, to route data packets to the destination. AODV uses *destination sequence numbers* to prevent routing loops and to determine freshness of routing information.

On-demand and hybrid routing protocols rely on some form of *broadcasting*; broadcasting is best suited in cases where information packets are transmitted to multiple hosts in the network. *Flooding* is the simplest broadcasting approach, where every node in the network forwards the packet exactly once; flooding ensures full coverage of the MANET provided that there are no network partitions. Flooding, though, generates too many redundant transmissions, causing the *broadcast storm problem* [39].

Various algorithms have been proposed to address this problem [35], [30]. These algorithms 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 former methods do not guarantee full coverage of the network, whereas the latter do provide coverage guarantees, and thus they are preferable.

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. All the categories of the deterministic algorithms, apart from the *local algorithms*, require (full or partial) global state information, thus they are impractical. The local or *neighbor-designating* algorithms maintain some local state information, i.e., 1-hop neighborhood information by periodic exchange of 'HELLO' messages, which is feasible and not costly. In the neighbor-designating methods, the forwarding status of each node is determined by its neighbors. As a matter of fact, the source node selects a subset of its 1-hop neighbors as forward nodes to cover its 2-hop neighbors. This forward node list is piggybacked in the broadcast packet. Each forward node in turn designates its own forward node list.

Remotely related to the topic of this paper is the issue of multicasting streaming media (audio/video) to MANETs (e.g., [16]) or unicasting audio to 3G UMTS devices [47]. These works though assume the existence of a central server/supplier, which provisions the mobile clients with multimedia data.

## III. Outline of the searching procedure

In this work, music similarity is used in order to identify similar musical pieces to a query musical piece, in a network of mobile hosts (as described in Section I-A). The problem of finding similar music sequences in a MANET requires a searching procedure, which will detect MHs in the MANET that have similar sequences, find those sequences in the MHs, and return them back to the querier. The already described requirements of the wireless framework formulate the examined searching procedure in the following way:

i) There is no prior knowledge of the data MHs store, that is the querier has no knowledge of the location of the required data.
ii) MHs that have qualifying sequences have to be reached in a way that addresses their mobility and minimizes traffic. Due to their relative positions and the preferred tolerance to traffic (see below), all such nodes may not be possible to reach.
iii) At each reached MH, the qualifying sequences have to be detected by detaining the MHs, in terms of CPU cost, as little as possible.
iv) 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 following paths different from those through which the MHs with qualifying sequences were reached, since intermediate MHs may have changed their position, and therefore be out of range. Due to this, every detected answer may not be possible to reach the querier.

An example is illustrated in Figure 1. The querier is MH $P_1$. During the forward phase (Figure 1a), the query is received by MHs $P_2$ and $P_3$. During the backward phase (Figure 1b), 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 $P_4$ (previously out the range of $P_1$).

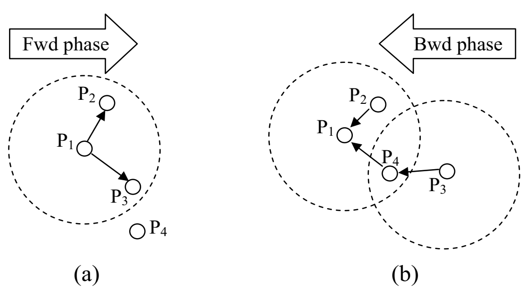



Fig. 1. An example of the searching procedure.

The searching procedure is initiated at the querying MH, aiming at detecting sequences in other MHs, which contain sequences whose similarity from the query sequence $Q$ is within user-defined boundaries, a threshold $\epsilon$. The definition of the distance measure is detailed in Section IV. Just for now, we can intuitively think of the distance as a measure of how dissimilar two music sequences are. The length of detected sequences is equal to the length of the query sequence $Q$.

To address traffic minimization, $Q$ has to be transformed to a representation form, denoted as $R$, through which qualifying sequences are detected.

Due to this transformation, it is possible that false-positive results may appear. A false positive result is a result that appears to be a true result when comparing with the transformed representation, though, under the non-transformed query is not a real result. Moreover, $R$ must present no false-negatives (real results that were missed due to the transformation). However, its particular implementation determines whether false-positives may be produced or if they will be completely avoided. Based on all the aforementioned issues, an abstract scheme to describe the entire searching procedure consists of the steps depicted in Figure 2, which are also summarized in four events.

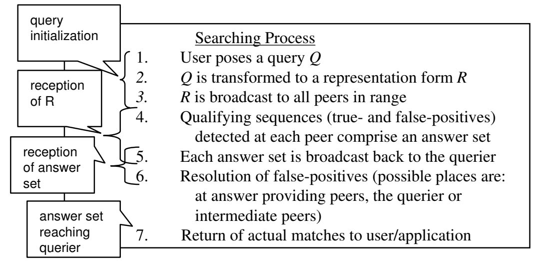



Fig. 2. Searching process and basic events.

To avoid duplicate effort, the procedure tags $R$ with an ID (see Section II-C). This way, MHs that have already received it will perform no further action. Additionally, the propagation of $R$ to the neighboring MHs is controlled by a parameter called $h$, which is a counter that is decreased at each receiving MH (denotes the available number of hops). Its initial value, at the querier, is equal to $MaxHop$. This value corresponds to the preferred tolerance to traffic and network reach/coverage. The propagation of answer sets (resulting from step 5) is handled similarly.

As already mentioned, the searching process consists of a forward and a backward phase. During the former, $R$ is propagated and during the latter answers are routed back to the querier. The two phases are interleaved, since during the propagation of $R$ by some MHs, other MHs are returning answers to the querier. The backward phase's volume mainly depends on the existence of answers and the number of false-positives, while the forward phase depends on the size of $R$, our coverage willingness as well as the network reachability. In general, the volume of information transferred during the backward phase is larger than that of the forward phase.

Having outlined the searching procedure, in the following sections we detail its parts. First we elaborate on the features that can be selected for the formation of $R$. Next, we describe the acceleration of similarity searching within each MH by using indexing. Based on these, we next describe two searching algorithms, which follow different choices with respect to the formation of $R$. Finally, we present methods to improve the backward phase.

## IV. Features and Indexing

### A. Features for CBMIR

One of the main challenges in MIR is the choice of representation of the musical information within the system. As the sequences of acoustic music objects tend to be quite large in size, they are commonly described by a set of features. Numerous standpoints exist on what features to retain and on how to select these features [5], [37], [38], [41], [57], [58]. The selection of appropriate features is considered very important in music information retrieval [19]. Meaningful features help in the effective representation of the objects and enable the use of indexing schemes for efficient query processing.

Ordinary features for symbolic music can be the any of the key characteristics of music (such as pitch, rhythm and timbre [7]) or even structural patterns [9], as used in [25]. The most typically encountered features for the acoustic representation are produced by time analysis, spectral analysis and wavelet analysis.

In this work, we do not concentrate on devising new features. Instead, we are interested in a methodology for the searching procedure. Our methodology is able to embrace any high performance feature extraction procedure. Accordingly, 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, as shown by [8], [33], [34].

The wavelet transform has long been used in image and signal processing while its use in information retrieval and data mining has been extensive [36]. A complete survey on wavelet application in data mining can be found in [33]. In general terms, the wavelet transform is a tool that provides quality time and frequency resolution, while dividing up data, functions, or operators into different frequency components and then studying each component with a resolution matched to its scale[33], [14], [21].

Wavelets present numerous favorable properties in contrast to other type of analyses. Among them, lie the efficient computation complexity, the vanishing moments that support denoising and dimensionality reduction while focusing on most important information, the compact support that guarantees the localization of the wavelet, the de-correlated coefficients that enable the reduction of complex processes of time domain into simpler in the wavelet domain and the support to the Parseval's theorem. In addition, wavelets present a multiresolution property that leads to hierarchical representations and manipulations of the objects treated.

The previously mentioned merits of the wavelet transform, corroborate the use of wavelets on music. The low computation complexity assists the already burdened process by the large size of the musical data. The vanishing moments and their denoising capability cope with the noise introduced in musical recordings by the ambient sounds, during recording. The compact support allows locally altered musical pieces to retain their overall similarity, while the multiresolution adheres to the perception model of the ear, according to which the perception of both large scale quantities and small scale events, rely upon the multiresolution capability of the ear [8].

More particularly, we consider the Haar wavelet transformation for its simple incremental computation, its capability concerning the capture of time dependant properties of data and overall multiresolution representation of signals [12] as well as for the incorporation of the previously mentioned properties. However, our approach can easily be extended to other types of wavelet transforms.

### B. *Indexing within peers*

To facilitate the searching within peers we use the following approach. In a peer, each original audio sequence is transformed to a number of multidimensional points. We use a sliding window of length $n$ over the sequence and apply Discrete Wavelet Transform (DWT) to the contents of each window, producing $n$ coefficients per window. An example is depicted in Figure 3a. Therefore, each audio sequence produces a set of $n$-dimensional points in the feature space. Since $n$ depends on the query length and, thus, takes relatively large values (e.g., 64 K), in order to efficiently index them in the feature space, we select only the first $d$ dimensions from each point (in our experiments we used $d = 64$). This procedure dramatically reduces both the size of the index and the number of dimensions without affecting much the quality of the index. The reason for the latter is the merit of DWT to concentrate the energy of the sequence in the first few coefficients. However, false-positives are possible and thus require resolution.

Most importantly, it has been proven by [11] that no false dismissals are introduced when using only the $d$ first coefficients (due to Parseval's theorem). Notice that this property is proven in [11] for the Euclidean distance. Although this distance measure is simple, it is known to have several advantages, as it has been illustrated by [31]. 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.

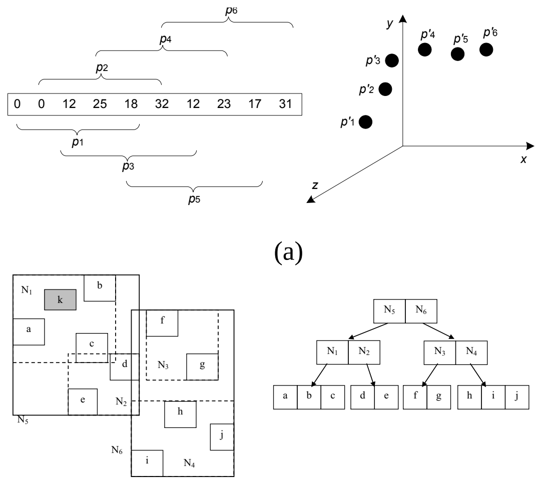
To speed-up the retrieval, for each sequence the collection of the resulting $d$-dimensional points is organized in Minimum Bounding Rectangles (MBRs), which are, then, stored in an R$^*$-tree [4]. Answering to query, the root is initially retrieved and its entries that intersect the query are only further examined recursively until reaching a leaf. All non intersecting nodes are not included in the search. An example is given in Figure 3b. Therefore, when searching for similar subsequences, we first retrieve candidates from the R$^*$-tree. We rank the candidates so as to process the most promising ones first (we observed that this saves a lot of CPU time) and then, those candidates are examined against the provided query representation. When the latter is reduced (as in the case of transcoding that will be explained), false-positives are still possible. Nevertheless, their number is significantly reduced. More details about indexing can be found in [27].

## V. Searching algorithms

In this section we describe the two algorithms that implement the searching procedure. The first is based on simple choices concerning the representation $R$ of the query sequence and its propagation during the forward and backward phases. The second (proposed) is based on more advanced choices with respect to the latter issues.

### A. *Algorithm based on maximal query representation*

A simplistic approach for the representation $R$ is to set it identical to the query sequence. The advantage is that no false-positives occur, since when a possible match has been found by




(a)

(b)

Fig. 3. Feature extraction process.

index probing, it can be immediately tested against the query itself (i.e., $R$). Thus, no false-positives will be included in the answer-sets, which could negatively impact the backward-phase traffic, as they would be propagated to the querier just to find that they are not actual matches. We have to note that, to be able to perform index probing (i.e., to avoid sequential searching at each MH), a small number of DWT coefficients are included in $R$ as well. However, their size is negligible compared to the size of the query sequence.

The resulting algorithm is denoted as ML (full *M*aximum representation with *L*ocal resolution at MHs). ML is summarized in Figure 4 according to the actions performed for each occurring event (see Figure 2).

- **Query initialization** The querier assigns to $R$ the entire query sequence (plus the few query coefficients) and propagates (broadcasts) it to all its neighbors.
- **Reception of** $R$ Upon the reception of $R$, each MH $P$ 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 neighbors of $P$ (backward phase). Accordingly, should there be available $h$, $R$ is conveyed to all $P$'s neighboring MHs (forward phase).
- **Reception of an answer-set** Each MH $P$, that is not the querier, receiving an answer-set, continues the propagation (backward phase) to *all* its neighboring MHs as long as there is available $h$.
- **An answer-set reaches the querier** When an answer-set reaches the querier, then the results are immediately presented to the user.

Fig. 4. The ML algorithm.

Although ML manages to control the traffic during the backward phase (due to the elimination of false-positives), this comes at the cost of excessive traffic during the forward phase. This is due to the representation $R$ that is propagated during the forward phase, which is equal to the entire query. For large query sequences this causes prohibitive forward traffic. Evidently, there emerges a trade-off between the two contrasting phases. What is, therefore, needed is a method that will balance the traffic between the two phases, aiming at overall improvement.

Another issue on which ML makes a simplistic choice is the selection of the neighboring MHs to which the answer-set is propagated during the backward phase. When handling the second and third events, ML selects all neighbors for this purpose, thus resorting to plain flooding. This simplistic selection can significantly impact the backward traffic. To overcome the problem we need to devise policies for the selective routing of the answer-sets. That is, we want to select only those nodes that are more promising to satisfy the receipt of the answers, thus significantly reducing backward traffic without reducing the chances of the answer-sets to reach the querier.

### *B. Algorithm based on reduced query representation and transcoding*

In Section V-A it was made clear that there is a tradeoff between the forward and backward traffic. ML focuses only on the improvement of backward traffic and incurs high forward traffic. In this section we present a new algorithm, which has a two-fold objective. The first is to produce a representation $R$ that achieves a balance between the two phases and minimizes the overall traffic. The second is to develop selective routing policies for the propagation of the answer-sets, leading to significant reduction of the backward traffic.

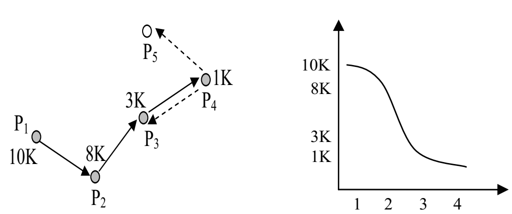
The first objective is confronted by setting $R$ between the two extremes cases: (i) the minimum possible representation with only the $d$ DWT coefficients that are required for the local index-searching (minimizing forward traffic), and (ii) the maximum possible representation with all $n$ elements in the query sequence itself (eliminating the burden of false-positives in terms of computation and backward traffic). Therefore, between the two extremes, $R$ can consist of the $l$ greater DWT coefficients, where $d \leq l \leq n$. Notice that this type of representation generalizes the two extreme cases: by setting $l = d$, $R$ becomes identical to the first (i) case; in contrast, by setting $l = n$, $R$ becomes identical to the second (ii) case, because the $n$ DWT coefficients are equivalent to the $n$ elements of the query sequence (due to the Parseval's theorem).[5] As described in Section IV, a number $l$ of the greater DWT coefficients can effectively capture the energy of the music sequence and reduce the number of false-positives. The result is that, compared to the second (ii) case, the forward traffic is expected to be smaller, because $l \leq n$. Compared to the first (i) case, the backward traffic is expected to be smaller too, due to the number of false-positives being significantly reduced, since $d \leq l$.

[5]In the case of ML we could have $R$ to consist of all the $n$ DWT coefficients. However, we choose the $n$ sequence elements in the time domain just to avoid the computation of the inverse DWT, since in our case the time domain presents a smaller storage requirement as the data values are in range 0-255.

The tuning of $l$, however, is difficult, because it depends on several factors, like the topology of the MANET, which are changeable. For this reason we follow a different approach. Initially, $l$ is assigned a large value (see Section VII for its tuning) and this value is monotonically reduced during the propagation of $R$ in the forward phase. This technique can be though of as a *transcoding* scheme, as it involves sequences with varying number of DWT coefficients that correspond to varying approximations of the initial query sequence. The transcoding scheme:

- Keeps forward traffic low, as the size of $R$ is reducing at each stage of the forward phase propagation.
- Reduces backward traffic by letting the MHs involved in the forward phase to cache the transcoded representation and, during the backward phase, to use it for early resolving false-positives, before they reach the querier. The problem of caching depends on several network parameters. This problem is independent to our approach, while effective solutions can be found in [17], [29]. In our experiments, we found that by simply caching the representations for a small, fixed amount of time, adequate performance is attained.
- Reduces the processing (CPU) time at each MH, as the cost of resolving false-positives at each MH depends on the size of $R$.

The reduction is performed by getting $l$ values according to an inverse sigmoid function (Figure 5b). Due to the shape of this function, the immediate neighborhood of the querier, which can provide results faster, receives a larger $R$, whereas the burden posed on MHs that are far is appreciably smaller. Also, this way we control the exponential growth of traffic that results by plain broadcasting of full-size representation. An example is depicted in Figure 5a. $P_1$ is the querier and $P_4$ is the node that starts propagating the answer-set. The MHs in the path from $P_1$ to $P_4$ are depicted gray shaded, and they are annotated with the size of $R$ that reaches them ($P_1$ starts with 10 K DWT coefficients). Figure 5b illustrates that these sizes are reducing, following an inverse sigmoid function. During the backward phase, starting from $P_4$, MHs $P_3$ and $P_5$ can be reached (depicted with dashed arrows). The cached representation in $P_3$ can help to resolve possible false-positives in the answer-set. The reason is that in $P_4$ the false-positives were examined against a smaller $R$ than the one in $P_3$. In contrast, $P_5$ was not in the path, thus cannot resolve any false-positives.




(a) (b)

Fig. 5. An example of the searching procedure.

Henceforth, the size of the initial query representation is given as a factor (denoted as $I$) of the complete query size, whereas the slope of the inverse sigmoid function is controlled by a parameter denoted as $\alpha$ (higher values of $\alpha$ produce a steeper slope).

Regarding the second objective, we do not follow the simplistic approach of ML, which propagates the answer-sets to all neighbors. In contrast, during the forward phase, as it is typical in any dynamic source routing protocol [20], each MH that receives $R$, additionally receives the ID of all MHs that were used in the path from the querier to it. These IDs can be maintained along with $R$ with minimal cost (only some bytes). When a MH starts propagating answer-sets, it selects among its current neighbors those that will propagate the answer-set (not all of them). To make this selection, it applies a policy that focuses on the neighbors that were included in the path from the querier to it. Since several such policies can be developed, in the next section we elaborate further on them. All the policies, despite their differences, emphasize on selecting neighboring MHs that were included in the path. The reason is that the cached representations that these nodes maintain, can resolve false-positives during the backward phase. Therefore, traffic is substantially reduced. More details will be given in Section VI.

The algorithm that combines all the aforementioned characteristics is denoted as RT (querying by *R*educed representation with *T*ranscoding) and is illustrated in Figure 6. The handling of success or failure is treated similarly to standard routing MANET protocols employing a TTL-like policy [59].

- **Query initialization** The querier sets $R$ equal to a sample with an initial size (parameter) plus the query coefficients, and propagates (broadcasts) it to all its neighbors.
- **Reception of** $R$ Upon the reception of $R$, each MH $P$ 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 $h$, $R$'s size is reduced, and the reduced $R$ is conveyed to all $P$'s neighboring MHs (forward phase).
- **Reception of an answer-set** When a MH receives a reply, it checks if it can resolve any false-positives. This is true should it have 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 MH). After any possible pruning, as long as there is available $h$, the answer-set is routed backwards following a policy.
- **An answer-set reaches the querier** When an answer-set reaches the querier, initially any remaining false-positives require resolution, and then the results are presented to the user.

Fig. 6. The *RT* algorithm.

## VI. Routing policies for the backward phase

In this section we describe three policies for routing the answer-sets in the backward phase. The first two policies (global and local counter, described in [10]) are based on existing methods, whereas the third one (critical mass) is novel. As mentioned, all policies try to select nodes that were included in the path during the forward phase. Nevertheless, the backward phase cannot be based only on such nodes. Due to the mobility of MHs, it may be impossible to reach the querier unless other MHs (not included in the path) are additionally involved. The objective of all policies is to control the number of involved MHs so as to reduce backward traffic. These policies constitute a hybrid approach between probabilistic broadcasting, where the broadcasting decision is completely local to each mobile host and the deterministic broadcasting which relies on the discovery of some form of connected dominating set [35].

### A. Global and local counter policies

To clarify the description of the first two policies, consider the example of Figure 7a, which depicts the path from MH $P_1$ to MH $P_6$, which was followed in the forward phase. Figure 7b depicts the routing of the answer-set from $P_6$ back to $P_1$. Comparing the two phases, several MHs have changed their location, others have switched off, and some new ones have become reachable. The MHs that are depicted grey color are the ones that were included in the forward path too, whereas the rest are new ones that were involved only in the backward phase.

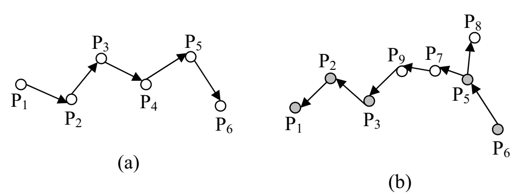



Fig. 7. An example of propagation in a MANET: a) forward phase, b) backward phase.

With the *global-counter* (GC) policy, when a MH ($P_6$ in the example) starts propagating an answer-set, it tags the answer-set with a maximum number of re-transmissions, $h$, equal to the length of the forward path plus an extra value $e$. In the example, the length (number of edges) is equal to 5. Let $e = 1$ and $h = 6$. GC tries to find among the neighbors, the one that was its predecessor in the path. In the example, at $P_6$ GC tries to find $P_5$. If this MH is reachable, then it is the only one selected to propagate the answer-set and $h$ is decreased by one. The same procedure is applied recursively. At $P_5$, GC tries to find $P_4$. If $P_4$ is not reachable, as it is now the case, then GC propagates the answer-set to all neighboring MHs (broadcasting to $P_7$ and $P_8$) and each of them receives a $h$ value decreased by one. Next, unless a MH in the forward path has been reached, GC continues by broadcasting to all neighbors. At each propagation of the answer-set, $h$ is decreased by one, thus actually acting as a decreasing global counter. If a MH from the forward path is reached at any point again, then, as previously, GC tries to find its predecessor. In the example, $P_3$ is a node from the path, which has been reached with $h$ equal to 2. Its predecessor is $P_2$, which then propagates (as $h$ is 1) the answer set to $P_1$ and the procedure terminates.

In summary, when selecting the MHs to route back the answer-set, GC tries to follow the MHs included in the forward path. However, to overcome problems from the alteration of the MANET (like the disappearance of $P_4$ in this example), it allows an amount of discrepancy by resorting to broadcasting. To control the discrepancy, and thus the backward traffic, it uses the value of $e$. Notice that with a very large $e$, GC resorts to broadcasting for a very large number of times, thus becoming equivalent to the simplistic policy used by the ML searching algorithm. In contrast, with a very small $e$, the querier may not become reachable, especially when the MANET changes very fast.

A variation of GC works as follows. After a discrepancy, when a MH from the path has been reached again, we reset $h$ to its initial value. In the previous example, when $P_3$ is reached again, available hop is reset to 6 (initial value). Thus, $h$ acts as a decreasing local counter, because it is reset independently at several MHs. For this reason this policy is denoted as *local-counter* (LC). Its objective is to increase the probability of reaching the querier, by rewarding the identification of the forward path. Nevertheless, this can increase the backward traffic.

### B. Critical mass policy

With the *critical-mass* (CM) policy, if at least a number, denoted as *critical-mass factor* (*CMF*), of the current neighbors was in the forward path, we select them as the only ones to propagate the answer-set. If their number is less than *CMF*, then we additionally select randomly some of the current neighbors (not in the path) in order to have at least *CMF* MHs to propagate the answer-set. In contrast, if their number is larger than *CMF*, then they are all selected. For example, consider the case in Figure 8. Figure 8a depicts the forward phase, whereas Figure 8b presents the backward case. As shown, during the backward phase some MHs have now relocated. Let *CMF* be 2. When $P_4$ starts propagating the answer-set, it first selects $P_3$, because it belongs to the forward path. Since this is the only such MH and *CMF* is 2, it also selects $P_5$ at random among the other reachable MHs.

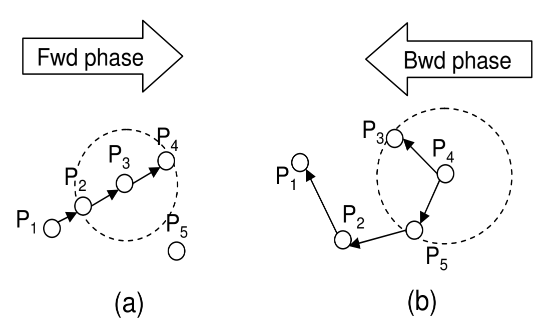



Fig. 8. Example of relative locations of MHs in forward and backward phase.

The nodes that were selected at random in order to fulfill *CMF*, are still provided with the path of the MH that initiated the propagation of the answer-set (for the previous example, $P_5$ that is selected by $P_4$, will also 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 neighbors that appear in the forwarded path (in the same example, $P_5$ finds $P_2$ that was in the path). Therefore, the impact of such randomly selected MHs on the proposed policy may be kept at a moderate level.

The CM policy differs from GC and LC in the following aspects: (i) It does not search predecessors in the path, as it focuses on identifying MHs that were in the forward path, regardless of their order (i.e. not searching for the previous node). This makes CM more flexible to the changes in the MANET. (ii) It never resorts to broadcasting to all neighbors. At worst case, the number of randomly selected MHs is equal to *CMF*. This attains better control of the backward traffic. Due to the aforementioned characteristics, CM is expected to outperform GC and LC, as will be shown in Section VII.

## VII. Performance Evaluation

### A. Simulation configuration

In this section, we provide an experimental comparison of the three described content-based audio retrieval algorithms. The performance of the algorithms was compared through simulation. The settings of the simulation were as follows. The mobile ad-hoc 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 MHs (default value equals to four). The aforementioned settings correspond to a realistic scenario for a MANET, like the one described in the Introduction. Accordingly, the average number of sequences per node was 12, a quantity that is quite reasonable for the state-of-the-art MP3 cell phones [48] and PDAs, both of which support the latest memory cards.

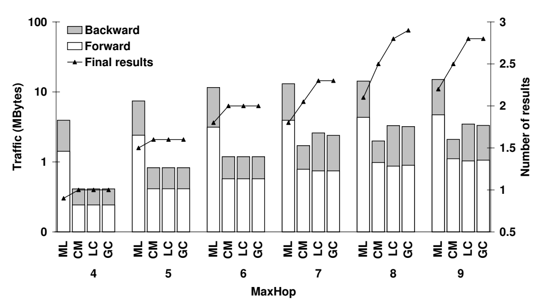
Regarding the simulation of mobility, we based our experiments on the GSTD simulator, as presented in [52], which considers hosts moving freely in a 2-D area. We used a squared area with side equal to 4,000 m, whereas the transmission/reception range of each MH was set to 500 m radius. Different degrees of velocity were selected for the moving MHs, 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). Additionally, to account for the fact that mobile devices may enter doze mode (power-safe status where the device is out of network), we take each time for the MHs a doze-mode probability, with default value equal to 0.1 (that is, at each time unit a MH is out of network with probability 0.1).

Regarding CM, the default value for *CMF* was 10% of the number of neighbors at each MH, whereas the default initial sample size was 10% of the query sequence's size. For GC and LC policies, the additional value $e$ added to MaxHop is set to 2 (we tried other similar values with no significant improvement). For all algorithms, the default value of $\epsilon$ was 0, and the default MaxHop was set to 5. Henceforth, when parameter values are not specified, we assume the default values.

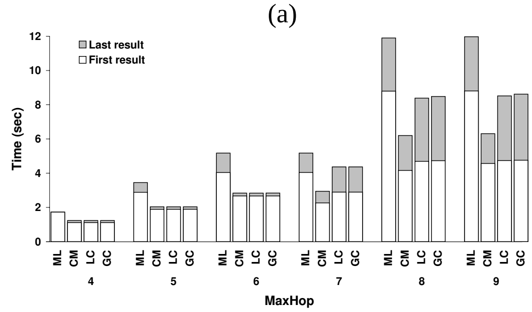
The evaluation metrics are the average traffic (measured in MBytes) that each query incurs, the number of results obtained and the time the first and last result were discovered (the time of the first result is a useful measure, since users may terminate searching early). The results on time reflect 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.

### B. Experimental results

In our first experiment, we examined the traffic against MaxHop. The results are illustrated in Figure 9a. The forward and backward traffics are depicted separately, whereas their addition (height of bars) gives the total traffic. As expected, ML produces the highest forward traffic in all cases (due to maximal query representation), whereas the forward traffic of CM, LC and GC are about the same. Regarding backward traffic, as described, ML attains a decreased number of returning results. However, due to the absence of an efficient backward routing policy, this advantage is invalidated. The rest approaches, considerably improve backward traffic, with CM performing better for MaxHop greater than seven. 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 CM, LC and GC, the requirement of optimizing the forward phase, is fair. Additionally notice that the number of results (depicted in Figure 9a with a solid line) obtained by ML are less than the results obtained by CM, LC and GC and although the difference is small, there is a clear trend.




(a)

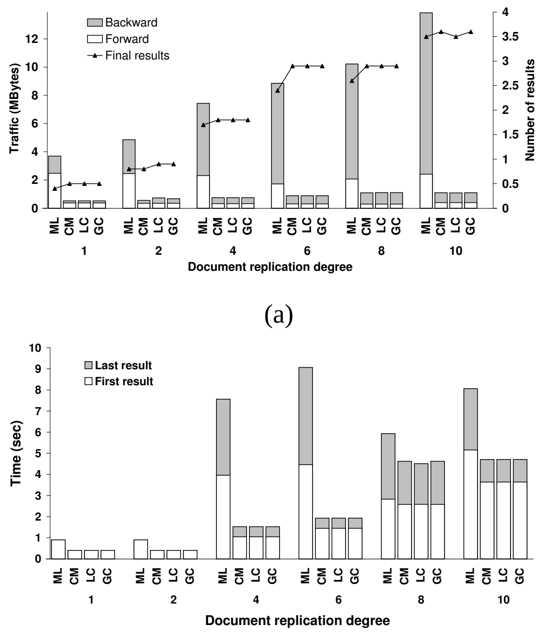



(b)

Fig. 9. Traffic, number and time of results vs. MaxHop

This result can be further clarified by the results on time of the first and last results, which are depicted in Figure 9b. In this figure, the height of the bars correspond to the time of last result, whereas the time of first result is depicted separately as its fraction. As expected, increase in available MaxHop produces longer times, since more MHs are examined. In all cases, the increase in time is far more steep for ML, while CM presents an advantage over LC and GC.

Next, we examined the impact of the document replication degree on the traffic and the time of the first and last result. The former experiment (given in Figure 10a) shows that an increase in the replication degree has a clear impact on both the number of results and the backward traffic. This is especially true for ML. Once again, the number of results obtained by ML are slightly less than for the other. Regarding times of first and last result (Figure 10b), an analogous behavior is observed.




(a)

(b)

Fig. 10. Traffic, number and time of results vs. document replication degree

The following experiment considered the traffic produced by the MH doze-mode probability (Figure 11). It is quite clear that for increased values of probability, the network becomes less connected, thus leading in decrease of the results returning to the querier. ML is clearly outperformed, whereas the others perform about the same. What is more, the increase in doze-mode probability leads to a decrease in traffic, since the diminished connectivity of the graph prohibits both the discovery of results and the propagation of any found.

Next, we examine the impact of query range $\epsilon$. Figure 12 shows the results for traffic with respect to $\epsilon$. Since CM, LC, and GC perform similarly, to improve clarity we only include the results for the former. As $\epsilon$ increases, more results are found and, thus, backward traffic increases too (forward traffic is unaffected). However, the increase is much more obvious for ML, whereas CM, due the effectiveness of the policy for the backward phase, has a very smooth increase.

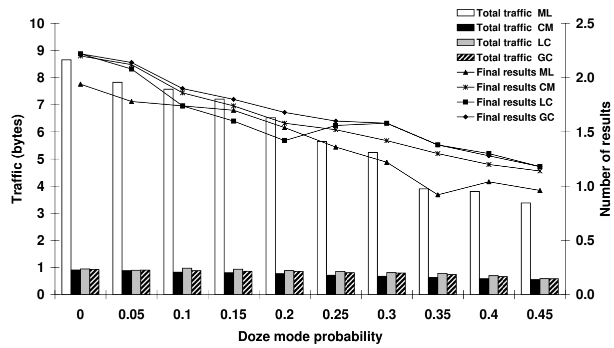



Fig. 11. Traffic and number of results vs. doze mode probability

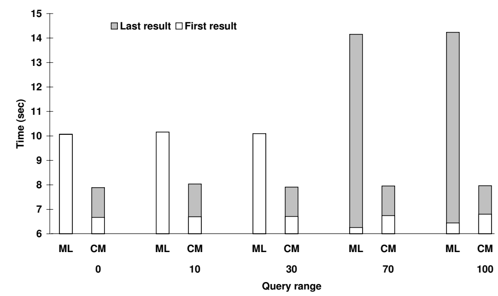


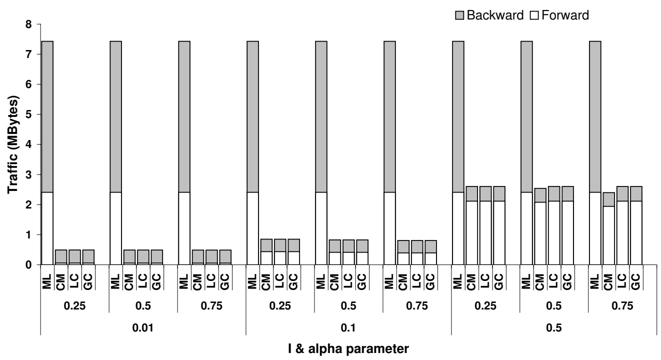

Fig. 12. Traffic vs. query range ($\epsilon$)

We also tested the sensitivity of the CM algorithm against $I$ and $\alpha$ parameters (as formally described in Section V-B). The results are illustrated in Figure 13a. For both $I$ and $\alpha$ parameters, the performance of ML remains totally unaffected and is only included for comparison purposes. As far as the $I$ parameter is concerned, backward traffic is unaffected. As expected, forward traffic increases with increasing sample size. For the examined range of values, the reduction in traffic is not combined with a change in the number of found matches, which are similar in the order of decimal values, and thus omitted. In contrast, the examined values for $\alpha$ parameter resulted to small differences between the approaches.

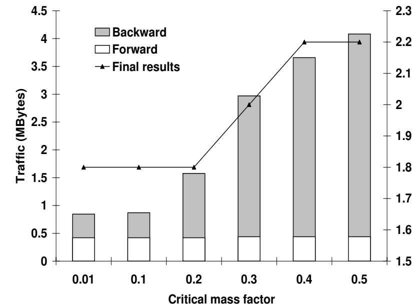
Finally, we examined the sensitivity of CM against *CMF* (the others are not affected by *CMF*). The traffic and number of results of CM, for varying *CMF* values, are depicted in Figure 13b. When *CMF* is high, the effectiveness of the policy for the backward phase is limited, since most MHs are selected at random by this policy. Thus, the resulting backward traffic is high (forward traffic is not affected). Notice that, for the examined range of *CMF* values, the reduction in traffic slightly affects the number of found matches (the difference between the results for the extreme values of *CMF* are only in the order of decimal values). On the other hand, the increase in the results comes at the cost of higher traffic. Conclusively, relatively small *CMF* values are sufficient.

## VIII. Conclusions

In this paper, we introduce the application of CBMIR application in wireless ad-hoc networks. We recognize the new challenges posed by this type of networks. To address them, we propose a novel algorithm, which is based on a twofold optimization: (i) the use of query representation with




(a)




(b)

Fig. 13. Traffic vs. I & $\alpha$ parameters (a) and Traffic & number of results vs. CMF (b)

reducing length, (ii) a selective policies 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 suitability of the proposed method.

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.

In future work, we plan to examine other features and to develop a real prototype with mobile devices. Additionally, we intend to extend the system so as to accommodate musical genre querying as well. The key idea is that based on a annotated querying feature set(such as the features described in [54]) the querier can identify similar genre audio data within a mobile ad-hoc network.

## References


[1] D. P. Agrawal and Q.-A. Zeng, *Introduction to Wireless and Mobile Systems*. Thomson, Brooks/Cole, 2003.

[2] G. Anastasi, M. Conti, and E. Gregori, "IEEE 802.11 adhoc networks: Protocols, performance and open issues," in *Mobile Ad Hoc Networking*, S. Basagni, M. Conti, S. Giordano, and I. Stojmenovic, Eds. IEEE Press & Wiley-Interscience, 2004, pp. 69–116.

[3] 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*, vol. 21, no. 10, pp. 1627–1641, 2003.

[4] N. Beckmann, H. P. Kriegel, and B. Seeger, "The R$^*$-tree: An efficient and robust method for points and rectangles," in *Proceedings of the ACM International Conference on Management of Data (SIGMOD)*, 1990, pp. 322–331.

[5] S. Bray and G. Tzanetakis, "Distributed audio feature extraction for music," in *Proceedings of the International Symposium on Music Information Retrieval (ISMIR)*, 2005, pp. 434–437.

[6] I. Bruno, S. L. Monni, and P. Nesi, "Automatic music transcription supporting different instruments," in *International Conference on WEB Delivering of Music*, 2003.

[7] D. Byrd and T. Crawford, "Problems of music information retrieval in the real world," *Information Processing & Management*, vol. 38, no. 2, pp. 249–272, 2002.

[8] A. P. C. Roads, S. Pope and G. D. Poli, Eds., *Musical Signal Processing*. Royal Swets & Zeitlinger, 1997.

[9] E. Cambouropoulos and P.-Y. Rolland, "Algorithmes et techniques pour l'analyse musicale," in *Informatique Musicale du Signal au Signe Musical*, F. Pachet and J.-P. Briot, Eds. Paris, France: Hermes Science Publications, 2004.

[10] R. Castaneda, S. R. Das, and M. K. Marina, "Query localization techniques for on-demand routing protocols in ad hoc networks," *ACM/Kluwer Wireless Networks*, vol. 8, no. 2/3, pp. 137–151, 2002.

[11] K. Chan and A.-C. Fu, "Efficient time series matching by wavelets," in *Proceedings of the IEEE International Conference on Data Engineering (ICDE)*, 1999, pp. 126–133.

[12] K.-P. Chan, A.-C. Fu, and C. Yu, "Haar wavelets for efficient similarity search of time-series: With and without time warping," *IEEE Transactions on Knowledge and Data Engineering*, vol. 15, no. 3, 2003.

[13] N. Collins, "Using a pitch detector for onset detection," in *Proceedings of the International Symposium on Music Information Retrieval (ISMIR)*, 2005, pp. 100–106.

[14] I. Daubechies, S. Mallat, and A. S. Willsky, "Introduction to the special issue on wavelet transforms and multiresolution signal analysis," *IEEE Transactions on Information Theory*, vol. 38, no. 2, pp. 529–532, 1992.

[15] J. DeVriendt, P. Laine, C. Lerouge, and X. Xu, "Mobile network evolution: A revolution on the move," *IEEE Communications magazine*, pp. 104–111, April 2002.

[16] A. Dutta, J. Chennikara, W. Chen, O. Altintas, and H. Schulzrinne, "Multicasting streaming media to mobile users," *IEEE Communications magazine*, pp. 81–89, October 2003.

[17] Y. Fang, Z. J. Haas, B. Liang, and Y.-B. Lin, "TTL prediction schemes and the effects of inter-update time distribution on wireless data access," *ACM/Kluwer Wireless Networks*, vol. 10, no. 5, pp. 607–619, 2004.

[18] Fraunhofer_IIS, "Audio & multimedia mpeg audio layer-3," retrievd from http://www.iis.fraunhofer.de/amm/techinf/layer3 on June 2006.

[19] M. Grimaldi, P. Cunningham, and A. Kokaram, "An evaluation of alternative feature selection strategies and ensemble techniques for classifying music," in *Proceedings of the Workshop on Multimedia Discovery and Mining*, 2003.

[20] D. Johnson and D. Maltz, "Dynamic source routing in ad hoc wireless networks," in *Mobile Computing*, T. Imielinski and H. Korth, Eds. Kluwer Academic Publishers, 1996, vol. 353, pp. 153–181.

[21] S. Kadambe and G. F. Boudreaux-Bartels, "Application of the wavelet transform for pitch detection of speech signals." *IEEE Transactions on Information Theory*, vol. 38, no. 2, pp. 917–924, 1992.

[22] T. Kageyama, K. Mochizuki, and Y. Takashima, "Melody retrieval with humming," in *Proceedings of the International Computer Music Conference*, 1993, pp. 349–351.

[23] 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*, pp. 961–970, 2004.

[24] V. Kalogeraki, D. Gunopulos, and D. Zeinalipour-Yazti, "A local search mechanism for peer-to-peer networks," in *Proceedings of the ACM International Conference on Information and Knowledge Management (CIKM)*, 2002, pp. 300–307.

[25] I. Karydis, A. Nanopoulos, and Y. Manolopoulos, "Symbolic musical genre classification based on repeating patterns," in *Proceedings of the ACM Workshop on Audio and Music Computing for Multimedia (AMCMM)*, 2006.

[26] I. Karydis, A. Nanopoulos, A. N. Papadopoulos, D. Katsaros, and Y. Manolopoulos, "Content-based music information retrieval in wireless ad-hoc networks," in *Proceedings of the International Symposium on Music Information Retrieval (ISMIR)*, 2005.

[27] I. Karydis, A. Nanopoulos, A. N. Papadopoulos, and Y. Manolopoulos, "Audio indexing for efficient music information retrieval," in *Proceed-*

*ings of the IEEE International Conference on Multi Media Modeling (MMM)*, 2005, pp. 22–29.

[28] ——, “Music retrieval in p2p networks under the warping distance,” in *Proceedings of the International Conference on Enterprise Information Systems (ICEIS)*, 2005.

[29] D. Katsaros and Y. Manolopoulos, “Web caching in broadcast mobile wireless environments,” *IEEE Internet Computing magazine*, vol. 8, no. 3, pp. 37–45, 2004.

[30] ——, “The geodesic broadcast scheme for wireless ad hoc networks,” in *Proceedings of the IEEE International Workshop on Mobile Distributed Computing (MDC)*, 2006, pp. 571–575.

[31] E. Keogh and S. Kasetty, “On the need for time series data mining benchmarks: A survey and empirical demonstration,” in *Proceedings of the ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD)*, 2002, pp. 102–111.

[32] A. Klapuri, “Signal processing methods for the automatic transcription of music,” Ph.D. dissertation, Tampere University of Technology, 2004.

[33] T. Li, Q. Li, S. Zhu, and M. Ogihara, “A survey on wavelet applications in data mining,” *SIGKDD Explorations*, vol. 4, no. 2, pp. 49–68, 2002.

[34] T. Li, M. Ogihara, and Q. Li, “A comparative study on content-based music genre classification,” in *Proceedings of the ACM International Conference on Research and Development in Information Retrieval (SIGIR)*, 2003, pp. 282–289.

[35] W. Lou and J. Wu, “Broadcasting in ad hoc networks using neighbor designating,” in *Handbook of Mobile Computing*, I. Maghoub and M. Ilyas, Eds. CRC Press, 2004.

[36] M. K. Mandal, T. Aboulnasr, and S. Panchanathan, “Fast wavelet histogram techniques for image indexing,” *Computer Vision and Image Understanding*, vol. 75, no. 1-2, pp. 99–110, 1999.

[37] D. McEnnis, C. McKay, I. Fujinaga, and P. Depalle, “Jaudio: A feature extraction library,” in *Proceedings of the International Symposium on Music Information Retrieval (ISMIR)*, 2005, pp. 600–603.

[38] M. Muller, F. Kurth, and M. Clausen, “Audio matching via chroma-based statistical features,” in *Proceedings of the International Symposium on Music Information Retrieval (ISMIR)*, 2005, pp. 288–295.

[39] S. Y. Ni, Y. C. Tseng, Y. S. Chen, and J. Sheu, “The broadcast storm problem in a mobile ad hoc network,” in *Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM)*, 1999, pp. 151–162.

[40] F. Pachet, O. Delerue, and F. Gouyon, “Extracting rhythm from audio signals,” in *Proceedings of SONY Research Forum*, 2000.

[41] F. Pachet and A. Zils, “Automatic extraction of music descriptors from acoustic signals,” in *Proceedings of the International Symposium on Music Information Retrieval (ISMIR)*, 2004.

[42] J. Paulus and A. Klapuri, “Measuring the similarity of rhythmic patterns,” in *Proceedings of the International Symposium on Music Information Retrieval (ISMIR)*, 2002.

[43] C. E. Perkins and E. M. Royer, “Ad hoc on-demand distance vector routing,” in *Proceedings of IEEE Workshop on Mobile Computing Systems and Applications (WMCSA)*, 1999, pp. 90–100.

[44] J. Pickens, “Harmonic modeling for polyphonic music retrieval,” Ph.D. dissertation, University of Massachusetts at Amherst, 2004.

[45] G. P. Premkumar, “Alternative distribution strategies for digital music,” *Communications of the ACM*, vol. 46, no. 9, pp. 89–95, 2003.

[46] C. Roads, *The Computer Music Tutorial*. Cambridge, MA, USA: MIT Press, 1996.

[47] 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*, vol. 25, no. 2, pp. 217–251, 2005.

[48] SanDisk, “Mobile phones with memory cards,” retrieved from http://www.sandisk.com/pdf/retail/Mobile_Phones_with_Memory_Cards.pdf on August 2005.

[49] E. Scheirer, “About this business of metadata,” in *Proceedings of the International Symposium on Music Information Retrieval (ISMIR)*, 2002, pp. 252–254.

[50] P. Shrestha and T. Kalker, “Audio fingerprinting in peer-to-peer networks,” in *Proceedings of the International Symposium on Music Information Retrieval (ISMIR)*, 2004.

[51] H. Takeda, T. Nishimoto, and S. Sagayama, “Rhythm and tempo recognition of music performance from a probabilistic approach,” in *Proceedings of the International Symposium on Music Information Retrieval (ISMIR)*, 2004.

[52] Y. Theodoridis, J. Silva, and M. Nascimento, “On the generation of spatiotemporal datasets,” in *Proceedings of the Symposium on Large Spatial Databases (SSD)*, 1999.

[53] R. Typke, F. Wiering, and R. Veltkamp, “A survey of music information retrieval systems,” in *Proceedings of the International Symposium on Music Information Retrieval (ISMIR)*, 2005, pp. 153–160.

[54] G. Tzanetakis and P. Cook, “Musical genre classification of audio signals.” *IEEE Transactions on Speech and Audio Processing*, vol. 10, no. 5, pp. 293–302, 2002.

[55] G. Tzanetakis, J. Gao, and P. Steenkiste, “A scalable peer-to-peer system for music information retrieval,” *Computer Music Journal*, vol. 28, no. 2, pp. 24–33, 2004.

[56] C. Wang, J. Li, and S. Shi, “A kind of content-based music information retrieval method in a peer-to-peer environment,” in *Proceedings of the International Symposium on Music Information Retrieval (ISMIR)*, 2002, pp. 178–186.

[57] A. Wieczorkowska, “Musical sound classification based on wavelet analysis,” *Fundamenta Informaticae*, vol. 47, no. 1/2, pp. 175–188, 2001.

[58] A. Wieczorkowska and Z. Ras, “Audio content description in sound databases,” in *Web Intelligence: Research and Development*, 2001, pp. 175–183.

[59] B. L. Y. Fang, Z. Haas and Y. Lin, “TTL prediction schemes and effects of inter-update time distribution on wireless data access,” *ACM/Kluwer Wireless Networks*, vol. 10, no. 5, pp. 607–619, 2004.

[60] C. Yang, “Efficient acoustic index for music retrieval with various degrees of similarity,” in *Proceedings of the ACM International Conference on MultiMedia (MM)*, 2002, pp. 584–591.

[61] ——, “Peer-to-peer architecture for content-based music retrieval on acoustic data,” in *Proceedings of the ACM International Conference on World Wide Web (WWW)*, 2003, pp. 376–383.

[62] G. V. Záruba and S. K. Das, “Off-the-shelf enablers of ad hoc networks,” in *Mobile Ad Hoc Networking*, S. Basagni, M. Conti, S. Giordano, and I. Stojmenovic, Eds. IEEE Press & Wiley-Interscience, 2004, pp. 47–68.