Data Dissemination and Query Routing in Mobile Peer-to-Peer Networks

Thomas Repantis, Vana Kalogeraki
Department of Computer Science & Engineering
University of California, Riverside
Riverside, CA 92521, USA
{trep, vana}@cs.ucr.edu

Abstract

In this chapter we study the problems of data dissemination and query routing in mobile peer-to-peer networks. We provide a taxonomy and discussion of existing literature, spanning overlay topologies, query routing, and data propagation. We proceed by proposing content-driven routing and adaptive data dissemination algorithms for intelligently routing search queries in a peer-to-peer network that supports mobile users. In our mechanism nodes build content synopses of their data and adaptively disseminate them to their most appropriate peers. Based on the content synopses, a routing mechanism is being built, to forward the queries to those peers that have a high probability of providing the desired results. We provide an experimental evaluation of different dissemination strategies that shows that content-driven routing and adaptive data dissemination is highly scalable and significantly improves resource usage.

Keywords: *Distributed systems, mobile peer-to-peer networks, Bloom filters, content-driven routing, adaptive data dissemination.*

1 Introduction

Mobile ad-hoc networks composed of mobile devices such as laptops, cell phones and PDAs with limited communication power and transmission range have emerged as a widely deployable infrastructure without the need of centralized support (Cao et al., 2005; Kortuem et al., 2001; Papadopouli and Schulzrinne, 2001). The typical characteristic of these networks is that the users are interested in receiving data and services available in their vicinity, or want to be notified about local events that are pertinent to their interests. Receiving this information by forming mobile peer-to-peer networks offers several advantages in comparison to retrieving it from fixed access points or the satellites of a cell phone carrier: First, the costs for installing, maintaining, and operating an infrastructure to provide the requested data are avoided. Second, data are updated automatically as users move, without the overhead of continuously collecting updates in centralized locations. This is particularly useful for locations where users change frequently, such as busy streets or stores. Third, users find themselves in places and situations where an infrastructure may not exist, or is not accessible. Examples include trying to access traffic information while driving in isolated areas (Xu et al., 2004), or trying to retrieve emergency notifications after a natural disaster. Finally, mobile peer-to-peer networks enable users to publish data in a cost-efficient way, in addition to receiving them. This gives rise to a wealth of new applications, such as identifying nearby users with similar interests (Amir et al., 2004), or locally organized events. Thus, enabling mobile devices to dynamically self-organize in ad-hoc networks and communicate in a peer-to-peer fashion enables cost-effective data dissemination for a variety of environments and applications. Mobile nodes that are in the transmission range of each other can communicate with their peers directly. To communicate with peers outside a node's transmission range, messages are propagated across multiple hops in the network.

The goal of this chapter is to study the problems of data dissemination and query routing in mobile peer-to-peer networks. We explore the synergy of mobility and peer-to-peer, focusing on the above problems. We identify the challenges introduced by mobility. We survey existing approaches and identify their shortcomings. Finally, we explore promising data dissemination and query routing solutions and evaluate their performance and overhead experimentally.

Delivering data to interested users in an efficient manner in a mobile environment challenges existing peer-to-peer solutions for wireline networks due to several reasons: First, peers have limited bandwidth and energy, which restrains the message overhead allowed for data dissemination protocols. Second, peer connections are transient, due to the frequent movement of the mobile nodes. This makes it costly to maintain an overlay topology on top of the underlying network topology, which is the way wired peer-to-peer networks are built, and to keep up-to-date routing information. The performance of wired peer-to-peer dissemination protocols would be further exacerbated in a mobile environment due to churn and network partitioning. Churn is introduced by peers frequently joining and leaving the network, while network partitioning is caused by node mobility.

The remaining of this chapter is organized as follows: Section 2 surveys existing solutions for data dissemination and query routing in mobile peer-to-peer networks, focusing on overlay topologies, query routing for data discovery, and data propagation. Section 3 presents mechanisms for content-driven routing and adaptive data dissemination. Section 4 describes an experimental study of the performance of data discovery and dissemination mechanisms. Section 5 discusses avenues to future work. Finally, Section 6 presents our conclusions.

2 Survey of Existing Research

We extend our discussion of existing research in the area into three different directions: Topologies for organizing the network to facilitate peer interaction, query routing mechanisms for data discovery, and mechanisms for data propagation. Table 1 summarizes our taxonomy of the relevant literature.

Overlay Topology	Location-Aware Structure	GHT (Ratnasamy et al., 2002) SOLONet (Patil et al., 2004)
	Location-Agnostic Structure	VRR (Caesar et al., 2006)
	Exploiting Peer Heterogeneity	Tork (Brown et al., 2007) SGDACp2p (Xue et al., 2004)
Query Routing for Data Discovery	Physical Network Routing	DSDV (Perkins and Bhagwat, 1994) DSR (Johnson et al., 2001) AODV (Perkins, 1999)
	Guided Query Routing	RBB (Wolfson et al., 2006) SBSD (Ouksel, 2006) SBSD-SP (Lundquist and Ouksel, 2007)
Data Propagation	Supporting User Mobility	Distr. DB (Pitoura and Samaras, 2001) Distr. Pub/Sub (Burcea et al., 2004)
	Epidemic Dissemination	7DS (Papadopouli and Schulzrinne, 2001) PDI (Lindemann and Waldhorst, 2004) PeopleNet (Motani et al., 2005)
	Cooperative Caching	Infrastructure-Less: (Goel et al., 2002; Sailhan and Issarny, 2003; Shen et al., 2005; Yin and Cao, 2004)
		Infrastructure-Based: (Chow et al, 2004; Hara 2002; Lim et al., 2004)

Table 1. Taxonomy of representative research in mobile peer-to-peer data dissemination and query routing.

2.1 Overlay Topology
Topologies like Distributed Hash Tables (DHTs) impose structure on wired peer-to-peer networks in order to improve data lookup times. Creating similar overlay topologies for mobile peer-to-peer networks is more challenging due to the transient peer connections. A few studies of the feasibility of applying existing structures to mobile environments have been performed (Gerla et al., 2005; Hu et al., 2004; Seet et al., 2007).

2.1.1 Location-Aware Structure
One approach for building a topology that aids data dissemination in a mobile environment is to create an overlay structure that takes into account the topological proximity. As representative examples we discuss GHT and SOLONet.

Geographic Hash Table (GHT) (Ratnasamy et al., 2002) creates an overlay structure to support a data-centric storage model. It hashes keys into geographic coordinates, and stores a key-value pair at the node geographically nearest to the hash of its key. Hence, rather than using hashing in the namespace, as DHTs do, GHT hashes in the geographic space. To support this functionality and route messages to the corresponding nodes, GHT requires GPS support and is built on top of the GPSR (Karp and Kung, 2000) geographic routing protocol.

SOLONet (Patil et al., 2004) builds an overlay multicast tree using location information. The goal is to have nodes that are physically close to each other be neighbors in the multicast tree, and have the logical distance of any member node from the source node be proportional to its actual distance from the source. By taking advantage of location information, the relaying of messages through other nodes is minimized and the latency of data dissemination in the tree is reduced. However, providing such a location-aware multicast tree induces the overhead of storing and broadcasting precise location information. SOLONet provides approximate location information, by dividing the physical topology into smaller cells. Only location changes that result to crossing the border of a cell are reported. Furthermore, a leader node is responsible for maintaining and providing location information for all nodes within each cell.

2.1.2 Location-Agnostic Structure

Another approach is to assign identifiers to nodes that are independent of the topology and allow peers to maintain paths between their overlay neighbors, while storing data objects at the nodes whose identifiers are closest to the objects' keys. As a representative example we discuss VRR.

Virtual Ring Routing (VRR) (Caesar et al., 2006). builds such a location-agnostic overlay. It is implemented directly on top of the link layer and does not rely on an underlying network routing protocol. VRR provides both traditional point-to-point network routing and DHT routing to the node closest to a key. It does not require network flooding or translation between fixed identifiers and location-dependent addresses. Nodes are organized into a virtual ring ordered by their identifiers and each node maintains a small number of routing paths to its neighbors in the ring. The nodes along a path store the next hop towards each path endpoint in a routing table. VRR uses these routing tables to route packets between any pair of nodes in the network, i.e., a packet is forwarded to the next hop towards the path endpoint whose identifier is numerically closest to the destination.

2.1.3 Exploiting Peer Heterogeneity

Taking advantage of the heterogeneity of the mobile environment is another approach for building a mobile peer-to-peer topology. As representative examples we discuss Tork and SGDACp2p.

Tork (Brown et al., 2007) is a variable-hop peer-to-peer overlay that adjusts hop-count according to a peer's bandwidth capabilities. Hop-count refers to the number of physical network hops required for an overlay hop. For high-bandwidth peers, Tork offers O(1)-hop performance, while for low-bandwidth peers, Tork has multi-hop performance. Hop-count can be decreased, and consequently communication latency can be increased, with the use of larger routing tables on each peer. However this leads to increased network traffic to maintain these larger routing tables. Tork combines active stabilization and opportunistic updating to decrease that maintenance traffic. Since a variable-hop overlay is adaptive, peers of different bandwidth capacities can exist in the same overlay. A peer might have high bandwidth capacity in one interval and low bandwidth in another.

SGDACp2p (Xue et al., 2004) is a Stable Group Differentiated Admission Control algorithm to exploit correlated mobility patterns in mobile ad-hoc networks. Its goal is fast amplification of the total streaming capacity of a mobile peer-to-peer media streaming system. Consumer peers store media data during a streaming session and therefore become supply peers themselves. Thus, system capacity is amplified when the number of peers it serves increases. To communicate with a peer outside its transmission range, a mobile peer has to rely on one or many intermediate peers as relays, which can be disconnected very frequently. SGDACp2p tries to improve the media streaming efficiency and build more stable routes, by attempting to predict the future availability of wireless links, based on individual peer mobility models.

2.2 Query Routing for Data Discovery

Routing queries for data discovery has been studied extensively in wired peer-to-peer networks. (Zeinalipour-Yazti et al., 2004) provides an overview of relevant research. Improving upon the efficiency of simple flooding is even more crucial for mobile peer-to-peer networks due to the energy and communication constraints.

2.2.1 Physical Network Routing

In wireless ad-hoc networks several protocols have been proposed to route messages on the physical network (Broustis et al., 2006) and one could utilize those to route queries in mobile peer-to-peer networks. (Oliveira et al., 2003) compares how a mobile peer-to-peer application performs under three different routing protocols, namely Destination-Sequenced Distance-Vector routing (DSDV) (Perkins and Bhagwat, 1994), Dynamic Source Routing (DSR) (Johnson et al., 2001), and Ad-Hoc On Demand Distance Vector routing (AODV) (Perkins, 1999). DSDV is proactive in maintaining routing information, while DSR and AODV are reactive. The peer-to-peer application examined was Gnutella-like (Gnutella Protocol Development, 2003) file sharing. The study showed that each of the analyzed routing protocols performed well in some scenarios and had drawbacks in others. Hence, the authors showed that it is important to consider characteristics of both the application and the network when choosing a routing protocol.

2.2.2 Guided Query Routing

To efficiently use discovery mechanisms employed in wired unstructured peer-to-peer networks in a mobile setting, disseminating information that can guide queries can be utilized. We discuss two representative algorithms that follow this approach, RBB and SBSD (and its extension SBSD-SP).

Rank-Based Broadcast (RBB) (Wolfson et al., 2006) uses a hybrid push and pull approach for resource discovery. Mobile peers broadcast both reports regarding available resources, and queries. Upon receiving a broadcast, a neighboring peer incorporates the reports and queries into its local database and subsequently broadcasts its most relevant reports and queries. RBB determines how to rank the reports and queries in terms of relevance, when to broadcast them, and how many to broadcast. The relevance of a report depends on the queries in the local database (which represent the global demand in the network), i.e, more relevant reports satisfy more queries. RBB triggers a broadcast when new information can be communicated to the neighbors, either because enough new information was received or because the set of neighbors has changed. Finally, RBB determines how many reports and queries to broadcast using a formula. The formula computes the optimal transmission, according to the length of time between subsequent broadcasts, so that overall dissemination is maximized.

Self-Balancing Supply/Demand (SBSD) (Ouksel, 2006) extends to a mobile environment the traditional publish/subscribe paradigm. Brokers are responsible for managing subscriptions and delivering events, while subscribers register with the brokers their interest in certain events. SBSD relies on the supply and demand principle to self-balance latency and workload through local updates of utility values. The utility functions favor the replication of the most recent and highly frequent profiles and events, while taking into account the negative congestion effects of excessive replication on the whole network. The goal is to minimize the average latency to find the matching events, and maximize the percentage of answered queries, as the network scales up. Self-Balancing Supply/Demand with Subscription-based Permission (SBSD-SP) (Lundquist and Ouksel, 2007) extends SBSD to reduce redundant broadcasting. It does so without sacrificing coverage, by exploiting high density. For any given subscription, Subscription-based Permission (SP) prohibits a consistent set of brokers from accepting its replicas. This both enlarges subscription propagation areas and increases the variety of subscriptions stored by the brokers within a given area.

2.3 Data Propagation

Data dissemination in mobile peer-to-peer networks refers both to disseminating data that users are interested in and data that can guide users' queries efficiently in the network, for example by observing peer interaction patterns (Repantis and Kalogeraki, 2005). Existing solutions from the domains of distributed databases or publish/subscribe systems need to be adapted in order to be used in networks of self-organizing, frequently disconnecting, resource-constrained, highly mobile nodes.

2.3.1 Supporting User Mobility

We discuss here how user mobility can be supported by extending distributed databases and publish/subscribe systems.

(Pitoura and Samaras, 2001) presents various approaches to the problem of storing, querying, and updating the location of nodes in mobile computing. Location management techniques use information concerning the location of moving nodes stored in location databases, in combination with search procedures that exploit knowledge about the nodes' previous moving behavior. Various enhancements include caching, replication, forwarding pointers, and partitioning. The databases for storing the location of mobile objects are distributed in nature and must support very high update rates since the location of objects changes as they move. The two most common architectures for distributed databases used for storing the location of moving users are a two-tier scheme, in which the current location of each moving user is saved at two network locations, and a tree-structured distributed database, in which space is hierarchically decomposed into subregions. Graphic-theoretic approaches that employ regional directories, as well as centralized database approaches have also been proposed.

(Burcea et al., 2004) discusses how a distributed publish/subscribe architecture can be extended to support mobility, i.e., moving users with intermittent network connectivity. Several factors affect the performance of such an architecture, such as the network (bandwidth and latency, placement of brokers, broker topology, number of brokers), the users (connection and disconnection times, mobility patterns), and the application (number of publishers and subscribers, publishing rate, specificity of subscriptions, subscriber locality, subscription (interest) locality, event (publication) size). An analytical model for the network cost of supporting mobility is presented, as well as mobility algorithms and optimizations such as prefetching and logging. The analysis focuses on unicast traffic generated to support mobile users, as opposed to the regular multicast traffic used for event dissemination to stationary clients.

2.3.2 Epidemic Dissemination

Exploiting the mobility of the peers to improve data transfer throughput has led to utilizing epidemic data dissemination algorithms. We discuss three representative systems that employ epidemic data dissemination, 7DS, PDI, and PeopleNet.

Seven Degrees of Separation (7DS) (Papadopouli and Schulzrinne, 2001) is a system for data exchange between mobile or stationary peers, enabling data sharing, message relaying and network connection sharing. It tries to exploit the host mobility and the spatial locality of information and queries to provide efficient epidemic data dissemination. 7DS supports both peer-to-peer (cooperative) and server-to-client interactions. 7DS peers collaborate by data sharing, by forwarding messages, or by caching popular data objects. The performance analysis of 7DS showed that the density of the cooperative hosts, their mobility, and their transmission power have great impact on data dissemination.

Passive Distributed Indexing (PDI) (Lindemann and Waldhorst, 2004) is a distributed lookup service. PDI stores (key, value) pairs in index caches located at mobile peers. Index caches are filled by epidemic dissemination of popular index entries. By exploiting node mobility, PDI can resolve most queries without sending messages outside the radio coverage of the inquiring node. For keeping index caches coherent, PDI uses implicit invalidation (configurable timeouts), as well as explicit invalidation (lazy caching). Inconsistency in index caches due to weak connectivity or node failure is handled by value timeouts. Lazy invalidation caches reduce the fraction of stale index entries due to modified data at the origin node. Similar to index caches, invalidation caches are filled by epidemic distributions of invalidation messages.

PeopleNet (Motani et al., 2005) is a wireless virtual social network which mimics the way people seek information via social networking. This approach offers significant advantages for information that is location-, community- and time-specific. PeopleNet propagates queries of a given type to users in specific geographic locations, called bazaars. The bazaars are pre-determined geographic regions and each bazaar handles only queries of certain pre-determined types. For example, a sports related query is directed to a sports bazaar. Within each bazaar, the query is further propagated between neighboring nodes via peer-to-peer connectivity, until it finds a match. By comparing a swap and a spread model for query propagation within a bazaar, the authors of PeopleNet have shown that swapping leads to higher matches. The probability of matching is further improved if prior to swapping queries the peers exchange some limited information about their buffer contents. Based on this observation, PeopleNet uses a greedy algorithm which uses this limited information to decide which queries to swap.

2.3.3 Cooperative Caching

Several parameters can affect data availability when introducing replication in a mobile peer-to-peer network. (Budiarto et al., 2002) discusses replication dynamics, replication level, and replica placement. Replication can be static or dynamic, depending on whether the location and the number of replicas are decided before deployment or can change following the access patterns. Regarding the replication level, increasing the number of replicas decreases the cost of queries but increases the maintenance cost. Finally, several choices exist regarding the replica location, e.g., close to home, close to writer, or close to reader. The comparison of different replication schemes examining the above trade-offs shows that the performance of a replication strategy depends heavily on many conditions, including network scale, mobility, access ratio and access concentration.

We now discuss replication among peers, i.e., cooperative caching, that allows the sharing and coordination of cached data among multiple peers. We divide cooperative caching approaches according to whether they target infrastructure-less or infrastructure-based environments. In the first case all peers are equal and communicate using an ad-hoc network, while in the second case an infrastructure of static stations that act as servers exists to enhance data delivery.

Infrastructure-Less Environments. (Goel et al., 2002) proposes a Street-and-Building mobility model that is suitable for a civilian environment, such as a college campus or a downtown business area. It then presents a protocol that enables efficient and reliable data dissemination in mobile ad-hoc networks by applying the technique of Tornado coding. In the protocol, supplying peers of a data file broadcast the Tornado-encoded file segments to requesting peers. A requesting peer downloads the encoded segments from different supplying peers, at different times and in different locations. When it receives sufficient segments, a peer is able to reconstruct the original file and also become a supplier. By using Tornado coding, the proposed protocol is more robust against packet losses, and highly efficient in data transmission, eliminating the need for multiple unicast downloading connections from multiple suppliers. The Street-and-Building mobility model is used to determine the Tornado coding parameters.

Proximity Regions for Caching in Cooperative MP2P Networks (PReCinCt) (Shen et al., 2005) is a cooperative caching scheme that aims to improve data accessibility in mobile environments. PReCinCt divides the network topology into geographical regions. Each region is responsible for a set of keys representing the data. Hashing is used to map keys to regions. To reach the region of a data object, queries are routed using a geographic-aided routing protocol such as GPSR (Karp and Kung, 2000). After reaching an object's region, localized flooding is used to locate the peer holding the requested object.

PReCinCt cooperatively caches data among a set of peers in a region. Cache replacement is determined by considering the importance of a data object to the peer caching it, and also to other peers in the same region. A utility function is employed to evaluate the importance of each data object in a peer's cache. That function takes into account the popularity of an object in its region, the size of the object, and the region distance between the requesting and responding peers. PReCinCt also employs a hybrid push/pull mechanism to maintain data consistency among replicas in the network, and manages to reduce both access latency and energy consumption.

(Yin and Cao, 2004) compares cooperative caching techniques that can improve the query delay and message complexity of data accesses in mobile ad-hoc networks. Specifically, three techniques are compared, namely caching the actual data, caching the data path, and a hybrid approach that combines the two. Caching the data enables intermediate nodes to serve future requests, instead of fetching data from a data center. Caching the data path enables mobile nodes to use it to redirect future requests to the nearby node which has the data instead of the faraway data center. Finally, in the hybrid approach caching the data is preferred for small data sizes, while caching the data path is preferred for data that expire soon, or data that are many hops away.

(Sailhan and Issarny, 2003) focuses on Web data caching to minimize the energy cost of peer-to-peer communication. The environment in this case is a mobile ad-hoc network of peers that share their Web access. The cooperative caching protocol builds upon the Zone Routing Protocol (ZRP) (Haas, 1997). ZRP implements proactive routing with mobile nodes and reactive routing with stationary nodes. The local cache replacement policy weighs for each cached data object its popularity, the energy cost to access it remotely, and its expiration time.

Infrastructure-Based Environments. In COoperative CAching (COCA) (Chow et al., 2004) data is replicated in low activity nodes so that the workload of high activity nodes can be reduced. The target environment includes mobile support stations that act as data servers, as well as mobile clients that can cooperatively cache data. Thus, when a client does not have a data object locally, it queries its neighboring client peers' caches, before enlisting the server for help. In COCA each mobile client and its neighboring peers residing in its transmission range work as a dynamic group that shares their cached data objects cooperatively. COCA is appropriate for an environment in which a group of mobile clients possesses a common access interest.

(Lim et al., 2004) discusses cache invalidation strategies for Internet-based mobile ad-hoc networks. In such networks, mobile nodes connect to the Internet directly through selected access points, or indirectly through message relay via other nodes. An enhanced scheme is proposed, called Global Positioning System-based Connectivity Estimation (GPSCE), for assessing the connectivity of a node to an access point. With this enhancement, a node can check whether it can access a server directly or indirectly through a multi-hop relay. Three cache invalidation schemes are compared, both push- and pull-based. The first one, called Aggregate Cache-based On Demand (ACOD), is based on a pull strategy; while the other two, namely modified timestamp (MTS) and

MTS with updated invalidation report (MTS+UIR) are based on a push strategy. The comparison indicates that the pull-based ACOD strategy provides high throughput, low query latency, and low communication overhead.

(Hara, 2002) discusses caching of data items in push-based information systems. In these systems a server repeatedly broadcasts data to clients through a broadband channel. Clients in such a system can construct and ad-hoc network to cache such data. Three cooperative caching strategies are discussed. These strategies shorten the average response time for data access by replacing cached items based on their access frequencies, the network topology, and the time remaining until each item is broadcasted next. Each of the three strategies differs in the set of nodes for which the average response time is minimized. In the case of a local optimal strategy, that set consists of only a single mobile node. In the case of a global optimal strategy, the set includes all connected mobile nodes. Finally, in a stable group optimal strategy, the set is a stable group of biconnected mobile nodes.

3 Mobile Peer-to-Peer Data Dissemination and Query Routing

When disseminating data in a mobile peer-to-peer network, the primary goal is to reach users with the same interests while keeping the number of propagated messages small. To achieve this goal we propose adaptively disseminating content summaries that are used to guide queries for data. As users move, their devices may keep establishing several short-lived connections to other peers along the way and thus become bombarded by unnecessary event notifications or advertisements about locally available services, data, and events. Even if a user is not highly mobile, the amount of forwarded queries from other mobile devices can be overwhelming, especially for devices with lower processing and communication capabilities. Guiding queries using the content summaries enables nodes to avoid this problem, by contacting only those peers that have a high probability of providing the desired results. Figure 1 illustrates our system's operation. In a mobile environment, changes to the stored data happen more often than they can be communicated to a single peer. We propose data dissemination algorithms that adaptively decide to which peers to propagate the content synopses to improve data discovery and make more efficient use of the bandwidth and processing power resources.

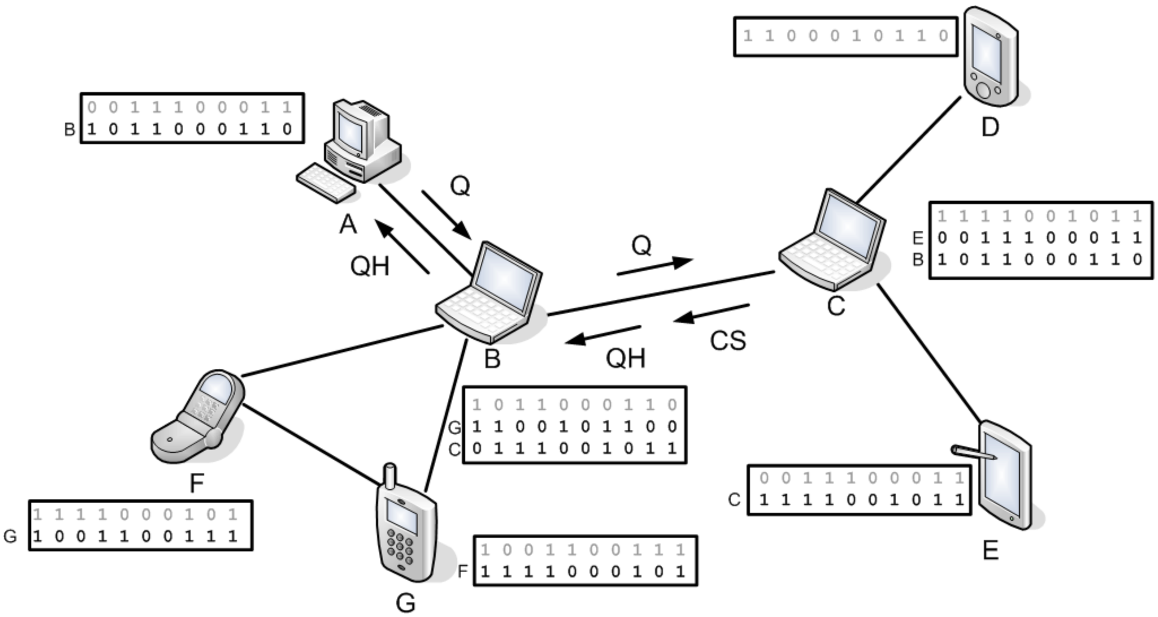

Figure 1. System operation example. Each node maintains a local content synopsis, as well as content synopses of remote peers. In this example, peer C propagated its content synopsis CS to peer B. B based on CS was able to route peer A's query Q only to C, and the result QH is routed back to A.

3.1 Overlay Model

We consider an overlay network of N nodes (peers) that store objects. We use the term "object" to refer to data, services or events. Each peer has a globally unique identifier (e.g., randomly generated using SHA-1) and maintains connections with other peers. The network is unstructured, decentralized and self-organizing, meaning that peers make their own decisions as to which peers to connect to or to query for objects. The number of connections of a peer can vary and is typically restricted by the resource capabilities of the peer. Peers that are not directly connected communicate through relaying. In other words, peers not only exchange messages with their neighbors, but also route messages coming from other peers. Each object is uniquely identified by the means of intrinsic references (Eshghi, 2002) which are generated when the object is first inserted in the system. Intrinsic references are based on the hash digest of the object's actual contents rather than its name or location and therefore allow us to create persistent, state-independent, and immutable storage. Alternatively, each object can be associated with a set of keywords to allow meta-data types of searching. The mechanisms presented in this work are orthogonal to the type of search and therefore we just focus on searching by an object's intrinsic reference.

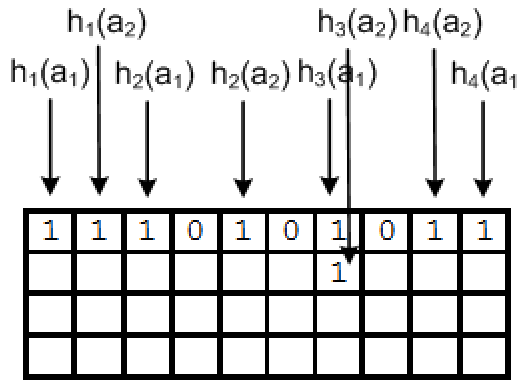

Figure 2(i). Counting Bloom filter: The counters keep track of the number of objects that are hashed in the same position.

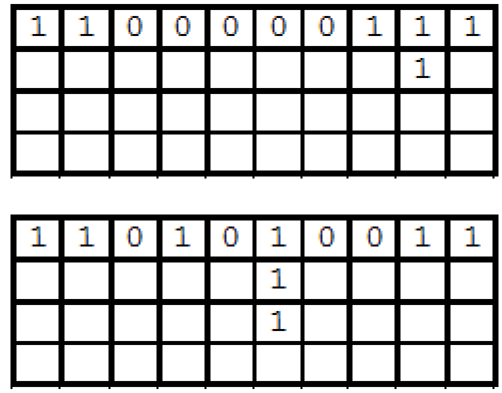

Figure 2(ii). Multi-level Bloom filter: The filter of each level is appended to that of the previous level.

3.2 Content Synopses

Each peer uses the *Bloom filter* data structure (Bloom, 1970) to build a synopsis of its local content which is disseminated to other peers. Assume that peer p has a group of n objects given by the set $S_p = a_1, a_2, ..., a_n$. The Bloom filter that represents the set S_p is described by a bit array BF_p of length m, with all bits initially set to 0. We assume k hash functions, $h_1, h_2, ..., h_k$ with $h_i : X \to 1...m$. Each hash function maps each element of the set S to a value between 1...m in a totally random fashion. For each element $s \in S$, the bits at position $h_1(s), h_2(s), ..., h_k(s)$ are set to 1. To determine whether a certain element x is in S, we check whether all the bits given by $h_1(x), h_2(x), ..., h_k(x)$ are set to 1. If any of them is 0, then we are certain that the data item x is not in the set S. If all $h_1(x), h_2(x), ..., h_k(x)$ are set to 1, we conclude that x is in S, although there is a certain probability that we are wrong. This is the case that a Bloom filter may yield a *false positive*. There exists a trade-off between k, m, n, and the accuracy of the objects' representation using Bloom filters (Repantis, 2005). This trade-off is investigated experimentally in section 4. Our system exploits the probability that a small number of false positives does not greatly affect the performance of our searching mechanism. This fact makes the Bloom filter approach highly suitable for locating objects accurately and fast.

To support the removal of members from the sets represented by the Bloom filters we use counting Bloom filters (Fan et al., 1998). In this approach, a counter is added to each bit in the filter, so that the number of objects that are hashed in the same position is counted. Removing members causes the corresponding counters to be decremented and therefore the representation accuracy is not affected. An example of a counting Bloom filter is shown in figure 2 (i). Each peer stores two types of filters, a *local filter* for the objects available locally at the node and *remote filters* for objects stored in remote peers, indexed by their IDs. Hence, to store multiple content synopses, we use multi-level Bloom filters. Figure 2 (ii) shows an example of a multi-level Bloom filter. Notice that the Bloom filter of each level is not merged but appended to that of the previous level. That approach consumes more memory space to store the Bloom filters, but allows us to estimate the location of a larger number of objects more accurately.

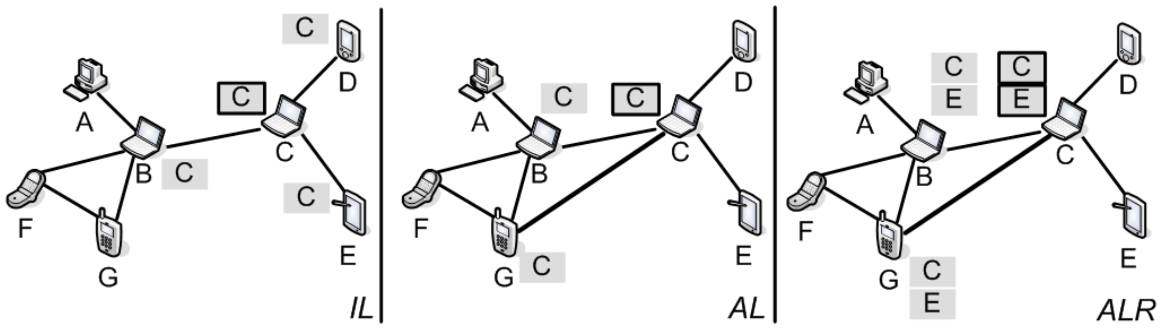

Figure 3. The content synopses dissemination strategies IL, AL, and ALR. In IL, node C propagates only its local synopsis to all peers one hop away. In AL, C propagates its local synopsis to selected immediate and remote peers. In ALR, C propagates both its local and stored remote synopses to selected immediate and remote peers.

3.3 Content-Driven Routing

Using content-driven query routing each peer stores the content synopses of other peers, and utilizes that information in order to route queries more efficiently. In particular, when a peer receives a query, apart from searching its local content, it also searches the stored content synopses of other peers. If there is no match in the peer's local content, the query is forwarded only to the immediate peers whose synopses indicate that they or their neighbors contain the requested object. Only if the object is not found in any content synopsis, is the query forwarded to a set of random neighbors. To provide a termination condition so that messages are not propagated indefinitely in the network when no objects are found, each message is associated with a time to live (TTL) field that represents the maximum number of times the message can be propagated in the network. Additionally, if a node receives the same message from two different peers, it detects the duplicates and discards the second message.

3.4 Adaptive Data Dissemination Strategies

We have implemented and compared three different strategies for content synopses dissemination, illustrated in Figure 3:

Disseminate local content synopsis to immediate peers (Immediate Local – IL). According to IL each peer sends its local content synopsis to all its immediate peers and routes queries by taking into account only the content synopses of its immediate peers. IL is simple, but of limited use: Since only a small number of content synopses is examined for the routing decision to be taken, a lot of the queries cannot be directed using the content synopses. The protocol then resorts to randomly choosing peers to further forward the query and thus generates a lot of traffic.

Disseminate local content synopsis to peers selected adaptively (Adaptive Local – AL). Using AL each peer sends its local content synopsis to a selection of peers, according to several parameters. Again the routing is done following the synopses of the local content of other peers. The recipients of the content synopsis of a peer are selected not only among its immediate neighbors, but also among remote peers whose queries have been answered successfully from local content in the past. The adaptive selection of the synopses recipients aims to make the content synopses available to the peers that have a high probability of using them again in the future and yet keep the number of synopses transfers limited. The parameters used to decide to which peers to disseminate the content synopses are described in a following section. As the number of content synopses used in routing is limited, AL is also often obliged to resort to randomly forwarding queries.

Disseminate both local and remote content synopses to peers selected adaptively (Adaptive Local Remote – ALR). ALR differs from the previous, in that the peers disseminate and use for their routing decisions not only the synopses of the local content of their immediate peers or peers they have interacted with, but also synopses of the content of remote peers. More specifically when a peer propagates its local content synopsis to other peers, it also propagates the content synopses of remote peers it has stored. Other peers store those remote content synopses together with the local synopsis of that peer and use them to route queries to it. Since each peer stores and disseminates remote content synopses of peers it is connected to, it can easily route queries for content stored in them. ALR enables the peers to examine a lot of content synopses before routing a query. Therefore a lot of the queries can be routed accurately and randomly forwarding queries is not used that often. The processing time spent in examining the content synopses is still small. The amount of information transferred between the nodes in order to disseminate the remote and the local synopses is higher than in the previous strategies, but is still restricted through the use of adaptive selection of the synopses recipients. The parameters used to decide to which peers to disseminate the content synopses are described in the next section.

Content synopses do not necessarily have to be propagated as individual messages, but can rather be piggybacked on the current usage messages (e.g. queries and replies). All of the above strategies are assuming a simple network infrastructure, where peers route queries through their immediate neighbors. In AL and ALR a more advanced overlay network is built, where peers can open direct connections to peers that provide them with good results (share similar interests with them) and routing can also be based on content synopses of peers outside a node's current horizon. Exploiting interest locality however comes at the cost of managing many –possibly short-lived– connections. Allowing "transient" content synopses to traverse the network would be the physical continuation of this approach. Yet, even though the cost of propagating a synopsis may not be too high, in a large-scale system the cost of maintaining up-to-date information throughout the path that a transient content synopsis travels, about where it came from and about how to reach its source would be prohibitive. This would be even more the case for dynamic environments, with frequent topology changes or content updates.

3.5 Adaptive Synopses Dissemination Parameters

Each node in the system is associated with a list of characteristics, which are summarized in Table 2. In order to decide which peers would benefit most from obtaining the content synopses, AL and ALR take into account the following parameters:

• The number of queries q_i a node has received by a peer, and their frequency. Peers that have sent a lot of queries to us will most probably make good use of our content synopsis in their routing decisions. A lot of forwarded queries indicate peers that route a lot of traffic. They can use our content synopsis to avoid sending us queries for content we do not have.

• The number of replies r_i a node has provided a peer with, and their frequency. This parameter identifies the popularity of our stored objects among specific peers. Peers that generated a lot of local hits and got a lot of replies by us to their requests will also most probably need our content synopsis in their routing decisions.

• The number of connections $conn_i$ other peers maintain. This parameter identifies the connectivity degree of a peer and is a factor in estimating the average number of messages per time unit this peer may route. A peer that plays the role of a hub in the network, routing many queries, will most probably need the content synopses more.

peer_id$_i$	The peer's globally unique identifier.
connected_peers$_i$	The list of peers currently connected to this peer.
object_list$_i$	The list of objects stored locally at the peer.
queries_received$_i$	The total number of queries this peer has processed.
searh_msgs_received$_i$	The number of search messages this peer has received, indexed by the IDs of the query originators.
local_hits$_i$	The number of local hits generated by queries, indexed by the IDs of the query originators.
sent_contentSynopsis_to$_i$	The list of peers that have received a current version of the local content synopsis.

Table 2. Parameters of each peer i in the system.

3.6 Dynamic Peer Behavior

Since the network is dynamic and self-organizing, nodes may leave or join independently. The system must be able to disseminate content synopses to reflect such changes in the connections. Moreover content synopses must be updated whenever an object is added, deleted, or its contents have changed. When the content is updated, a new content synopsis is disseminated by the peer. To minimize the traffic in the network our approach does not generate an update unless the contents of the peers have changed and groups individual Bloom filter updates into group updates to propagate them to the peers. Content synopses are disseminated due to both local and remote content changes.

As nodes in a mobile peer-to-peer network move, the peers in their vicinity change. Therefore old connections are dropped and new ones are established. We distinguish between two different modes in a peer's operation: i) When a peer is *static*, its position does hardly change and neither do the connections with its neighbors. ii) When a peer is *dynamic*, it moves frequently and its neighbors and the connections with them change constantly. A peer can alternate between static and dynamic mode. A peer is considered static by its neighbors, when it has been connected with them for longer than a time threshold t_s. To efficiently disseminate content synopses we push them to static peers, but let dynamic peers only pull them when needed. When a dynamic peer needs to search for something or route a query, it first asks for the content synopses of its current neighbors. On the other hand, a static peer's vicinity does not change often and by pushing synopses we avoid the overhead of explicitly asking for them. The same rules apply to a newcomer's decision to disseminate its own content synopsis.

When a peer permanently disconnects from the network, neither the content synopses of other peers stored in it, nor its content synopsis stored in other peers will be useful anymore. Its immediate peers sense the dropped connection and all the relevant content synopses are removed after a time threshold t_r. In addition, a DISCONNECTED message is sent to the non-immediate peers to remove their corresponding content synopses.

4 Experimental Study

We continue by presenting an extensive experimental evaluation of the presented mechanisms in large-scale, mobile, peer-to-peer networks. We used the Neurogrid simulator (Joseph, 2003) with the Gnutella (Gnutella Protocol Development, 2003) P2P communication protocol. Our implementation of content-driven routing and adaptive data dissemination was done in approximately 3500 lines of Java code. The parameters used in the simulation are presented in Table 3. We chose the network size to vary up to 3000 nodes, an estimate of the number of concurrently active nodes in a university campus. In our implementation we used counting, multi-level Bloom filters. To create the hash functions used in generating the Bloom filters, similarly to (The XLattice Project, 2005), we took advantage of a cryptographic message digest algorithm (SHA-1) and of its property of pseudo randomness. More specifically, we used SHA-1 to hash strings of arbitrary length, representing the peers' content, to 160 bits. We then built the hash functions by dividing the SHA-1 output into smaller sets of bits.

Node Parameters	Number of nodes	Varying
Network Parameters	TimeToLive of query messages	7
	Initial number of connections per node	3
	Minimum number of connections per node	3
	Maximum number of connections per node	10
	Network topology	Random
Content Parameters	Size of pool of available objects	2000
	Number of objects per node	30
	Distribution of objects over nodes	Uniform
Bloom Filter Parameters	Size of filter, in bits	10
	Number of hash functions	4
	Size of counter for each position, in bits	4
Simulation Parameter	Number of averaged measurements	20
	Number of searches per experiment run	400
	Number of search targets	100

Table 3. Simulation settings.

In our **first** set of experiments we investigated the tradeoff in the representation of objects through Bloom filters we discussed in section 3.3. Three parameters may affect the accuracy of the representation, i.e. the number of false positives yielded: The size of the Bloom filter in bits (memory overhead), the number of hash functions used (computation overhead), and the number of represented objects. We used counting Bloom filters with 4-bit counters, the simplest content-driven routing algorithm (IL), 4000 possible objects, and 1000 nodes and focused on the number of false positives.

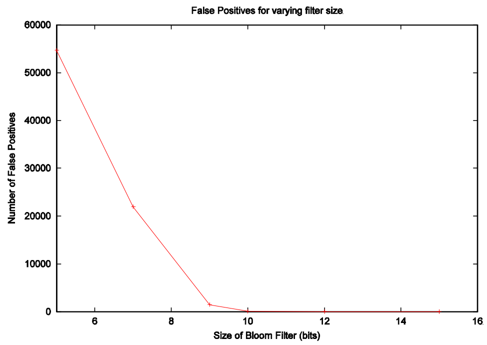

Figure 4. Bloom filter false positives for varying size of the filter (in bits).

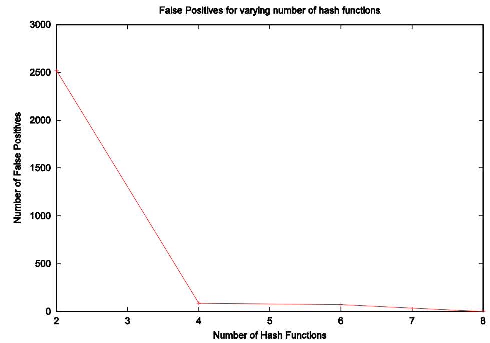

Figure 5. Bloom filter false positives for varying number of hash functions.

Effect of filter size to the number of false positives. As Figure 4 shows, filter size can greatly affect the number of false positives. Small filter sizes can result to thousands of false positives. However false positives are virtually eliminated above 10 bits (when representing 30 objects per filter and using 4 hash functions).

Effect of number of hash functions to the number of false positives. As Figure 5 shows, the number of false positives greatly decreases when using 4 hash functions or more (when representing 30 objects per filter and using 10 bits for the filter size).

Effect of number of represented objects to the number of false positives. Figure 6 shows that when using 4 hash functions and Bloom filters of 10 bits size, not more than 30 objects can be represented by a filter without significant loss in accuracy.

Taking into account the above results, we decided to use Bloom filters 10 bits long, 4 hash functions, and 30 objects per node (chosen out of 2000 unique objects) for the rest of the experiments that use content synopses.

In our **second** set of experiments we compared content-driven query routing to a traditional Breadth-First Search (BFS) algorithm.

Average message transfers during a search. Figure 7 shows that content-driven routing drastically decreases the number of query messages transferred during a search. As the number of nodes increases, the number of message transfers grows dramatically in flooding-based BFS, while the content-driven routing mechanisms manage to keep the message transfers almost at a fixed level. Thus, by using the network bandwidth efficiently, content-driven routing is able to scale to thousands of nodes. ALR, by disseminating content synopses of both local and remote peers adaptively, achieves the minimum number of message transfers needed to answer a query. It is noteworthy that the decrease in query messages between ALR and BFS reaches 97%.

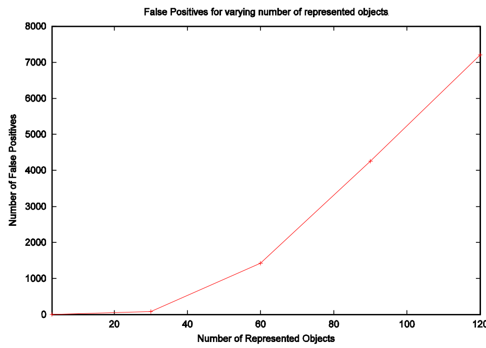

Figure 6. Bloom filter false positives for varying number of represented objects per node.

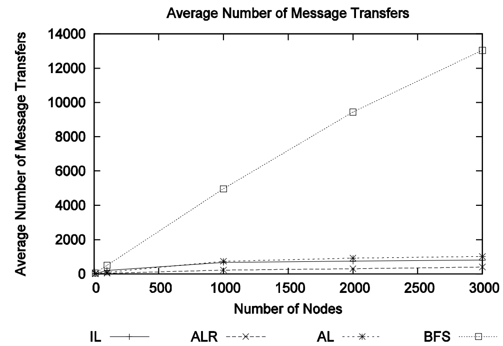

Figure 7. Average number of query messages sent during a search for different network sizes.

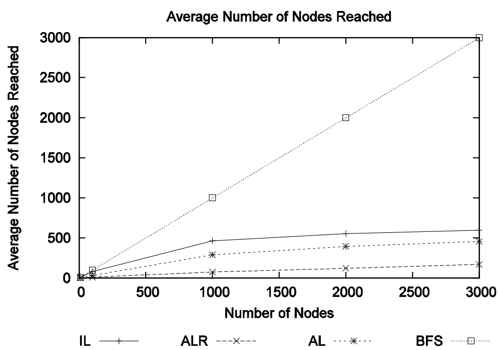

Figure 8. Average number of nodes reached during a search for different network sizes.

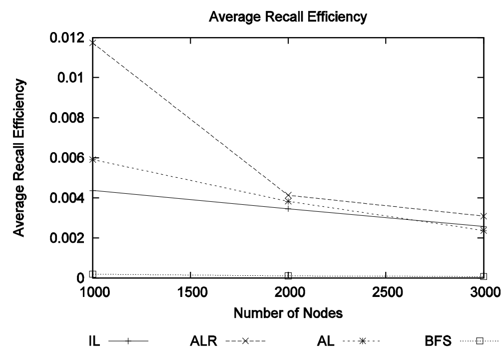

Figure 9. Average proportion of possible matches discovered over the number of queries transferred.

Average number of nodes reached during a search. Figure 8 again shows the benefits of content-driven routing in terms of bandwidth and processing power usage efficiency. All the content-driven routing techniques are able to provide query hits by contacting more than one order of magnitude less peers than BFS, which contacts a lot of peers unnecessarily. Moreover the content-driven routing strategies keep the number of reached nodes at an almost constant level, while the nodes that are reached with BFS grow linearly as the total number of nodes increases. The figure shows that the adaptive AL and ALR techniques guide queries more efficiently than the simplistic IL, in which content synopses are disseminated blindly to all immediate peers. ALR is again the most efficient and scalable technique of all, due to the adaptive use of the multi-level Bloom filters.

Average Recall Efficiency during a search. Figure 9 shows the value of the query messages that are disseminated during a search, in terms of their contribution to the discovery of possible matches. Even though BFS is able to discover a lot of matches, it does so by flooding, which results to its low recall efficiency. ALR again has the highest recall efficiency, followed by the other adaptive content-driven routing strategy, AL. The reason is that adaptive content synopses dissemination places the Bloom filters where they are more likely to be needed, achieving better performance than the blind IL. As the number of nodes grows, the proportion of the total matches discovered by the content-driven routing mechanisms decreases, since the queries are guided, in order to contact a small number of nodes and to produce a small number of messages.

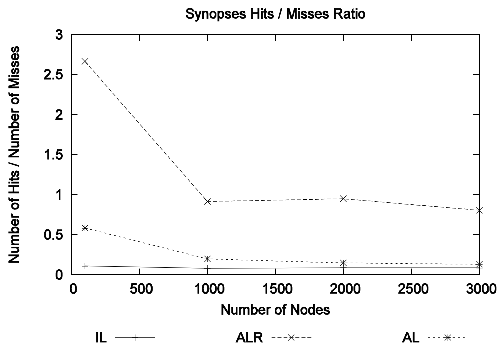

Figure 10. Ratio of hits over misses of the content synopses for different network sizes.

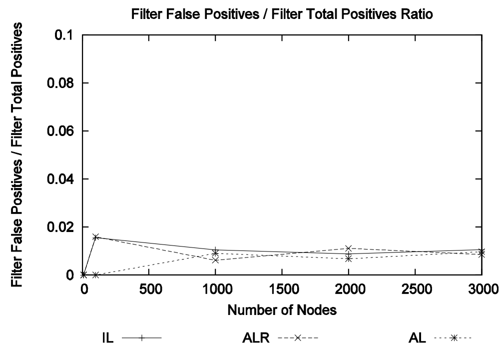

Figure 11. Ratio of Bloom filter false positives over total positives for different network sizes.

In our **third** set of experiments we compared the different content-driven routing protocols to each other in more detail.

Content synopses hits over misses. Figure 10 shows how much the query routing actually benefits from the use of the content synopses. We notice that simply placing content synopses of local content to immediate neighbors (IL) is useful for routing only about 10% of the queries. On the other hand, adaptively placing content synopses (AL and ALR) improves their usefulness to 20% for AL and to 90% for ALR. By disseminating local and remote content synopses, ALR manages to drastically decrease the number of Bloom filter misses and achieves a hits/misses ratio close to 1, meaning that half of the queries can be routed based on the content synopses.

False positives over total positives. Figure 11 shows that content-driven routing is extremely accurate. For all three routing strategies that use content synopses only a very small percentage (around 1%) of the total queries that are routed based on them is falsely routed, due to Bloom filter false positives. Thus, our choice of the Bloom filter parameters allowed us to minimize the false positives.

Total content synopses messages. Figure 12 shows the relative cost of the different content-driven routing protocols, in terms of content synopses dissemination messages. By simply disseminating content synopses only to immediate peers, IL keeps the protocol overhead low. However the usefulness of the content synopses in that approach is limited, as figure 10 indicates. AL on the other hand has to disseminate a lot of content synopses for them to be useful in query routing. ALR, by adaptively disseminating local and remote content synopses, manages to route queries effectively and yet keep the protocol overhead at a reasonable level, even as the number of nodes increases. That overhead is acceptable, if one takes into account the drastic saving of query messages ALR achieves.

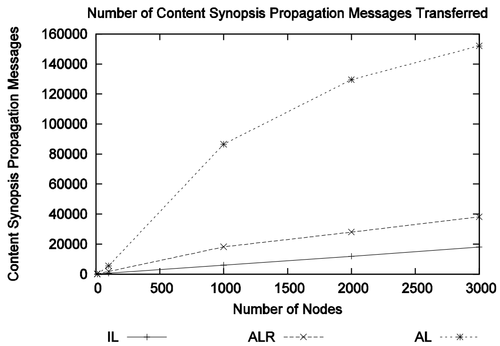

Figure 12. Total content synopses dissemination messages transferred for different network sizes.

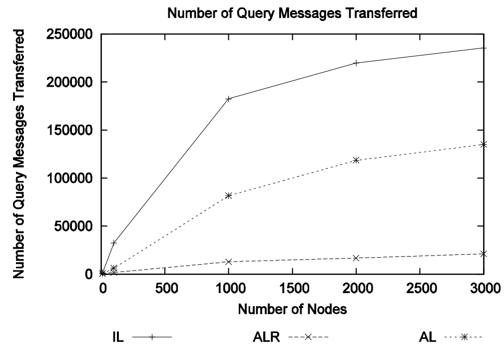

Figure 13. Total number of query messages transferred for varying network size.

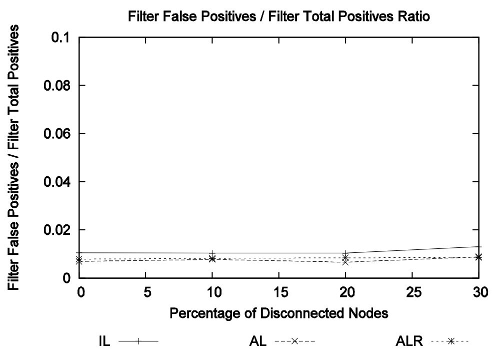

Figure 14. Bloom filter false positives over total positives for different percentages of disconnected nodes.

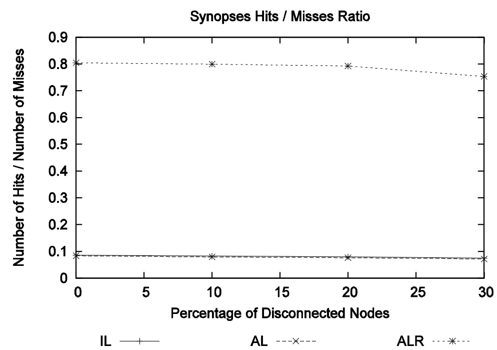

Figure 15. Content synopses hits over misses for different percentages of disconnected nodes.

Total query messages. Figure 13 shows the query messages transferred in the overlay, regardless of the underlying physical network. The adaptive strategies achieve significant savings in query messages. Especially ALR, by guiding queries through the use of local and remote content synopses, manages to keep the number of query messages low and easily scale to thousands of nodes. Bandwidth is thus used more efficiently in ALR than in any other of the content-driven routing protocols. ALR reduces the number of query messages by utilizing a lot of content synopses and placing them intelligently in the network. Notably, ALR decreases the number of query messages transferred by half an order of magnitude compared to AL and by one order of magnitude compared to IL.

In our **fourth** set of experiments we evaluated our protocols in a mobile environment, where peers leave the network dynamically. We gradually disconnected peers throughout the experiment run and we conducted experiments for disconnections reaching to 10, 20, and 30% of the total number of peers, which was initially 3000. We report the effects of the disconnections on the Bloom Filter behavior.

False positives over total positives. Figure 14 shows that content-driven routing remains very accurate even when a lot of peers disconnect. The neighbors of a leaving peer realize the disconnection and update their summaries, while peers further away also update their synopses when they are notified by a DISCONNECTED message they receive from the immediate peers. Hence false positives are not increased by the peer disconnections.

Content synopses hits over misses. Figure 15 shows that peer disconnections do not affect the success of the synopses in query routing either. ALR, which is the most aggressive mechanism in synopses dissemination, often routes queries successfully using the summaries. When a lot of peers disconnect, less synopses are available to help in query routing, hence a small degradation in the hit ratio.

5 Conclusions and Future Trends

In this chapter we have discussed data dissemination and query routing for mobile peer-to-peer networks. We have provided a comprehensive study of existing literature in the area, spanning overlay topologies, query routing, and data propagation. We have proceeded by presenting adaptive content-driven routing and data dissemination mechanisms. Based on content synopses, nodes forward queries intelligently only to their peers that are able to provide replies with a high probability. By disseminating the content synopses adaptively, we have shown how they can be strategically placed where they are most probably going to be needed.

User mobility and peer-to-peer interaction introduce several challenges with regards to data dissemination and query routing. Important questions that do not have a clear answer as of yet include:
• Can peer-to-peer applications be directly used in mobile settings?
• How does network scale affect query routing and data discovery?
• How structured should an overlay topology be?
• What are realistic user mobility models to evaluate overlay topologies under?
• How can peer-to-peer incentive, security, trust, and reputation techniques be applied to mobile environments?

Data dissemination and query routing in mobile peer-to-peer networks give rise to a multitude of exciting opportunities. Existing infrastructure-based mobile applications such as content delivery and media streaming can be augmented by peer-to-peer interaction. Furthermore, novel purely peer-to-peer platforms may emerge, such as rooftop or vehicular networks, introducing new use cases. These come to complement related research in sensor and ad-hoc networking.

References

Amir, A., Efrat, A., Myllymaki, J., Palaniappan, L., and Wampler, K. (2004). Buddy tracking – efficient proximity detection among mobile friends. In *Proceedings of IEEE INFOCOM*.
Bloom, B. H. (1970). Space/time trade-offs in hash coding with allowable errors. *Communications of the ACM*, 13(7):422–426.
Broustis, I., Jakllari, G., Repantis, T., and Molle, M. (2006). A comprehensive comparison of routing protocols for large-scale wireless MANETs. In *Proceedings of the 3rd International Workshop on Wireless Ad Hoc and Sensor Networks, IWWAN*.

Brown, A., Buford, J., and Kolberg, M. (2007). Tork: A variable-hop overlay for heterogeneous networks. In *Proceedings of the 4th IEEE International Workshop on Mobile Peer-to-Peer Computing, MP2P*.
Budiarto, Nishio, S., and Tsukamoto, M. (2002). Data management issues in mobile and peer-to-peer environments. *Elsevier Data & Knowledge Engineering*, 41:183–204.
Burcea, I., Jacobsen, H., Lara, E., Muthusamy, V., and Petrovic, M. (2004). Disconnected operation in publish/subscribe middleware. In *Proceedings of the 2004 IEEE International Conference on Mobile Data Management, MDM*.
Caesar, M., Castro, M., Nightingale, E., O'Shea, G., and Rowstron, A. (2006). Virtual ring routing: Network routing inspired by DHTs. In *Proceedings of ACM SIGCOMM*
Cao, H.,Wolfson, O., Xu, B., and Yin, H. (2005). MOBI-DIC: MOBIle DIscovery of loCal resources in peer-to-peer wireless network. *Bulletin of the IEEE Computer Society Technical Committee on Data Engineering*, 28(3):11–18. Special Issue on Database Issues for Location Data Management.
Chow, C., Leong, H., and Chan, A. (2004). Peer-to-peer cooperative caching in mobile environments.
In *Proceedings of the 24th International Conference on Distributed Computing Systems Workshops, ICDCSW*.
Eshghi, K. (2002). Intrinsic references in distributed systems. Technical Report HPL-2002-32, HP Labs.
Fan, L., Cao, P., Almeida, J., and Broder, A. (1998). Summary cache: A scalable wide-area web cache sharing protocol. In *Proceedings of ACM SIGCOMM*.
Gerla, M., Lindemann, C., and Rowstron, A. (2005). P2P MANETs - new research issues. In *Perspectives Workshop: Peer-to-Peer Mobile Ad Hoc Networks - New Research Issues*, number 05152 in Dagstuhl Seminar Proceedings.
Gnutella Protocol Development (2003). Retrieved October 7, 2008, from http://rfc-gnutella.sourceforge.net/.
Goel, S., Singh,M., Xu, D., and Li, B. (2002). Efficient peer-to-peer data dissemination inmobile ad-hoc networks. In *Proceedings of the 2002 International Conference on Parallel Processing Workshops, ICPPW*.
Haas, J. (1997). A new routing protocol for the reconfigurable wireless networks. In *Proceedings of the 6th IEEE International Conference on Universal Personal Communications*.
Hara, T. (2002). Cooperative caching by mobile clients in push-based information systems. In *Proceedings of the 11th International Conference on Information and Knowledge Management, CIKM*.
Hu, Y., Das, S., and Pucha, H. (2004). *Theoretical and Algorithmic Aspects of Sensor, Ad Hoc Wireless and Peer-to-Peer Networks*, chapter Peer-to-Peer Overlay Abstractions in MANETs. CRC Press.
Johnson, D., Maltz, D., and Broch, J. (2001). *Ad Hoc Networking*, chapter DSR: The Dynamic Source Routing Protocol for Multi-HopWireless Ad Hoc Networks. Addison-Wesley.
Joseph, S. (2003). An extendible open source P2P simulator. *P2P Journal*, pages 1–15.
Karp, B. and Kung, H. (2000). Greedy perimeter stateless routing for wireless networks. In *Proceedings of the 6th ACM/IEEE International Conference on Mobile Computing and Networking, MobiCom*.

Kortuem, G., Schneider, J., Preuitt, D., Thompson, T., Fickas, S., and Segall, Z. (2001). When peer-to-peer comes face-to-face: Collaborative peer-to-peer computing in mobile ad-hoc networks. In *Proceedings of the 2001 International Conference on Peer-to-Peer Computing, P2P*.
Lim, S., Lee, W., Cao, G., and Das, C. (2004). Performance comparison of cache invalidation strategies for Internet-based mobile ad hoc networks. In *Proceedings of the 1st IEEE International Conference on Mobile Ad-hoc and Sensor Systems, MASS*.
Lindemann, C. and Waldhorst, O. P. (2004). Exploiting epidemic data dissemination for consistent lookup operations in mobile applications. *ACM Mobile Computing and Communications Review*, 8(2):44–56.
Lundquist, D. and Ouksel, A. (2007). An efficient demand-driven and density-controlled publish/subscribe protocol for mobile environments. In *Proceedings of the 2007 Inaugural International Conference on Distributed Event-Based Systems, DEBS*.
Motani, M., Srinivasan, V., and Nuggehalli, P. (2005). PeopleNet: Engineering a wireless virtual social network. In *Proceedings of ACM MobiCom*.
Oliveira, L., Siqueira, I., and Loureiro, A. (2003). Evaluation of ad-hoc routing protocols under a peer-topeer
application. In *Proceedings of the IEEE Wireless Communications and Networking Conference, WCNC*.
Ouksel, A. (2006). Self-balancing selective information dissemination and discovery in mobile environments. In *Proceedings of the 22nd IEEE International Conference on Data Engineering, ICDE*.
Papadopouli, M. and Schulzrinne, H. (2001). Effects of power conservation, wireless coverage and cooperation on data dissemination among mobile devices. In *Proceedings of the 2nd ACM International Symposium on Mobile Ad Hoc Networking and Computing, MobiHoc*.
Patil, A., Liu, Y., Xiao, L., Esfahanian, A., and Ni, L. (2004). SOLONet: Sub-optimal location-aided overlay network for MANETs. In *Proceedings of the 1st IEEE International Conference on Mobile Ad-hoc and Sensor Systems, MASS*.
Perkins, C. (1999). Ad hoc on-demand distance vector routing. In *Proceedings of the 2nd IEEE Workshop on Mobile Computing Systems and Applications, WMCSA*.
Perkins, C. and Bhagwat, B. (1994). Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers. In *Proceedings of ACM SIGCOMM*.
Pitoura, E. and Samaras, G. (2001). Locating objects in mobile computing. *IEEE Transactions on Knowledge and Data Engineering*, 13(4):571–592.
Ratnasamy, S., Karp, B., Yin, L., Yu, F., Estrin, D., Govindan, R., and Shenker, S. (2002). GHT: A geographic hash table for data-centric storage. In *Proceedings of the 1st ACM InternationalWorkshop on Wireless Sensor Networks and Applications, WSNA*.
Repantis, T. (2005). Adaptive Data Dissemination and Content-Driven Routing in Peer-to-Peer Systems. In *M.Sc. thesis, University of California, Riverside.*
Repantis, T. and Kalogeraki, V. (2005). Data dissemination in mobile peer-to-peer networks. In *Proceedings of the 6th IEEE International Conference on Mobile Data Management, MDM*.
Sailhan, F. and Issarny, V. (2003). Cooperative caching in ad hoc networks. In *The 4th International Conference on Mobile Data Management, MDM*.

Seet, B., Lau, C., and Hsu, W. (2007). *Encyclopedia of Mobile Computing and Commerce*, chapter P2P Models and Complexity in MANETs. IGI Global.
Shen, H., Joseph, M., Kumar, M., and Das, S. (2005). PReCinCt: A scheme for cooperative caching in mobile peer-to-peer systems. In *Proceedings of the 19th International Parallel and Distributed Computing Symposium, IPDPS*.
The XLattice Project (2005). Retrieved October 7, 2008, from http://xlattice.sourceforge.net/.
Wolfson, O., Xu, B., Yin, H., and Cao, H. (2006). Search-and-discover in mobile p2p network databases. In *Proceedings of the 26th IEEE International Conference on Distributed Computing Systems, ICDCS*.
Xu, B., Ouksel, A., and Wolfson, O. (2004). Opportunistic resource exchange in inter-vehicle ad hoc networks. In *Proceedings of the 2004 IEEE International Conference on Mobile Data Management, MDM*.
Xue, G., Li, M., Deng, Q., and You, J. (2004). Stable group model in mobile peer-to-peer media streaming system. In *Proceedings of the 1st IEEE International Conference on Mobile Ad-hoc and Sensor Systems, MASS*.
Yin, L. and Cao, G. (2004). Supporting cooperative caching in ad hoc networks. In *Proceedings of IEEE INFOCOM*.
Zeinalipour-Yazti, D., Kalogeraki, V., and Gunopulos, D. (2004). Information retrieval in peer-to-peer systems. *IEEE CiSE Magazine, Special Issue on Web Engineering*, pages 12–20.