2003

Mobile Distributed Information Retrieval For Highly-Partitioned Networks

Katrina M. Hanna
University of Massachusetts - Amherst

Brian Neil Levine
University of Massachusetts - Amherst

R. Manmatha
University of Massachusetts - Amherst

Mobile Distributed Information Retrieval For Highly-Partitioned Networks

Katrina M. Hanna Brian Neil Levine R. Manmatha
Department of Computer Science
University of Massachusetts, Amherst, MA 01003
{hanna, brian, manmatha}@cs.umass.edu

Abstract

We propose and evaluate a mobile, peer-to-peer Information Retrieval system. Such a system can, for example, support medical care in a disaster by allowing access to a large collections of medical literature. In our system, documents in a collection are replicated in an overlapping manner at mobile peers. This provides resilience in the face of node failures, malicious attacks, and network partitions. We show that our design manages the randomness of node mobility. Although nodes contact only direct neighbors (who change frequently) and do not use any ad hoc routing, the system maintains good IR performance. This makes our design applicable to mobility situations where routing partitions are common. Our evaluation shows that our scheme provides significant savings in network costs, and increased access to information over ad-hoc routing-based approaches; nodes in our system require only a modest amount of additional storage on average.

1 Introduction

Disaster management and response, and many other applications of mobile computing, require first, *the rapid deployment of a communication system among mobile peers*; and second, that the mobile peers support *a robust method of sharing and retrieving essential information among users*.

Ad hoc routing protocols [13, 9, 16] are commonly proposed to address the problem of rapidly deploying a communication system among mobile peers. This is because disaster responders cannot rely on the availability of a pre-deployed communications infrastructure as it may be destroyed, blocked, or incapacitated by the disaster.

However, ad hoc routing protocols themselves are not robust. Specifically, ad hoc routing protocols are ineffective when the population of users is too sparse to form a fully-connected network. Additionally, geographic features and scene obstructions, as well as device power and size, may limit the communication methods and range available to radios, resulting in network partitions. Finally, adversaries may arrive on-scene to disrupt communications, route formation, and devices.

This paper investigates the provision of a mobility application that is subject neither to the assumption of a pre-deployed communication infrastructure, nor to the limitations inherent in ad-hoc routing. We propose and evaluate a mobile, peer-to-peer (p2p) application-level service that is tolerant of network partitions caused by sparse populations of users, attacks, or other failures and threats. In particular, we show that our system is more tolerant of network partitions and device failures than an ad hoc routing protocol linking mobile hosts to a centralized application server.

Specifically, the application/service we provide among peers is meant to address our stated second requirement of disaster management: information sharing and retrieval among users. While much networking research has concentrated on sharing collections of audio and video files, disaster response requires search of text- and image-oriented collections. Information Retrieval (IR) systems retrieve the documents and articles from stored collections that are the *most relevant* to a client-supplied query. Such systems can support disaster response by, for example, allowing emergency workers access to a large collection of literature and documents (e.g., medical algorithms, maps, chemical hazard sheets, field manuals, area information) on-scene using networked mobile computers.

Our p2p IR system specifically does not use ad hoc or other routing between peers in order to avoid the failures associated with such schemes, and it solves several problems. It allows mobile users use of an IR system without Internet connectivity to centralized server. It removes the assumption that a single host in a federation of mobile hosts is capable of indexing voluminous content or responding to numerous queries. It removes the single point of failure and attack that such a centralize host represents. Finally, it removes the assumption that there exists a network route to any particular server — ad hoc routing protocols do not provide coverage when physical

This paper was supported in part by National Science Foundation awards ANI-033055, EIA-9983215, and EIA-0080199; and in part by the Center for Intelligent Information Retrieval. Katrina Hanna's work is supported in part by a National Science Foundation Graduate Research Fellowship.

layer partitions exist nor when a protocol is successfully attacked. Though mobile computing devices can be resource poor, a group of mobile peers can share the work of indexing documents, storing those indicies, and responding to queries while providing coverage in a partitioned wireless environment. In fact, our results show that our system has fewer network costs than an ad hoc routing-based approach.

This paper is divided into two parts. First, we describe the design of our p2p system. We examine how best to divide collections of documents among peers with the following requirements: each peer stores only a small portion of the full collection; peers query only their own collection and the collections of neighbors in direct radio range; the set and number of neighbors are dynamic; and finally, the *accuracy* of resolved queries in such a distributed system must be sufficient.

Second, we compare the effectiveness and cost of our system to that of a centralized IR server connected to mobile peers with an ad hoc routing protocol. In particular, our evaluations show quantitatively that our system is significantly more fault tolerant than the latter scenario at the cost of higher (but manageable) storage requirements at peers.

The remainder of this paper is organized around those goals. We summarize related work (Section 2); overview the application of our designs to mobile devices (Section 3); review our experimental methodology and evaluate our system designs (Section 3.2); and offer conclusions in Section 5.

2 Related Work

The techniques we propose here are motivated by the needs of disaster management [3]. According to the Pan-American Health Organization, "health crisis management cannot be accomplished without access to timely and quality information" [10]. Emergency rescue and medical workers might arrive at a remote disaster location (e.g., tornado, refugee camp, or other long-term, subacute disasters) with information such as: GIS information; medical algorithms and literature; chemical hazard sheets; field medical manuals; the World Health Organization disaster medicine library (including images); immunization algorithms; and local or community info, such as information on health, fire, and police agencies, and related Incident Command System job action sheets and command structures. All of this information could not fit on a single (inexpensive) mobile device; in our scheme it would be accessible within a federated set of devices.

Our work is related to past work in peer-to-peer systems, distributed file systems and databases, and distributed information retrieval. Although our work contains elements of these fields, it differs from existing work in important ways.

2.1 Peer-to-peer systems

Most previous related work in peer-to-peer systems is focused on searching for files using well-known identifiers or a limited set of key words. Many p2p systems have been proposed in the research literature (in addition, several related commercial p2p applications, like Napster and Gnutella, are available). For example, Chord [17] and CAN [14] use consistent hashing techniques to provide a location service. These systems map identifiers (i.e., *keys*) to nodes in large-scale distributed systems. Although these and other related systems may be suitable for searches based on keywords, they are not useful for full-text search of documents: each and every word in the document collection would be an identifier, ruining the scalability of these systems.

Papadopouli and Shulzrinne have proposed a related system for mobility. Their system, called 7DS [11], allows mobile peers to share cached data with other peers in an ad hoc system. In contrast with our system, 7DS finds only matches with exact data (e.g., an exact URL). The same authors have also evaluated power management schemes in broadcast and multicast search schemes for mobile devices [12]. That work is applicable to ours.

2.2 Distributed IR

IR systems manage unstructured, full-text documents, while traditional database retrieval techniques require that documents be either highly structured or tagged with meta information (e.g., "name" or "address"). For example, Google offers a *centralized IR* service over unstructured web pages; INQUERY [4] is another example.

Distributed IR entails characterizing a subset of remote databases to determine which ones are most likely to contain useful information, searching that subset of databases, and then merging the results to create a single ranked list. This model assumes that each remote search engine indexes a specific database or collection of documents (the databases may or may not overlap). Moreover, it assumes that the included databases are highly available; if a particularly good collection is not reachable, accuracy results suffer. The main challenges in distributed IR are in determining how a client decides which database to search and how the results from multiple databases are merged to produce a single ranked ordering. Clients may choose databases based on resource descriptions that are provided by the owners of the databases. STARTS [7], for example, is a standard format for describing and communicating about the resources of each database.

Thus, previous work in distributed IR has largely concentrated on the database selection problem as well as the merging of results. In contrast, the focus of our system is to provide a fault tolerant, full-text search and retrieval capability for information spread over a nodes or peers. Speaking broadly, this fault tolerance enables resistance to censorship, terrorist attacks, disasters, or mobile peers simply moving out of range.

2.3 Distributed file systems

The provision of file systems and structured databases in a partitioned environment has been considered for decades (e.g., [6, 2]) and more recently in the mobile context (e.g, [18, 15]). We distinguish this work from ours in that it is attempting the more difficult problem of transaction consistency, e.g., propagating write operations across the network correctly despite partitions. To our knowledge, the mobility or unavailability of servers has not been studied in the IR literature. First, more than one document is likely to be relevant to a user's query, and any subset of documents will suffice; thus availability is the primary goal, but not the guarantee of specific data. Thus, as we discuss in the next section, evaluation of distributed information retrieval systems has a different set of metrics by which to evaluate performance. Second, we are not concerned with transactions and consistency, but rather with the savings gained by using local storage to provide availability and understanding how to eliminate routing and forwarding among peers.

3 Design and Evaluation of an Efficient IR System

In this section, we examine the problem of how to best divide collections of documents among peers. The result of this section is a single method; in the next section, we then evaluate that choice in a mobile environment to discuss its performance from a network perspective.

Recall the requirements of our system: each peer stores only a small portion of the full collection; peers query only their own stored collection and the collections of neighbors in direct radio range; the set and number of neighbors are dynamic; finally, the *accuracy* of query results in such a distributed system must be sufficient.

Often, collections of documents have a natural content focus, and many documents within a collection may be relevant to a client's query. However, when a peer is unavailable to clients (due to route failure or device failure or loss), the of portion documents stored by that peer are unavailable for query retrieval — accordingly, the accuracy of query resolution will suffer. Here, we propose a strategy for intentionally replicating documents among peers and evaluate its accuracy compared to several alternatives.

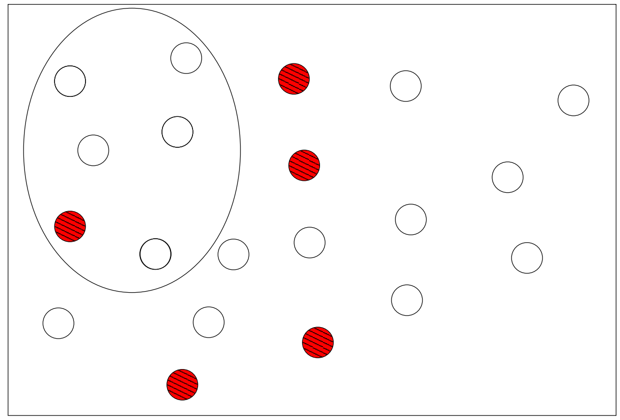

Figure 1: A Group of mobile peers. Colored nodes have a particular document. The circle represents a collection of a node's neighbors within radio range.

3.1 Replication Strategy

We expect that documents can be pre-loaded onto devices when users are federated. For this initial study, we assume the following simple replication strategy:

1. An initial peer receives a new document, and indexes it with some (pre-determined) probability p.

2. Regardless of whether the document is indexed, the peer passes the document along to all its neighbors, who follow the same algorithm.

We compare this hypothesized strategy with three other related alternatives subsequently.

Alternatively, documents can be added interactively: they can be broadcast to all peers as the documents are brought into the system. Each peer indexes the document with some pre-established probability p. Documents may be temporarily stored to overcome temporary network partitions. However, we don't consider on-line addition of documents in detail in this paper.

Figure 1 shows a diagram of a sample system. The system shown contains 20 nodes; the five shaded nodes represent peers that have probabilistically indexed a given document. The circled subset represents the peers within radio range of a host initiating a query. 802.11 offers a hello-message protocol that will detect such neighbors. No messages are relayed by neighbors and no ad hoc routing strategy is followed in our system. Given the random replication strategy, it is likely that neighbors have indexed a different, though not disjoint, subset of documents. In general terms, the more neighboring nodes the initial peer contacts to assist in resolving a query, the more accurate the merged results will be; however, contacting more neighbors delays the response and requires more work from the collective system.

We assume that nodes are homogeneous in the resources they have available to them. As with any file storage and retrieval system, the hardware determines the real limit on the amount of content that can be stored. In Section4.1.3 we discuss the storage requirements of each node in the system.

3.2 Experimental Methodology

The remainder of this section discusses our evaluation of the IR performance achieved by our model, which proves to be good. In Section 4, we discuss our evaluation of the networking performance of our system.

The quality of the information returned in an IR system may be evaluated by computing the *precision* of the system at R retrieved documents. Precision is the proportion of the information retrieved that is relevant to the query. A system may also be evaluated by computing the *recall* over R retrieved documents; recall is the proportion of all relevant information in the system retrieved for the user.

In our evaluations, the method of choosing relevant documents at each peer is constant. What changes is the document replication method, and thus the set of documents available at each peer. When useful documents are not available to a subset of peers, precision (and recall) is limited.

IR systems may be compared using different metrics. One such metric commonly used for IR systems is a *recall-precision graph*. In theory, a recall-precision graph is computed over all documents in the system. However, users performing a search are usually only interested in the top page or two of document results that are retrieved. Therefore, the performance of distributed retrieval systems is often measured solely in terms of the precision at the top R documents retrieved [5]; this is the metric we use in this paper.

Our method of evaluation is a standard IR research technique. The source databases we use in our experiments are from the Text Retrieval Conferences (TREC) run by the National Institute of Standards and Technology (NIST). Specifically we use volumes 1, 2, and 3 from TREC. The approximately 3.2 gigabytes of documents is divided by source and publication year. We used a set of short query strings related to a range of topics covered by the databases. The queries are standard TREC queries; the relevance scores of every document in the database in relation to these queries have been pre-established by NIST based on evaluations by real users. Therefore, it is possible to determine the exact precision of any result returned by the IR system as per NIST standards.

3.2.1 Comparison with Related Replication Strategies

To compare different strategies of dividing documents among peers, we evaluated the precision of each scheme. A set of five, randomly-chosen nodes initiated queries to varying numbers of peers in the system, again, randomly-chosen. We evaluated a simulated system of 50 peers; we felt this number was reasonable in the context of the application we wish to support (i.e., disaster management), though we were limited by a resource-intensive and time-consuming evaluation method.

We found that the method of distributing document indices across nodes greatly affects IR performance. It also affects the amount of space required at each node and in the system as a whole. The strategies we studied can be roughly categorized as *replicated* or *not replicated*, and consisting of homogeneous or heterogeneous content. We refer to the latter two categories as *sources-together* and *sources-split* respectively. In our experiment, the distinction between the two lies in whether we distributed documents with document-level granularity, or as one or more chunks of a database from a single source; e.g., all Wall Street Journal articles from 1999 would be a single source.

We compared the four combinations of the above categories:

- **Not Replicated, Sources-Together:** Database sources are divided into 50 chunks, each comprised, roughly, of the same number of documents. All documents in a chunk are placed on one node.
- **Not Replicated, Sources-Split:** We distribute the documents over the set of nodes in round-robin fashion. Each document is placed on exactly one node.
- **Replicated, Sources-Together:** All of the documents from a given database source are copied to three randomly-chosen nodes.
- **Replicated, Sources-Split:** This is our proposed distribution strategy as outlined in Section 3.1. Each document index is placed on each node with some probability p.

These four distribution strategies require very different amounts of disk space. Table 1 shows the total space required to store the documents in TREC 1-2-3 as well as the minimum and maximum at a single node. Note that the variation of storage costs at nodes is smaller with our scheme, which is essential for our assumption of devices with homogeneous resources. Figure 2 shows more detail on the replicated, source-split strategy that we advocate,

Strategy	**Total Space** (all nodes) MB	**Min node** MB	**Max node** MB
Not replicated, sources-split	3,185	60	66
Not replicated, sources-together	3,185	15	451
Replicated, sources-together	9,546	33	345
Replicated, sources-split ($p = 0.02$)	2,831	54	59
Replicated, sources-split ($p = 0.05$)	7,086	138	144
Replicated, sources-split ($p = 0.1$)	14,203	278	290

Table 1: Disk space required to store documents (Mbytes).

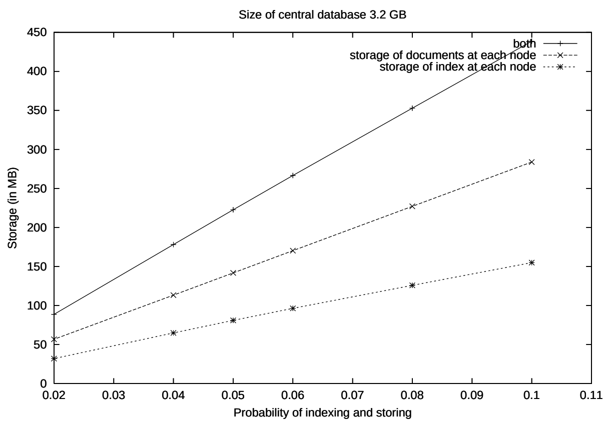

Figure 2: Example storage requirements of each node for the rep-split scheme with increase probabilities of storage and indexing.

including the average indexing and document costs per node in terms of p.

One disadvantage of our replication strategy is that there is no guarantee that documents will be archived. A minor modification could be made to ensure that the probability of archiving any document is one by simply ensuring that the first peer to receive a new document indexes it with probability one. However, a stronger constraint can be made. We could present an analysis that determines a bound on the value of p necessary to ensure that a specific document is available with high probability within a subset of nodes. However, such analysis does not directly tell us the probability of finding a relevant document since, in practice, search engines do not necessarily find every document that is indexed but only some fraction of all the relevant documents stored and indexed. When we factor this in, the probabilities of replication necessary to achieve good retrieval performance are much lower — i.e., if the search engine can only find 50% of the relevant documents then clearly it is not necessary to have all 100% of the relevant documents in the subset. Since search engines are quite difficult to model, the appropriate probabilities of replication necessary to achieve good retrieval performance must be determined empirically, which we do below.

For our queries we used the INQUERY system [4]. INQUERY implements an inference net model for full-text retrieval. In conjunction with INQUERY, we used CORI [5], which allows retrieval from a distributed set of document databases. In its default mode of operation, CORI characterizes the content of the databases and when presented with a query, chooses the best databases to contact. Ranked results from different databases can usually not be compared directly [5]; CORI uses a heuristic method to normalize document scores based on the maximum and minimum score the document could achieve, allowing rankings to be merged.

In our experiments we specified the peers to use in a particular query, bypassing CORI's auto-selection mechanism. However, we did use CORI's heuristic method for combining the results returned by peers.

In this experiment, for each strategy, we examined the performance of querying subsets of $s = 5, 10, 15, 20$ and 25 nodes. For each subset size we randomly chose s nodes, then ran 50 queries to the chosen nodes. For each subset size, we repeated this 20 times, averaging the results, shown in Figure 3. Error bars on all graphs represent standard deviations.

3.3 Evaluation Results

The four plots of Figure 3 show the results for retrieval of different numbers of documents ranging from 10 to 200. As is well known in IR, as more documents are retrieved by users, precision tends to drop. This is because a larger set of results makes it more challenging to locate only relevant results.

A number of important observations can be made from the graphs.

- First, in our experiment *the replicated-source-split (or rep-split) strategy achieved higher accuracy than the other three scenarios for values of* p *from* 0.04 *to* 0.10. For example, for retrieval of 20 documents, with subsets of five, rep-split with $p = 0.1$

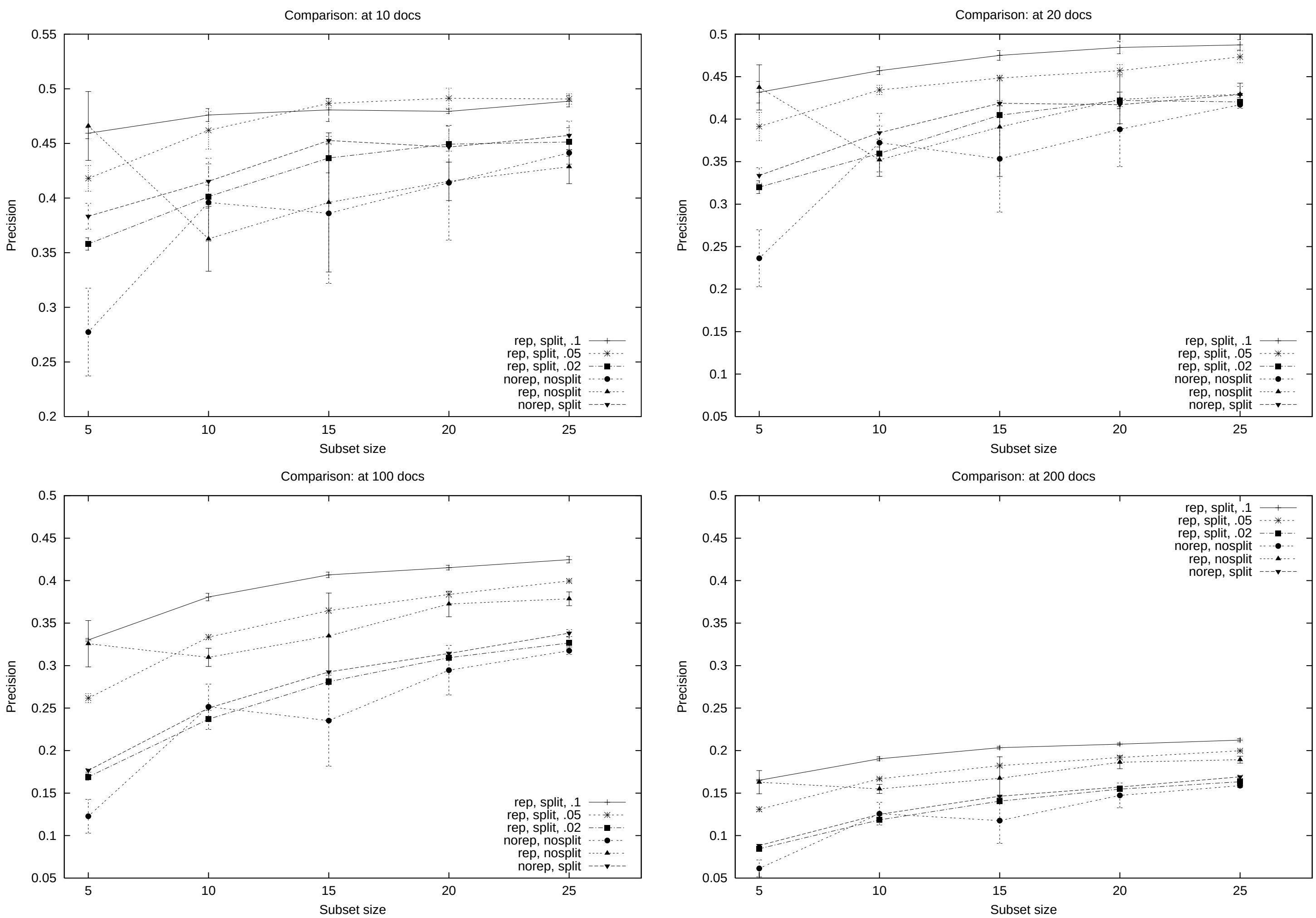

Figure 3: Precisions comparison of all techniques and varying probabilities of indexing for different numbers of retrieved documents: (Top Left) 10 documents; (Top Right) 20 documents; (Bottom Left) 100 documents; (Bottom Right) 200 documents.

has a 30% increase from a precision of 34% to 44% as compared to the non-replicated/split strategy. Moreover, the increase is greater over the no-replication/no-split and replicated/no-split strategies. When $p = 0.02$ there is not an advantage to our proposed strategy; this is also the point where the (total) storage costs are no greater than those of the non-replicated strategies(see Table 1). *Hence, we see a direct relationship between the cost of p in terms of storage and increased precision.*

- Second, *the other scenarios suffer from a high variance in the disk space used at nodes, and accordingly a high variance in their performance*. Here we see an advantage of dividing collections at the document level.

- Third, increased subset size does not make a significant difference in performance. From here we conclude that *most of the advantage of contacting multiple peers is gained at lower subset sizes*. The implication is that ad hoc routing is not required if enough neighbors are in range. In fact, in our evaluations of mobile nodes, presented in the next section, we do not employ any ad hoc routing between nodes. This simplifies the operational complexity of devices, reduces traffic on the network caused by flooded route requests (e.g., as done by AODV [13] or DSR [9]), and reduces work required of peers.

4 Mobility Evaluation

In the previous section, we proposed and evaluated a method of distributing documents among peers. In this section, we evaluate the IR quality and the costs of our system in a simulated mobile environment. Specifically, we compare our system (which we term *p2pir*) against the quality and cost of a mobile centralized IR server contacted by mobile clients through ad-hoc routing (which we term *ad hoc*).

4.1 Quantitative Results

We evaluated the simplest cause of partitions: sparse user populations. Accordingly, we evaluated a constant set of 50 peers mobile in four varying obstruction-free geographic areas: 500m-by-500m, 1000m-by-1000m, 1500m-by-1500m, and 2000m-by-2000m.

Our simulations of the eight scenarios (i.e., two systems, four geographic areas) used the Rice University Monarch mobility extensions to NS2 [1]. We used the AODV ad hoc routing simulator provided in version 2.1b9. Each node in the simulation was configured to simulate the range of an 802.11 interface. There was no packet loss in the simulation other than by MAC collisions. For all simulations, we assumed the 50 nodes moved according to the *random waypoint model* (RWM). This model was arguably to the advantage of the ad hoc system since our p2pir scheme would have performed better under a model that groups nodes together(e.g,[8]); i.e., RWM does not coordinate movement among peers. Nodes moved at a speed of 2 meters/second and paused at their randomly-chosen destinations for 20 seconds.

The second column of Table 2 shows the density of each scenario for average nodes; i.e., the average number of peers in radio range during the 1000-second simulation.

4.1.1 Precision

In our simulations, starting at 50 seconds into the simulation and then every 18 seconds, a node initiates a *query-round*: it sends 50 queries that are resolved by its direct neighbors in the p2pir scenario, and by the centralized server in the ad hoc scenario (if a route exists). For all six simulations, we evaluated the average precision of the query-rounds initiated at a consistent set of five nodes that we randomly chose; this limitation is due only to the extremely long processing time of evaluating queries. Data for number of neighbors of each specific node we evaluated is shown in Table 2.

Figure 4 (left) shows the precision of the nodes for our p2pir method for a range different numbers of retrieved documents. The results show that the p2pir strategy is able to manage node mobility quite well and with low variance. Because documents are randomly replicated, which neighbors are near to a node is of no consequence.

Figure 4 (right) shows the precision of the ad hoc method for a range of different retrieved document set sizes. There are two views of the results.

The first is measure how often connectivity to IR resources is achieved. The p2pir scheme has 100% connectivity (each peer has some portion of the database locally). The ad hoc scenario has connectivity much less often for any sparse population. Table 3 shows the percentage of times during the 52 query-rounds for which no route could be found to the central IR server. The p2pir technique offered a high precision result when the ad hoc service offered none; when a route was available, the p2pir quality was not significantly worse when compared to ad hoc precision.

The second viewpoint is to count the precision of unroutable queries as "0.0" in averaged precision shown. The 500m-by-500m field offered almost no partitions (see Table 3). However, degradation of the average service is extremely significant in sparser populations.

Note that variation among the nodes in performance is relatively small for the p2pir case despite a variation in the number of neighbors for each node of the five we evaluated. Routes available through ad hoc routing are unpredictable and cause high variances in performance.

We conclude that our p2pir service manages network partitions better than ad hoc routing. Next we evaluate the costs of each system.

4.1.2 Network Costs

Evaluating the amount of work performed by nodes in the p2pir method is simple. Remember that every 18 seconds, each peers issues a round of queries. If we assign a unit cost to each query group received by a node from its neighbors, then on average, the amount of work performed per node per query group is proportional to the average number of neighbors, shown in Table 2. We did not limit nodes to contacting five neighbors in order to reduce work, though Figure 3 predicts that this strategy would maintain high precision; as we stated in the last section, most of the advantage of contacting multiple peers is gained at lower subset sizes. Additionally, the precision would be no worse than that shown for the 1500m-by-1500m scenario where on average less than 5 neighbors were present.

For the ad hoc method, the number of queries answered by the 49 client-peers is zero. However, these nodes must forward traffic. Therefore, as an estimate of this amount of work, we compute the average number of paths to the central server on which each peer lies on average every query round. These results are shown in Table 4

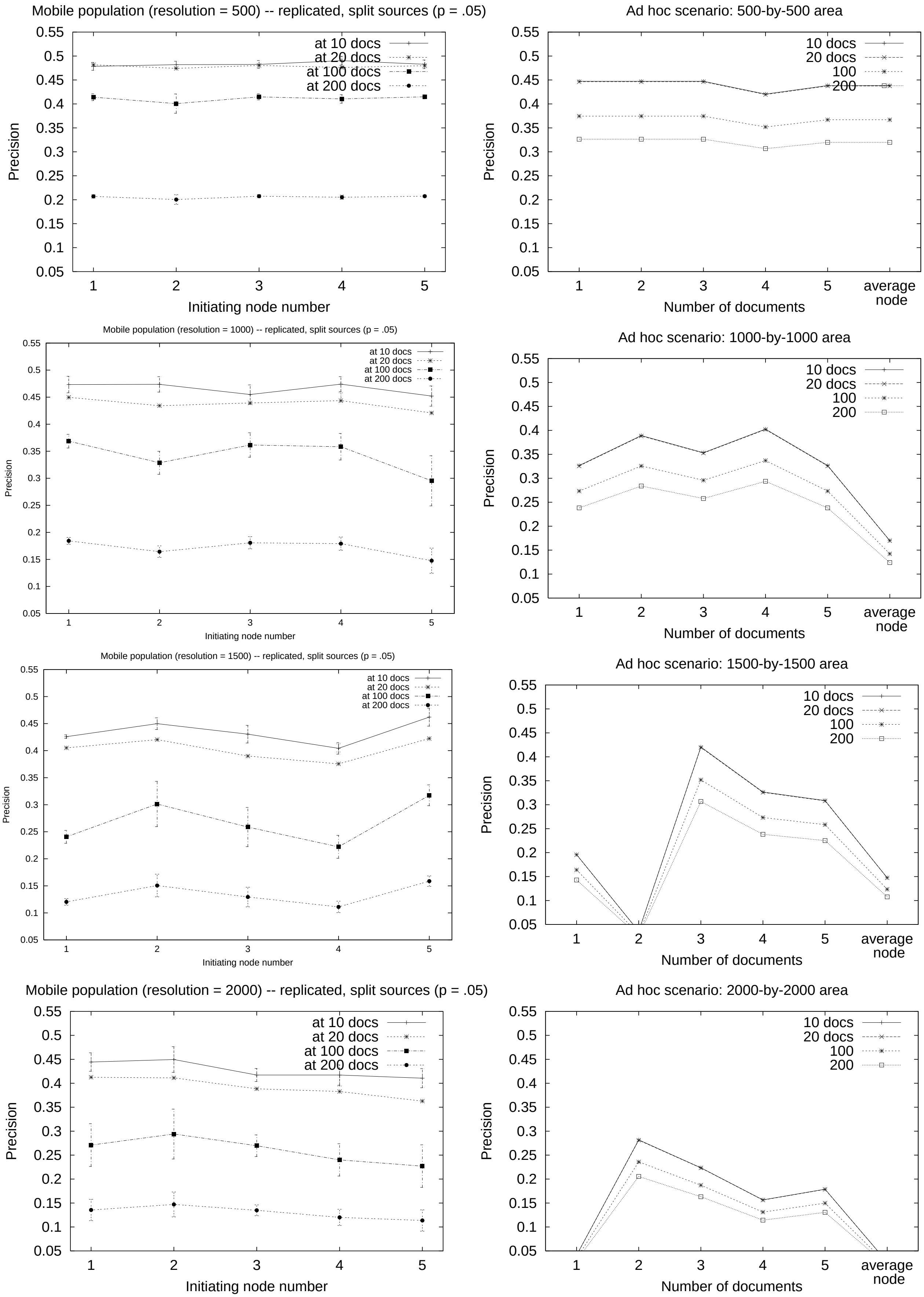

Figure 4: [Left] Avg. precision of five p2pir peers. [Right] Avg. precision of ad hoc peers. [Both sides] From top to bottom: 500m-by-500m; 1000m-by-1000m; 1500m-by-1500m; 2000m-by-2000m;

Mobility Area	Average node	Node 1	Node 2	Node 3	Node 4	Node 5
500m-by-500m	34±	36 ± 6.2	30 ± 9.3	37 ± 6.0	33 ± 7.1	38 ± 4.4
1000m-by-1000m	10±	14 ± 2.8	8 ± 2.0	14 ± 3.8	13 ± 4.0	6 ± 3.9
1500m-by-1500m	4±	3 ± 0.0	7 ± 3.2	4 ± 1.7	2 ± 0.0	8 ± 1.4
2000-by-2000	3±	4 ± 2.5	6 ± 3.2	4 ± 1.4	3 ± 1.0	2 ± 2.0

Table 2: Number of neighbors of each peer on average, and averages for specific nodes (± standard deviation).

Mobility Area	Average ± stddev	Node 1	Node 2	Node 3	Node 4	Node 5
500m-by-500m	$96\% \pm 14\%$	100%	100%	94%	98%	98%
1000m-by-1000m	$73\% \pm 22\%$	87%	79%	90%	73%	38%
1500m-by-1500m	$60\% \pm 26\%$	8%	94%	73%	69%	33%
2000m-by-2000m	$30\% \pm 21\%$	63%	50%	35%	40%	6%

Table 3: Connectivity percentages for the average node in each scenario for the 52 query rounds.

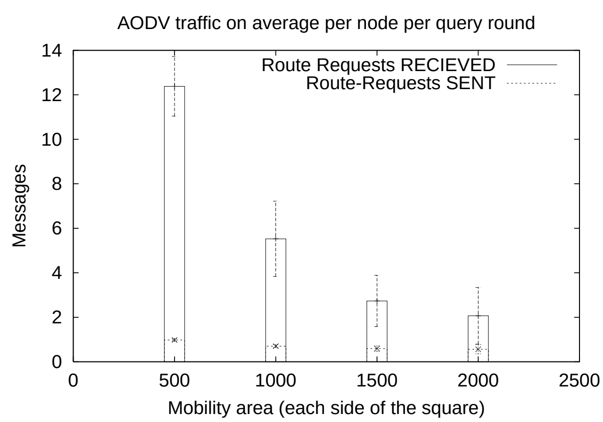

Figure 5: The major cost of running AODV at each node per query-group is RREQ messages. (Error bars indicate standard deviations.)

The centralized node performed work equivalent to the connectivity rates (Table 3) times 49 (nodes). The ad hoc scenario also had costs associated with running AODV; these are shown in Figure 5 for the average node on a per-query-round basis for each area size. The most significant costs are from broadcast and flooded AODV Route-Requests (RREQ). Route-replies and Route-Error messages were not significant. Less work was performed in sparser scenarios because traffic was blocked by partitions.

4.1.3 Storage Costs

The TREC 1-2-3 database we examined requires 3.2 Gb of storage, however, even with $p = 0.05$, nodes in our simulation require on average only 141Mb of storage space for documents and 81 Mb for indicies; this is within

Mobility Area	Avg. Paths through node
500m-by-500m	35± 11
1000m-by-1000m	38± 10
1500m-by-1500m	32 ±15
2000m-by-2000m	19 ±16

Table 4: The average number of paths that lies through each peer in the ad hoc scenario. (± standard deviation)

the hardware resources of even a currently available Compaq iPAQ 3580. Even with only five neighbors available on average, the nodes were able to retrieve documents with high accuracy. We expect our technique will scale to much larger databases as each individual mobile device is capable of storing more data. For example, Compaq IPaq 3850s accepts SD memory cards; currently, 512 Mb SD cards are available and cards up to 4 Gb are planned. Extrapolating our results, with 4Gb on each mobile device, a database of over 60 Gb could be distributed over 50 peers when $p = 0.05$.

5 Conclusion

We have designed and evaluated a system for fault tolerant mobile information retrieval. Fault tolerance and partitioning for distributed IR systems has not be previously studied. Random replication of split sources makes it difficult for attackers to remove specific indexed content from the system. Moreover, by setting our indexing probability p to low values (below 0.1), the system is able to return relevant results even when 45 out of 50 nodes are unavailable. We have shown that our design manages the randomness of mode mobility. While contacting only direct neighbors who change frequently, and not use ad hoc routing protocols, nodes still maintain good IR performance. This makes our design applicable to mobility situations where routing partitions are common. Our evaluation of storage requirements show that nodes require about the same about of storage, making our system ideal for collections of homogeneous hardware. We quantified the savings our system provides by not employing ad hoc routing, as well as showing it provides consistently good IR accuracy even when network connectivity varies significantly among peers.

Acknowledgments

We thank Dr. Jeffery Arnold, of the Department of Emergency Medicine, Baystate Medical Center, for the illuminating multidisciplinary collaboration from which this work developed.

References

[1] Network simulator version 2 (ns2). http://http://www.isi.edu/nsnam/ns.

[2] E.L. Abbadi and S. Toueg. Maintaining availability in partitioned replicated databases. *Trans. Database Systems*, 14(2), June 1989.

[3] B.A. Bissell, B.M. Becker, and F.M. Burkle Jr. Health care personnel in disaster response: Reversible roles or territorial imperatives? *Emergency Medicine Clinics of North America*, 14(2):267–288, 1996.

[4] J. Broglio, J.P. Callan, and W.B. Croft. Inquery system overview. In *Proceedings of the TIPSTER Text Program (Phase I)*. Morgan Kaufmann, 1994.

[5] Jamie Callan. *Advances in Information Retrieval*, chapter Distributed Information Retrieval, pages 127–150. Kluwer Academic Publishers, 2000.

[6] S. Davidson, H. Garcia-Molina, and D. Skeen. Consistency in partitioned networks. *Computing Surveys*, 17(3):341–370, 1985.

[7] L. Gravano, K. Chang, H. Garcia-Molina, and A. Paepcke. STARTS Stanford protocol proposal for Internet retrieval and search. In *Technical Report SIDL-WP-1996-0043, Computer Science Department, Stanford University*, 1996.

[8] X. Hong, M. Gerla, G. Pei, and C.-C. Chiang. A group mobility model for ad hoc wireless networks. In *ACM/IEEE MSWiM*, 1999.

[9] D. B. Johnson, D. A. Maltz, Y.-C. Hu, and J. G. Jetcheva. The dynamic source routing protocol for mobile ad hoc networks. IEEE Internet Draft, February 2002. http://www.ietf.org/internet-drafts/draft-ietf-manet-dsr-07.txt.

[10] Pan-American Health Organization. Health crisis and the internet: Harnessing the power of the internet for disasters and epidemics. *Prehospital Disaster Medicine*, (13):15–20, 1998.

[11] M. Papadopouli and H. Schulzrinne. "seven degrees of separation in mobile ad hoc networks". In *IEEE GLOBECOM*, November 2000.

[12] M. Papadopouli and H. Schulzrinne. Effects of power conservation, wireless coverage and cooperation on data dissemination among mobile devices. In *ACM SIGMOBILE Symposium on Mobile Ad Hoc Networking & Computing (MobiHoc)*, October 2001.

[13] C. E. Perkins and E. M. Royer. Ad hoc on-demand distance vector routing. In *IEEE Workshop on Mobile Computing Systems and Applications*, pages 90–100, February 1999.

[14] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker. A scalable content-addressable network. In *Computer Communication Review*, 2001.

[15] David Ratner, Peter Reiher, and Gerald Popek. Roam: A scalable replication system for mobile computing. In *Workshop on Mobile Databases and Distributed Systems (MDDS)*, 1999.

[16] Kimaya Sanzgiri, Bridget Dahill, Brian N. Levine, Elizabeth M. Belding-Royer, and Clay Shields. A Secure Protocol for Ad hoc Networks. In *Proc. IEEE International Conference on Network Protocols (ICNP)*, Paris, France, November 2002.

[17] I. Stoica, R. Morris, D. Karger, F. Kaashoek, and H. Balakrishnan. Chord: A scalable peer-to-peer lookup service for internet applications. In *Computer Communication Review*, 2001.

[18] A. Wang, P. Reiher, R. Bagrodia, and G. Popek. A simulation evaluation of optimistic replicated filing in a mobile environment. In *18th IEEE International Performance, Computing, and Communications Conference*, February 1999.