Spatial Queries in Disconnected Mobile Networks*

Xinjuan Zhu
Department of Computer Science
Xi’an Polytechnic University
Xi’an, Shanxi, 710048, China
zhu_xjcn@yahoo.com.cn

Bo Xu and Ouri Wolfson
Department of Computer Science
University of Illinois at Chicago
Chicago, IL 60607
{boxu,wolfson}@cs.uic.edu

ABSTRACT

In this paper we study in-network query processing in disconnected mobile environments, where both ad-hoc communication and infrastructure communication are available. Depending on how the infrastructure is utilized, various query-processing schemes are classified. Analytical models are developed to compute the average delay and the average energy consumption for each of these schemes. Based on the analytical models, the query processing schemes are compared under various environment conditions. It is found that none of the studied schemes is optimal in all the conditions. Therefore the paper provides a method that allows the optimal scheme to be chosen for a given set of environmental conditions.

Categories and Subject Descriptors

H.2.4 [**Database Management**]: Systems – Distributed databases; C.2.4 [**Computer-Communication Networks**]: Distributed Systems – Distributed databases; distributed applications

General Terms

Algorithms, Performance

Keywords

Mobile ad hoc networks, sensor networks, spatial queries, mobile data management, mobile peer-to-peer networks

1. INTRODUCTION

Mobile ad hoc and sensor networks are usually referred to as wireless ad hoc networks. Wireless ad hoc networks arise in applications where data sources are geographically spread (e.g., Intelligent Transportation Systems (ITS) [15], battlefield surveillance systems [16], and wildlife tracking [17]). Due to unattended and untethered node deployments, many applications in wireless ad hoc networks specify their interests using geographical predicates (i.e., spatial queries). Consider two common types of spatial queries in a wireless ad hoc network. The first type, the nearest neighbor (NN) query, finds the node that is the nearest to a certain geospatial query-point among all the nodes in the network. The second type, the range query, finds all the nodes that are inside a certain geospatial region. In a static sensor network, the answer of such a query is fixed and does not change over time. In a mobile sensor network, the answer to a NN or range query changes over time, as the nodes move. For example, the answer to a NN query at time 2 may be different from the answer at time 1, since the node that was closest to the query point at time 1 may have moved away.

Consider now the in-network processing of these queries. In this style, the query is processed collaboratively by the nodes themselves, rather than by a central site [1, 2, 3]. Existing in-network processing algorithms assume that the network is connected at any point in time. In a static network these algorithms compute the fixed answer. Furthermore, it is assumed that the query and answers are transmitted between neighboring nodes instantaneously, thus in a mobile and connected network, in-network NN and range query-processing algorithms compute the answer at the time the query is issued.

Now consider the case where a static network is disconnected, i.e. the communication graph (the graph which has the nodes as vertices, and where two nodes are connected by an edge if they are within transmission range) is disconnected. In this case, the in-network query processing may not be able to compute the answer due to disconnection of the nodes participating in the computation. For example, if the node issuing a NN query is disconnected from the node that is closest to the query point, in-network query processing is not possible.

A mobile disconnected network is a network in which some pair of nodes is sometime disconnected. In other words, a disconnected network may not be always disconnected, only sometimes; or it may be highly disconnected in the sense that most of the time an average node has no neighbors. In a mobile disconnected network, in-network query processing seems possible, but it may not be instantaneous. To continue the above example, there may not be a contemporaneous path between the querying node and the nearest-neighbor node, but as the nodes move the query may be carried to the query point, answered, and the answer carried back to the querying node. However, one would like to obtain an answer as of some time instance, either now or in the future. None of the existing in-network query-processing methods is able to do this. The reason is that the moving nodes need to guarantee a global spatial property that changes over time. While verifying that the property holds, it may change. For example, nodes *a* and *b* may send their id to the querying node, indicating that they are in the query-region, although it is possible that they were never in the query region at the same time.

ACM GIS '08, November 5-7, 2008. Irvine, CA, USA
(c) 2008 ACM ISBN 978-1-60558-323-5/08/11...$5.00

---

* Research supported by US NSF Grants DGE-0549489, 0513736, 0326284.

Therefore in this paper we propose replacing the NN and range query in mobile disconnected networks by a different type of query, which we call *spatial-existence* query. A spatial-existence query simply selects a node that is inside a geospatial region as of some time instance, either now or in the future. The difference from a range query is that a spatial-existence query is not requesting all the nodes in the query region. It is only requesting one of them. It does not matter which one, thus the query is nondeterministic. In contrast to a NN query, a spatial-existence query does not necessarily request the node that it closest to the query point, but one that is "close enough" (e.g. within 5 meters of the query point). We believe that spatial-existence queries are more appropriate for mobile disconnected environments because they can be in-network processed, whereas, as argued above, NN and range queries cannot. Observe that spatial-existence queries are useful in applications domains such as the digital battlefield and urban air pollution monitoring. For example, in a digital battlefield, assume that each soldier continuously takes pictures of his/her surrounding area using a helmet mounted camera. A spatial-existence query would ask for a picture taken by some soldier within 10 meters of the target. In a city, assume that sensors are mounted on buses which detect the air pollution level. A spatial-existence query would ask for the air pollution level detected by a bus inside a certain geospatial region. Furthermore, spatial-existence queries can be in-network processed in a mobile disconnected network. This is because, unlike the processing of NN or range queries, the processing of spatial-existence queries does not require visiting all the nodes in the queried region; the query is satisfied when one node is found.

Figure 1 depicts in the shaded boxes the environments for which the proposed query type (in the shaded ellipse) is most appropriate.

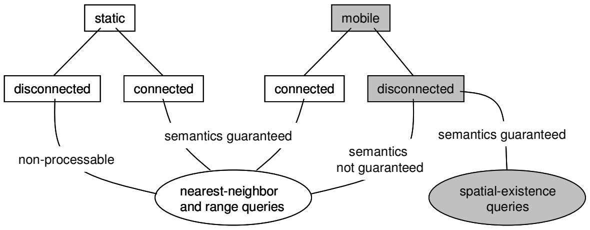


Figure 1. In-network query processing environments

In this paper we provide the analysis for spatial-existence queries. Particularly, we analyze a store-and-forward query processing scheme. Store-and-forward (also called epidemic routing), has been proposed as an approach for data dissemination or routing in disconnected mobile networks (see e.g., [4, 8, 9, 10]). In the store-and-forward scheme, a node receiving a data item (a query or an answer in our context) carries the data item as it moves, and forwards the data item to new nodes that it encounters. The analysis is based on ordinary differential equations (ODE's) to study the delay and energy consumption of the scheme under various environmental parameters such as the mobility, the network density, and the query selectivity.

In many environments, e.g. in an urban area or a digital battlefield, in addition to short-range P2P communication, mobile nodes can also communicate over long distances using an infrastructure such as a cellular or satellite network. If an infrastructure is available, then the query can be propagated either via P2P or via the infrastructure, and so can the answer. This gives rise to four query processing schemes. In this paper we study three schemes that utilize the infrastructure in query processing, and compare them along with the fourth, the pure P2P scheme. The performance metrics include the delay, the energy consumption, and the combination of the two. It turns out that none of the studied schemes is optimal in any individual performance metric for all the environmental conditions. Therefore the paper provides a method that allows the optimal scheme to be chosen for a given performance metric and a given set of environmental conditions.

In summary, the contributions of the paper are as follows:

1. We propose the spatial-existence query as a more appropriate spatial query type than the NN and range query in the disconnected mobile networks.
2. We develop analytical models to compute the average delay and the average energy consumption for various spatial-existence query processing schemes, some of which exploit the infrastructure communication.
3. We provide a method that enables users to choose the optimal scheme for a given environmental condition and a given performance metric (delay, energy consumption, or their combination).

The rest of this paper is organized as follows. In section 2 we introduce the model and the assumptions. In section 3 we analyze the query processing scheme which relies purely on P2P communication. In section 4 we analyze the schemes that combine P2P and infrastructure communication, and compare them together with the pure P2P scheme. In section 5 we discuss relevant work and in section 6 we conclude the paper.

2. MODEL AND ASSUMPTIONS

In this section we describe the system model and the assumptions made in our analytical model.

2.1 Mobility and Communication

- The system consists of a fixed set of N+1 point (i.e., without an extent) mobile nodes that move in a closed area. For example, the mobile nodes may be soldiers in a digital battlefield. Every mobile node is capable of short-range wireless communication (such as Bluetooth or 802.11), with the *transmission range* denoted by r. In other words, two mobile nodes can directly communicate if the distance between them is at most r.
- The system is sparse. Specifically, at any point in time, for each mobile node, there is at most one other node within its transmission range.
- An *encounter* is an event in which the distance between two mobile nodes A and B becomes smaller than r. As long as A and B stay within transmission range they will not encounter each other again, but they may do so after they disconnect (i.e., the distance between them becomes bigger than r). Upon an encounter between A and B, A and B exchange the answers/queries they carry. During the exchange, one of the two nodes, say A, transmits to another node (B) the id's of the answers/queries that A is carrying. B replies with the id's of the

answers/queries that are new to B (i.e., have not been received by B). Then A transmits to B the requested answers/queries. Finally B transmits to A the answers/queries that B is carrying and are new to A. The exchange is finished while A and B stays within the transmission range. As long as A and B stay within transmission range there is no reason for them to communicate beyond this initial exchange. This is so because neither A nor B has a third neighbor to interact with during the contact interval of A and B (according to the previous assumption), and therefore none of them can communicate to the other new answers or queries.

- For any pair of mobile nodes, they encounter each other by a Poisson process with intensity λ. The encounter processes for all pairs of mobile nodes are mutually independent. This assumption is supported by the results of [4], which showed that the encounters between a pair of mobile nodes follow a Poisson distribution, if the nodes move in a limited region (of size *S*) according to common mobility models (such as the random waypoint or the random direction model [5]), their transmission range (*r*) is small compared to *A*, and their speeds are sufficiently high. The authors further derived the following estimation of the pairwise encounter intensity λ: $\lambda \approx \frac{2\omega r E[V^*]}{S}$. ω is a constant specific to the mobility model. $E[V^*]$ is the average relative speed between two mobile nodes.

2.2 Queries and Answers

- We focus on the processing of a single spatial-existence query. Specifically, a query *Q* is issued by a node *Z* at time 0. *Z* is referred to as the *query node* and all the other *N* nodes are the *non-query nodes*.

- Starting from time 0, each non-query node produces answers to the query *Q* by a Poisson process with intensity μ. Before time 0, no non-query node produces any answer to the query *Q*. In other words, we assume that each non-query node enters into the region queried by *Q* by a Poisson process. If a node has ever produced an answer, i.e. entered the query region, then it keeps the latest produced answer which includes the time interval during which the node is/was in the query-region.

- All the answers have an equal size L_r in bytes. The size of the query *Q* is *Lq* in bytes.

- Denote by E_{Pr} the amount of energy consumed to communicate an answer between two encountering nodes. E_{Pr} includes the energy for transmission and that for receiving. Similarly, denote by E_{Pq} the amount of energy consumed for communicating a query between two encountering nodes. A constant amount of energy E_h is consumed at each encounter for the handshake. The handshake is used to determine which answers/queries are new to the encountered node and therefore should be transmitted (see subsection 2.1).

The notations are summarized below:

N—Number of non-query nodes in the system.
r—Transmission range of P2P communication.
λ—Encounter intensity between an arbitrary pair of nodes.
ω—Constant specific to the mobility model.
$E[V^*]$—Average relative speed between two mobile nodes.
S —Size of the area within which nodes move.
Z—Single node that issues the query.
Q—Query issued by *Z*.
μ—Selectivity ratio.
t_b —Backchannel transmission time.
E_{Pq} —Energy consumed for communicating a query between two encountering nodes in the P2P mode.
E_{pr} — Energy consumed for communicating an answer between two encountering nodes in the P2P mode.
E_h— Energy consumed for the handshake to determine whether an answer is new to the receiver.
L_q —Length of query in bytes.
L_r —Length of answer in bytes.

3. PROCESSING OF SPATIAL-EXISTENCE QUERIES IN P2P MODE

In subsection 3.1 we describe the in-network processing of spatial-existence queries in pure P2P mode. In subsection 3.2 we introduce the performance metrics used in our analysis. In subsection 3.3 we develop the analytical model. In subsection 3.4 we present the analysis results.

3.1 Pure P2P Query Processing Scheme

The pure P2P query processing scheme consists of two stages. At the first stage, called the *query-dissemination stage*, the query *Q* is disseminated via the P2P transmissions (see subsection 2.1), until either one of the following two events occurs: (i) a non-query node that has produced at least one answer is reached by *Q*; (ii) a non-query node that has been reached by *Q* produces an answer. Such a non-query node is referred to as the *return node*. At the time when the return node turns out, the second stage, called the *answer-return stage*, is initiated. In the answer-return stage, the answer is returned from the return node to the querying node *Z*, again via the P2P transmissions. We refer to this query processing scheme as Q_pA_p where Q_p indicates that the query is disseminated via P2P and A_p indicates that the answer is also disseminated via P2P.

3.2 Performance Metrics

The performance metrics we are interested in are:

1. Average delay: Average time it takes for the query node *Z* to receive an answer.
2. Average energy: Average amount of energy consumed totally in the system until *Z* receives an answer. The average energy consists of two components. The first component is the energy consumed for the dissemination of the query in the query-dissemination stage. The second component is the energy consumed for the dissemination of the answer in the answer-return stage. We do not consider the energy consumed for the dissemination of the query in the answer-return stage, even though the query may continue to be disseminated after the query-dissemination stage.

3. Energy efficiency: $\dfrac{1}{average\ delay \times average\ energy}$.

   Intuitively, if multiple queries are generated at node Z, one after another is answered, then $\dfrac{1}{average\ delay}$ represents the average number of queries that can be processed for Z per time unit (i.e., throughput). In this case, $\dfrac{1}{average\ delay \times average\ energy}$ represents the throughput enabled by each unit of energy consumption.

3.3 Analytical Model for the Q$_p$A$_p$ Scheme

3.3.1 *Average Delay of Q_pA_p*

First, we introduce a theorem that will serve as a basis for much of the analysis conducted in the rest of the paper.

Theorem 1: Let the variable $P(t)$ denote the probability that an arbitrary non-query node has the query Q at time t ($t \geq 0$). Then $P(t) = 1 - \dfrac{N}{N-1+e^{\lambda Nt}}$ □

Proof: Consider $P(t+\Delta t)$ which is the probability that at time $t+\Delta t$ an arbitrary non-query node O has Q. Let Δt be sufficiently small such that O encounters at most one node between t and $t+\Delta t$. $P(t+\Delta t)$ is the probability that one of the following mutually exclusive events happens:

1. O already has Q at time t. The probability of this event is $P(t)$.

2. O does not have Q at time t, and it encounters the query node Z between t and $t+\Delta t$. The probability of this event is $\lambda\Delta t$.

3. O does not have Q at time t, it encounters a non-query node between t and $t+\Delta t$ and this non-query node has Q at time t. The probability of this event is $\lambda\Delta t(N-1)P(t)$.

Thus,

$$P(t+\Delta t) = P(t) + (1-P(t))(\lambda\Delta t + \lambda\Delta t(N-1)P(t))$$

After simplification of the above difference equation, we get the following differential equation:

$$\frac{dp}{dt} = \lambda(1-P(t)(1+(N-1)P(t)) \qquad (1)$$

Since at time 0, none of the non-query nodes has Q, $P(0)=0$. Solving Eq. 1 with $P(0)=0$ yields

$$P(t) = 1 - \frac{N}{N-1+e^{\lambda Nt}} \qquad (2)$$

□

Now we compute the average delay of the Q$_p$A$_p$ scheme. The average delay of the Q$_p$A$_p$ scheme is the sum of the average delay of the query-dissemination stage and that of the answer-return stage. In the following we compute the average delays of these two stages respectively.

The average delay of the query-dissemination stage.

Denote by T_1 the delay of the query-dissemination stage. For any $t \geq 0$, denote by Prob($T_1 \leq t$) the probability that T_1 is smaller than or equal to t. According to the definition of the query-dissemination stage, Prob($T_1 \leq t$) is the probability that the return node turns out by time t. This is the probability that at least one non-query node has Q at time t and has produced at least one answer by t. The probability that a non-query node has Q at time t is $P(t)$. According to the Poisson process of answer production, the probability that a non-query node has produced at least one answer by time t is $1-e^{-\mu t}$. Therefore,

$$\begin{aligned}
&\text{Prob}(T_1 \leq t) = \text{Prob(at least one non - query node has } Q \text{ at } t \\
&\text{and has produced at least one answer by } t) \\
&= 1 - \text{Prob(no non - query node has } Q \text{ at } t \text{ and has produced} \\
&\text{at least one answer by } t) \\
&= 1-(1-P(t)\cdot(1-e^{-\mu t}))^N
\end{aligned}$$

Replacing $P(t)$ by Theorem 1 yields

$$\text{Prob}(T_1 \leq t) = 1-(1-(1-\frac{N}{N-1+e^{\lambda Nt}})(1-e^{-\mu t}))^N \qquad (3)$$

Since T_1 is a non-negative random variable, the expected value of T_1, denoted E[T_1], can be computed as

$$\begin{aligned}
\text{E}[T_1] &= \int_0^\infty (1 - \text{Prob}(T_1 \leq t))dt \\
&= \int_0^\infty (1-(1-\frac{N}{N-1+e^{\lambda Nt}})(1-e^{-\mu t}))^N dt \qquad (4)
\end{aligned}$$

The average delay of the answer-return stage.

Define T_2 to be the delay of the answer-return stage. Denote by t_0 the time at which the answer-return stage starts. Based on Theorem 1 and the symmetry between the query-dissemination stage and the answer-return stage,

$$\begin{aligned}
&\text{Prob}(T_2 \leq t) = \text{Prob}(Z \text{ has the answer at } t_0 + t) \\
&= P(t) \\
&= 1 - \frac{N}{N-1+e^{\lambda Nt}}
\end{aligned}$$

Since T_2 is a non-negative random variable, the expected value of T_2, denoted E[T_2], can be computed as

$$\begin{aligned}
\text{E}[T_2] &= \int_0^\infty (1 - \text{Prob}(T_2 \leq t))dt \\
&= \int_0^\infty \frac{N}{N-1+e^{\lambda Nt}} dt \qquad (5) \\
&= \frac{\ln N}{\lambda(N-1)} \qquad (6)
\end{aligned}$$

Thus, the average delay of the Q$_p$A$_p$ scheme, denoted *delay*(Q$_p$A$_p$), is

$$\begin{aligned}
&delay(Q_pA_p) = E[T_1] + E[T_2] \\
&= \int_0^\infty (1-(1-\frac{N}{N-1+e^{\lambda Nt}})(1-e^{-\mu t}))^N dt + \frac{\ln N}{\lambda(N-1)} \qquad (7)
\end{aligned}$$

3.3.2 Average Energy Consumption of Q_pA_p

The energy consumption of the Q_pA_p scheme consists of two components. The first component is the data communication energy, namely energy consumed for P2P communicating the answer and the query. The second component is the energy consumed for handshakes which determine whether the answer or the query is new to the receiver. In the following we compute the two energy components respectively.

The average energy for data communication.

First let us consider the average energy for data communication in the query-dissemination stage, denoted *EDQ*. Observe that during a P2P encounter, the query Q is transmitted only if it is new to the receiver. Thus, the number of times Q is communicated during the query-dissemination stage equals to the number of non-query nodes that have Q when the query-dissemination stage ends. Since the average delay of the query-dissemination stage is $E[T_1]$ (see Eq. (4)), and the probability that an arbitrary non-query node has Q at time $E[T_1]$ is $P(E[T_1])$, the average number of non-query nodes that have Q when the query-dissemination stage ends is $N{\cdot}P(E[T_1])$. Thus the average number of times Q is communicated during the query-dissemination stage is $N{\cdot}P(E[T_1])$. Since the energy consumed for each communication of Q is E_{Pq}, we have

$$EDQ = N \cdot P(E[T_1]) \cdot E_{Pq} \quad (8)$$

The average energy for data communication in the answer-return stage, denoted *EDA*, can be computed in the same way. We have

$$EDA = N \cdot P(E[T_2]) \cdot E_{Pr} \quad (9)$$

Thus, the average energy for data communication, denoted *ED*, is

$$\begin{aligned} ED &= EDQ + EDA \\ &= N \cdot \Big(E_{Pq} \cdot \big(1 - \frac{N}{N-1+e^{\lambda \cdot N \cdot E[T_1]}}\big) + E_{\mathrm{Pr}} \cdot \big(1 - \frac{N}{N-1+e^{\lambda \cdot N \cdot E[T_2]}}\big)\Big) \end{aligned} \quad (10)$$

The average energy for handshakes.

Since there is a handshake upon each encounter, the number of handshakes executed during the query processing equals to the number of encounters during the query processing. The average number of encounters during the query processing is easily computed to be $\frac{N(N-1)}{2} \cdot \lambda \cdot delay(Q_pA_p)$. Thus the average energy for handshakes, denoted *EH*, is

$$EH = \frac{N(N-1)}{2} \cdot \lambda \cdot delay(Q_pA_p) \cdot E_h \quad (11)$$

Thus, the average energy consumption of the Q_pA_p scheme, denoted *energy*(Q_pA_p), is

$$\begin{aligned} energy(Q_pA_p) &= ED + EH \\ &= N \cdot \Big(E_{Pq} \cdot \big(1 - \frac{N}{N-1+e^{\lambda \cdot N \cdot E[T_1]}}\big) + E_{\mathrm{Pr}} \cdot \big(1 - \frac{N}{N-1+e^{\lambda \cdot N \cdot E[T_2]}}\big)\Big) \\ &+ \frac{N(N-1)}{2} \cdot \lambda \cdot delay(Q_pA_p) \cdot E_h \end{aligned} \quad (12)$$

3.4 An Instantiation of the Analytical Model

3.4.1 Parameters Setting

We consider N+1 nodes moving within a 400m×400m terrain according to the random direction mobility model [5]. Each node chooses an initial direction, speed and travel time, and then travels in that direction with given speed for the chosen travel time. When the travel time expires, the node chooses a new direction, speed and travel time at random, independently of all previous directions, speeds and travel times. If a node hits the boundary of the terrain, it wraps around at the other side of the terrain. The node speed is chosen uniformly within the range Vmin=0.8km/hour and Vmax =2.4km/hour, a typical range of pedestrian speeds. The average relative speed is $E[V^*]=2.1$ according to the formula in [6]. The transmission range r is 30 meters. The pair-wise encounter intensity for this setting is computed to be $\lambda = 0.7875$ using the formula in [2]. This setting simulates, for example, soldiers in a digital battlefield.

For the energy consumption we use the model introduced in [7]. [7] describes a series of experiments which obtained detailed measurements of the energy consumption of an IEEE 802.11 wireless network interface operating in an ad hoc networking environment. According to [7], the energy overhead of P2P transmitting each byte of data is 1.9×10^{-6} Joule; the energy overhead of P2P receiving each byte of data is 0.5×10^{-6} Joule.

The system parameters and their values are listed in Table 1.

Table 1: System parameters and their values

Parameter	Description	Value
N	Number of nodes	10~100
r	Transmission range	30 m
S	Area within which nodes move	400m×400m
Vmin	Minimal velocity	0.8 km/h
Vmax	Maximal velocity	2.4 km/h
λ	Encounter intensity between an arbitrary pair of nodes	0.7875/h
t_b	Backchannel transmission time	1 minute
μ	answer production intensity	0.1~1 per hour
Lq	query size	100 bytes
Lr	answer size	100K bytes

3.4.2 Analysis Results

Figures 2 and 3 show the analysis results. Here are some interesting observations to make from the figures.

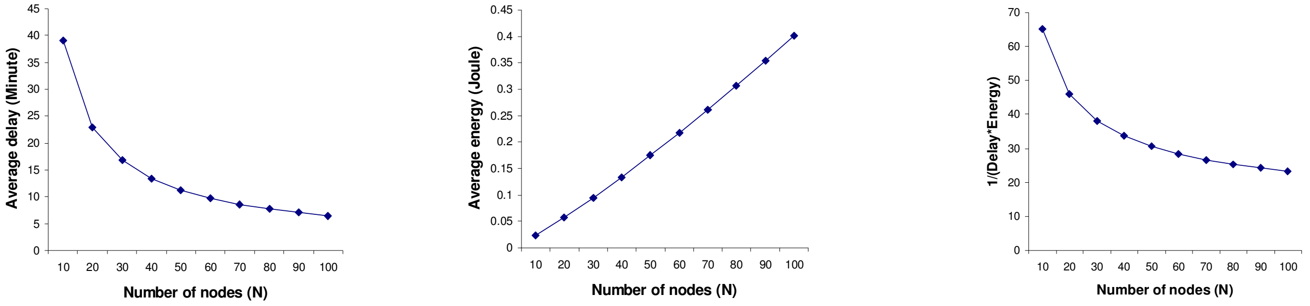
1. The delay decreases as the network density increases (see Figure 1(a)). Clearly, a higher network density results in more frequent interactions and thus fast propagation of the query. Observe that the decrease of the delay is not uniform. The delay decreases faster when the network density is low. Particularly, the delay is reduced by half when the network density increases from 10 to 20, whereas it is reduced by half again when the network density increases from 20 to 100.

2. The energy consumption increases as the number of nodes increases (see Figure 2(b)). Intuitively, when the number of nodes increases, two opposite effects are generated on the energy consumption. On the one hand, the query processing ends earlier (see Figure 2(a)), which would reduce the energy consumption. On the other hand, there are more encounters

and therefore more transmissions of reports and the query, which would increase the energy consumption. Figure 2(b) shows that the latter effect dominates.

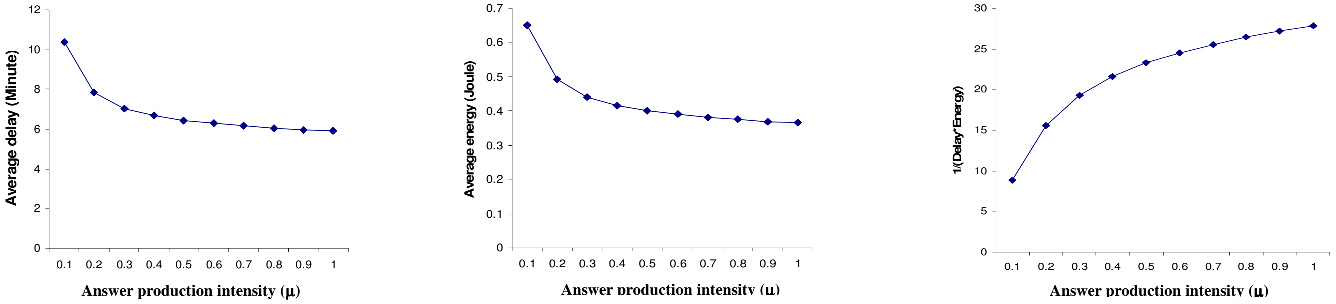
3. The energy efficiency decreases as the number of nodes increases (Figure 2(c)).

4. When the answer production intensity increases, e.g. when the query region increases in size, all the three performance metrics improve (Figures 3(a), (b), and (c)).



(a) Delay versus N (b) Energy versus N (c) Energy efficiency versus N

Figure 2. The performance of Q_pA_p as a function of the network density (μ=0.5)



(a) Delay versus μ (b) Energy versus μ (c) Energy efficiency versus μ

Figure 3. The performance of Q_pA_p as a function of the answer production intensity (N=100).

4. EXPLOITING BACKCHANNEL COMMUNICATION

In many environments, e.g. in an urban area or a digital battlefield, in addition to short-range P2P communication, mobile nodes can also communicate over long distances using an infrastructure such as a cellular or satellite network. We refer to such an infrastructure as a *backchannel*. In this section we analyze the in-network query processing schemes that exploit backchannel communication. In subsection 4.1 we describe the backchannel augmented query processing schemes. In subsection 4.2 we build analytical models for them. In subsection 4.3 we analyze them and compare with Q_pA_p.

4.1 Backchannel Augmented Schemes

From now on, in addition to the P2P mode, each mobile node is assumed to be able to communicate with every other node via the backchannel, regardless of their distance to each other. A backchannel transmission from one node to another takes a constant time t_b. Each node O can be receiving and transmitting simultaneously. When doing so, the node from which O is receiving can be the same as or different than the node to which O is transmitting. In other words, the node can be receiving from one node while transmitting to another. However, O cannot be receiving from or transmitting to more than one node simultaneously. Now we describe the backchannel augmented schemes.

Query via P2P Answer via Backchannel (Q_pA_b)

In the Q_pA_b scheme, the query Q is disseminated in the P2P mode, until a return node turns out. Upon receiving Q, the return node returns the answer to the querying node Z via the backchannel.

Query via Backchannel Answer via Backchannel (Q_bA_b)

In the Q_bA_b scheme, the query *Q* is sequentially transmitted by the querying node *Z* to every other node via the backchannel. A node that receives *Q* does not respond to *Z* if it does not have an answer. And *Z* does not wait for the response of the current receiver before it transmits *Q* to the next receiver. When the return node turns out, the return node returns the answer to *Z* via the backchannel.

Query via Backchannel Answer via P2P (Q_bA_p)

In the Q_bA_p scheme, the query *Q* is sequentially transmitted by the query node *Z* to every other node via the backchannel. A node that receives *Q* does not respond to *Z* if it does not have an answer. And *Z* does not wait for the response of the current receiver before it transmits *Q* to the next receiver. When the return node generates, the return node returns the answer to *Z* via the P2P dissemination.

4.2 Analytical Models for Q_pA_b, Q_bA_b, and Q_bA_p

The following notations are used in the analytical models.

E_{Bq} —Energy consumed for communicating a query via the backchannel.

E_{Br} —Energy consumed for communicating an answer via the backchannel.

t_b—Delay of a backchannel transmission.

Analytical Model for Q_pA_b

The average delay of the query-dissemination stage of Q_pA_b equals to E[T_1] (see Eq. 4). The delay of the answer-return stage of Q_pA_b is t_b. Thus the average delay of Q_pA_b is

$$delay(Q_pA_b) = E[T_1] + t_b \quad (13)$$

The average energy consumption of Q_pA_b is

$$e\text{nergy}(Q_pA_b) = EDQ + \frac{N(N-1)}{2} \cdot \lambda \cdot E[T_1] \cdot E_h + E_{Br} \quad (14)$$

where *EDQ* is computed by Eq. 8.

Analytical Model for Q_bA_b

First let us compute the average delay of the query-dissemination stage of Q_bA_b, denoted by T_3. Consider the probability distribution function Prob($T_3 \le t$). When $0 \le t < t_b$, Prob($T_3 \le t$)=0. In other words, since it takes t_b for the query *Q* to reach the first non-query node, it is impossible that the delay of the query-dissemination stage is smaller than t_b. When $t_b \le t < 2t_b$, Prob($T_3 \le t$) equals to the probability that the first receiver of *Q* produces at least one answer by time *t*. Thus, Prob($T_3 \le t$)=$1 - e^{-\mu t}$. When $2t_b \le t < 3t_b$, Prob($T_3 \le t$) equals to the probability that either the first or the second receiver of *Q* produces at least one answer by time *t*. Thus, Prob($T_3 \le t$)=$1 - e^{-2\mu t}$. Following this reasoning yields

$$\text{Prob}(T_3 \le t) = \begin{cases} 0 & 0 \le t < t_b \\ 1 - e^{-\mu t} & t_b \le t < 2t_b \\ \vdots & \vdots \\ 1 - e^{-(N-1)\mu t} & (N-1)t_b \le t < Nt_b \\ 1 - e^{-N\mu t} & Nt_b \le t \end{cases} \quad (15)$$

Prob($T_3 \le t$) is a mixed-type probability distribution (i.e., it is neither a discrete distribution nor a continuous distribution). The expectation of this distribution can be computed as follows. For $i = 1,2,...,N$, define

$$\begin{aligned} J_i &= \text{Prob}(T_3 \le i \cdot t_b) - \text{Prob}(T_3 \le (i \cdot t_b)^-) \\ &= (1 - e^{-i \cdot \mu \cdot i \cdot t_b}) - (1 - e^{-(i-1) \cdot \mu \cdot i \cdot t_b}) \\ &= e^{-(i-1) \cdot i \cdot \mu \cdot t_b} - e^{-i^2 \cdot \mu \cdot t_b} \end{aligned}$$

and

$$P_D = \sum_{i=1}^{N} J_i$$

Prob($T_3 \le t$) is decomposed into a discrete distribution X_D and a continuous distribution X_C, respectively defined as follows:

$$\text{Prob}(X_D = i \cdot t_b) = \frac{J_i}{\sum_{i=1}^{N} J_i} \qquad i = 1,2,...,N$$

$$\text{Prob}(X_C \le t) = \begin{cases} 0 & 0 \le t < t_b \\ \dfrac{1 - e^{-\mu t} - J_1}{1 - P_D} & t_b \le t < 2t_b \\ \vdots & \vdots \\ \dfrac{1 - e^{-(N-1)\mu t} - \sum_{i=1}^{N-1} J_i}{1 - P_D} & (N-1)t_b \le t < Nt_b \\ \dfrac{1 - e^{-N\mu t} - \sum_{i=1}^{N} J_i}{1 - P_D} & Nt_b \le t \end{cases}$$

Thus,

$$E[X_D] = \sum_{i=1}^{N} (\Pr(X_D = i \cdot t_b) \cdot i \cdot t_b) = \frac{1}{\sum_{i=1}^{N} J_i} \sum_{i=1}^{N} (J_i \cdot i \cdot t_b)$$

$$E[X_C] = \int_0^{\infty} (1 - \Pr ob(X_C \le t)) dt$$

$$E[T_3] = P_D \cdot E[X_D] + (1 - P_D) \cdot E[X_C] \quad (16)$$

Therefore the delay of Q_bA_b is

$$delay(Q_bA_b) = E(T_3) + t_b \quad (17)$$

The average energy consumption of Q_bA_b is

$$energy(Q_bA_b) = E_{Bq} \cdot \min(\left\lceil \frac{E[T_3]}{t_b} \right\rceil, N) + E_{Br} \quad (18)$$

where $\left\lceil \frac{E[T_3]}{t_b} \right\rceil$ represents the minimum integer that is greater than or equal to $\frac{E[T_3]}{t_b}$. $\min(\left\lceil \frac{E[T_3]}{t_b} \right\rceil, N)$ is the average number of backchannel query transmissions before the return node turns out.

Analytical Model for Q_bA_p

The average delay of Q_bA_p is

$$delay(Q_b A_p) = E[T_3] + E[T_2] \qquad (19)$$

where $E[T_3]$ and $E[T_2]$ are computed by Eq. 16 and Eq. 6 respectively.

The average energy consumption of the answer-return stage of Q_bA_p is $EDA + \frac{N(N-1)}{2} \cdot \lambda \cdot E[T_2] \cdot E_h$, where EDA is computed by Eq. 9. Thus, the average energy consumption of Q_bA_p

$$energy(Q_b A_p) = E_{Bq} \cdot \min(\left\lceil \frac{E[T_3]}{t_b} \right\rceil, N) + EDA + \frac{N(N-1)}{2} \lambda \cdot E[T_2] \cdot E_h \qquad (20)$$

4.3 Comparison of Q_pA_b, Q_bA_b, Q_bA_p, and Q_pA_p

We compare the schemes by numerical computation. In the following analysis, the energy overhead of backchannel transmitting each byte of data is set to be 32×10^{-6} Joule; the energy overhead of backchannel transmitting each byte of data is 44×10^{-6} Joule. The delay a backchannel transmission (i.e., t_b) is 1 minute. The answer size (*Lr*) is either 100 bytes or 100K bytes.

Figures 4-7 show the comparison results. Figure 4 and Figure 5 show the performance as a function of the network density. They differ in the answer size. Figure 6 and Figure 7 show the performance as a function of the answer production intensity μ. They differ in the answer size. The following observations can be made:

1. Out of the four compared schemes, there does not exist one that outperforms all the other three in either delay or energy consumption for all the cases. The optimal scheme depends on the system parameters and on the performance metric. For example, in Figure 4, when N=10, in terms of delay, Q_bA_p is optimal (see Figure 4(a)). In terms of energy, Q_pA_p is optimal (see Figure 4(b)). When N=30, Q_pA_b becomes optimal in delay.

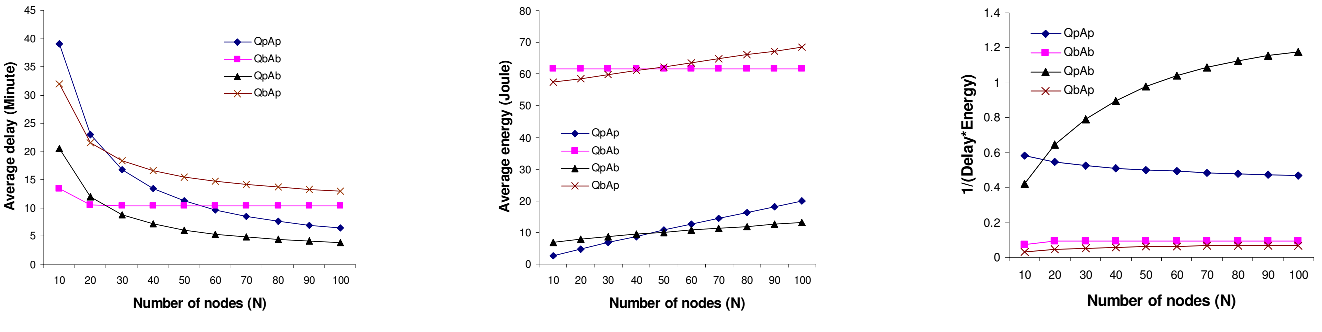
   The analytical models developed in this paper enables users to determine the optimal scheme for a given parameter configuration and a given performance metric. And this is the significance of our results.

2. Q_pA_p, Q_pA_b, Q_bA_b are feasible schemes, in the sense that for each of these three scheme, there exists a parameter configuration such that the scheme is optimal for at least one performance metric, either the delay, or the energy consumption, or the energy efficiency. For example, Q_pA_p is optimal for energy consumption when N=10 in Figure 4(b). Q_pA_b is optimal for delay when N>30 in Figure 4(a). Q_bA_b is optimal for energy efficiency when μ>10 in Figure 7(c).

3. Q_bA_p is not a feasible scheme in this parameter configuration, because it is not optimal for any of the three performance metrics.

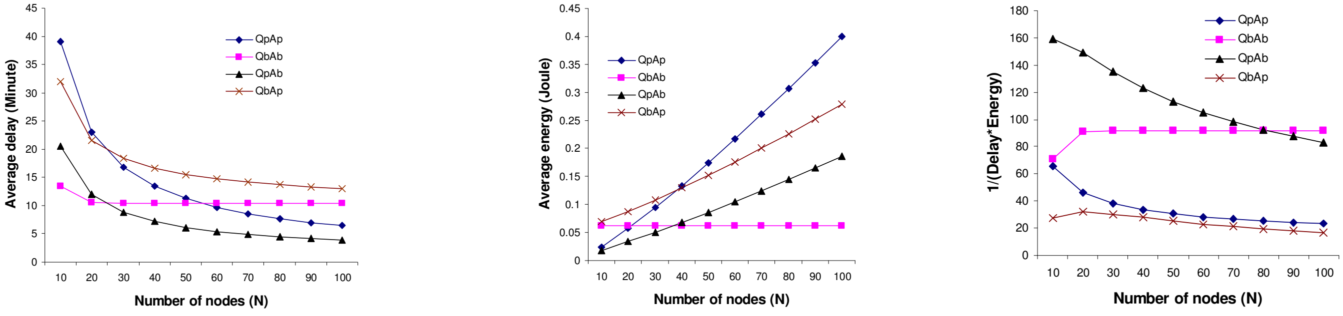
4. The backchannel, if appropriately utilized, helps reducing the delay. In fact, in all the cases, it is a backchannel augmented scheme (either Q_pA_b or Q_bA_b) that has the lowest delay (see Figures 4(a), 5(a), 6(a), 7(a)). However, when the network density increases, the advantage of the backchannel on delay decreases (see Figures 4(a) and 5(a)).

5. The backchannel schemes do not necessarily incur higher energy consumption, even though the energy consumed by the backchannel to transmit one byte of data is 32 times higher than that consumed by the P2P (see Figures 4(b), 5(b), 6(b), and 7(b)).



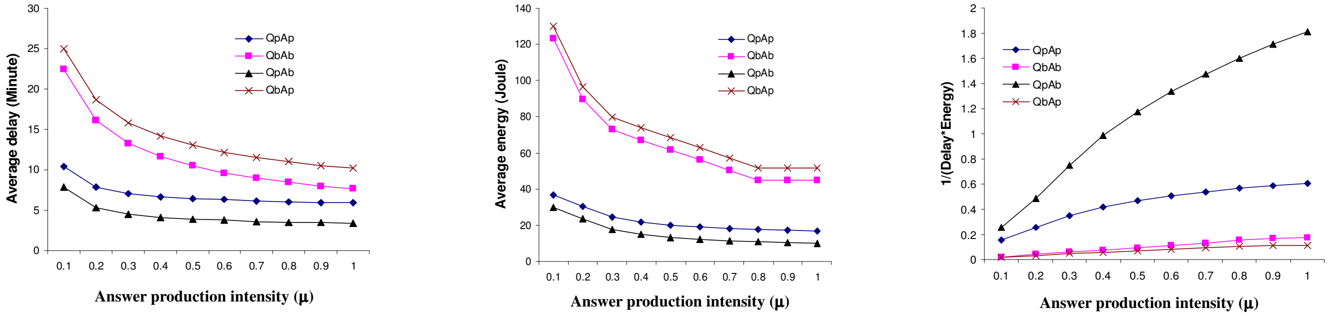
(a) Delay versus *N* (b) Energy versus *N* (c) Energy efficiency versus *N*

Figure 4. The performance of the compared schemes as a function of the network density (μ=0.5, *Lr*=100K bytes)



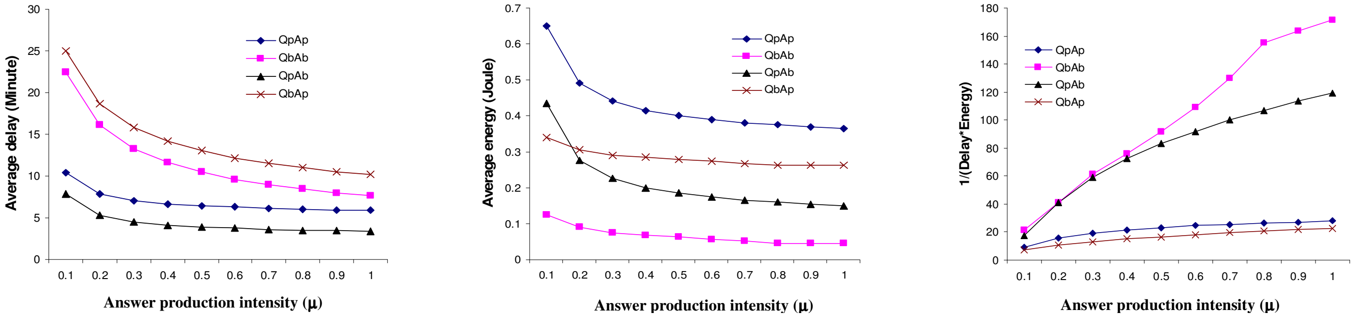
(a) Delay versus *N* (b) Energy versus *N* (c) Energy efficiency versus *N*

Figure 5. The performance of the compared schemes as a function of the network density (μ=0.5, *Lr*=100 bytes)



(a) Delay versus μ (b) Energy versus μ (c) Energy efficiency versus μ

Figure 6. The performance of the compared schemes as a function of the answer production intensity (*N*=100, *Lr*=100K bytes)



(a) Delay versus μ (b) Energy versus μ (c) Energy efficiency versus μ

Figure 7. The performance of the compared schemes as a function of the answer production intensity (*N*=100, *Lr*=100 bytes)

5. RELEVANT WORK

Static-sensor databases. A database approach has been applied to static sensor networks (see e.g., [11]). Most of these methods require that a certain routing structure, e.g. a tree, be established in the network. It has been shown in [12] that such a method is unsuitable for a dynamic network topology because the routing structure is hard to maintain.

Resource discovery (e.g. [13]) and publish/subscribe (e.g. [14]) in MANET's. These papers often build a routing structure for resource information dissemination. Consequently they can be inefficient, particularly in networks that are prone to frequent topology changes and disconnections due to mobility and turnover. In such an environment, either a lot of communication has to be expended to keep the routing structure up to date, or the routing structure rapidly becomes obsolete and misses many matches. Furthermore, these methods depend on network connectivity, and do not work in sparse networks.

Epidemic routing. [8] develops an analytical model to compute the expected delay of delivering a message from a source to a destination via epidemic routing. Using a Markov model, [8] obtained the same result for the probability that we computed in Eq. 2 (see subsection 3.3.1). We note that the development is much simpler using our ODE model.

[9] develops an epidemic model for information diffusion in MANET's. Our Eq.2 is consistent with the result of [9]. However, we use a probabilistic approach whereas [9] uses a deterministic approach.

In addition, we deal with the combination of P2P and infrastructure, whereas neither [8] nor [9] does so.

6. CONCLUSION

In this paper we proposed replacing the NN and range query in mobile disconnected networks by a different type of query, namely spatial-existence query. We developed analytical models to compute the average delay and the average energy consumption for four spatial-existence query processing schemes, three of which exploit the communication infrastructure. Using these analytical models we compared the four schemes in terms of delay, energy consumption, and the combination of the two, under various environmental conditions. It is found that none of the four schemes is optimal in any individual performance metric for all the conditions. Therefore the analytical models enable users to choose the optimal scheme for a given environmental configuration.

7. REFERENCES

[1] M. Demirbas and H. Ferhatosmanoglu. Peer-to-peer spatial queries in sensor networks. In Proc. of ICP2PC, 2003.

[2] Y. Xu, W. Lee, J. Xu, and G. Mitchell. Processing window queries in wireless sensor networks. In Proc. of *ICDE*, 2006.

[3] Wu, S.-H., Chuang, K.-T., Chen, C.-M., Chen, M.-S.: Diknn: an itinerary-based knn query processing algorithm for mobile sensor networks. *ICDE'07*, Istanbul, April 2007.

[4] R. Groenevelt, P. Nain, G. Koole, The message delay in mobile ad hoc networks, Elsevier Journal of Performance Evaluation (2005).

[5] C. Bettstetter, Mobility modeling in wireless networks: categorization, smooth movement, and border effects, in: ACM SIGMOBILE Mobile Computing and Communications Review, vol. 5, Issue 3, July 2001.

[6] Groenevelt, R. Stochastic Models for Ad Hoc Networks. PhD thesis, INRIA, April 2005.

[7] Feeney, L., and Nilsson, M. Investigating the Energy Consumption of a Wireless Network Interface in an Ad Hoc Networking Environment. In Proceedings of IEEE INFOCOM (Anchorage, AK, 2001).

[8] X. Zhang, G. Neglia, J. Kurose, D. Towsley. Performance modeling of epidemic routing. Computer Networks 51(2007) 2867-2891.

[9] A. Khelil, C. Becker, J. Tian, K. Rothermel. An Epidemic Model for Information Diffusion in MANETs, MSWiM02, pages 54-60, Atlanta, Georgia, September, 2002.

[10] O. Wolfson, B. Xu, R. Tanner. Mobile Peer-to-Peer Data Dissemination with Resource Constraints, *Proc. of the 8th International Conference on Mobile Data Management, Mannheim, Germany*, 2006.

[11] S. R. Madden, M. J. Franklin, J. M. Hellerstein, and W. Hong. TinyDB: an Acquisitional Query Processing System for Sensor Networks. *ACM TODS*, 30(1):122–173, 2005.

[12] Ouri Wolfson, et al. Search-and-Discover in Mobile P2P Network Databases. ICDCS, 2006.

[13] S. Das et al. Ekta: An efficient dht substrate for distributed applications in mobile ad hoc networks. *WMCSA 2004*.

[14] Y. Huang, H. Garcia-Molina. Publish/Subscribe Tree Construction in Wireless Ad-Hoc Networks. *MDM'03*.

[15] T. Zhong, B. Xu, O. Wolfson. Disseminating Real-Time Traffic Information in Vehicular Ad-Hoc Networks. Proc. of the IEEE Intelligent Vehicles Symposium, 2008.

[16] F. of American Scientists. Remote battlefield sensor system (rembass). http://www.fas.org, 2000.

[17] http://www.princeton.edu/~mrm/zebranet.html