Wireless Mobile Ad-hoc Sensor Networks for Very Large Scale Cattle Monitoring

Milena Radenkovic   Bartosz Wietrzyk
School of Computer Science and IT
University of Nottingham
Nottingham, NG8 1BB, UK
{mvr, bzw}@cs.nott.ac.uk

Abstract— **This paper investigates the use of wireless mobile ad hoc sensor networks in the nationwide cattle monitoring systems. This problem is essential for monitoring general animal health and detecting outbreaks of animal diseases that can be a serious threat for the national cattle industry and human health.**

We begin by describing a number of related approaches for supporting animal monitoring applications and identify a comprehensive set of requirements that guides our approach. We then propose a novel infrastructure-less, self organized peer to peer architecture that fulfils these requirements. The core of our work is the novel data storage and routing protocol for large scale, highly mobile ad hoc sensor networks that is based on the Distributed Hash Table (DHT) substrate that we optimize for disconnections. We show over a range of extensive simulations that by exploiting nodes' mobility, packet overhearing and proactive caching we significantly improve availability of sensor data in these extreme conditions.

I. Introduction

THE production of beef and milk has always been a major constituent of agriculture in the United Kingdom. The total number of cattle in England in 2003 was estimated to be over 5,709,000. The beef and veal consumption in 2002 was on average 19.7kg per capita in UK and 19.6kg in EU [1].

In the last two decades, two major animal diseases outbreaks have occurred in the UK. The outbreak of bovine spongiform encephalography (BSE), or 'mad cow disease', which began in the United Kingdom in 1986, has infected about 200 thousand cattle and has led to pre-emptive slaughter and destruction of 4.5 million cattle. The foot-and-mouth disease (FMD) epidemic in 2001 led to the eradication of 4 million livestock [2]. FMD is a highly contagious viral disease that primarily afflicts cloven-hoofed animals. It can be debilitating to infected animals, causing losses in meat and milk production.

Nationwide cattle monitoring [2] has been identified as an important means of detecting outbreaks of these diseases and is potentially able to save large amounts of money. Currently, in UK cattle births and movements are reported manually by farmers over the web interface, e-mail, or post and stored in a central database containing currently entries of over 5 millions animals [3]. This data is very general and does not cover for instance the information about the proximity of animals that can lead to contamination. In order to get more detailed information about the potential spread of diseases it is necessary to use automated system monitoring location and proximity of the animals. The farm level monitoring systems that are currently in operation are not sufficient because they are voluntary, highly heterogeneous and poorly integrated [2]. The cattle can also be contaminated when they are transported between pastures or handed between farmers, which takes place outside the scope of farm monitoring systems.

A nationwide cattle monitoring system can also serve other purposes i.e. indicate general health issues associated with animals [4] such as lameness, mastitis and other infection or metabolic diseases.

The contributions of this paper are two fold and comprise: (1) a novel scalable, self-organized architecture for a large scale animal monitoring utilizing mobile ad hoc networks (MANETs), Delay Tolerant Networks (DTNs) [5, 6] and a peer to peer Internet overlay (2) a protocol for proactive caching addressing disconnections in MANETs.

The paper is organized in the following way. Section II reports on the related existing work. Section III defines the requirements and challenges for the national monitoring system. Section IV gives a high level overview of the proposed architecture. Section V presents in detail our novel distributed data storage and routing protocol for large number of mobile ad hoc sensor networks that is based on Distributed Hash Tables (DHTs). Section VI describes the simulation methodology and discusses our results of the protocol evaluation. Finally Section VII gives conclusions and discusses future work.

II. Related Work

This section evaluates current approaches to the animal monitoring. There is a number of existing approaches concerning animal monitoring. None of them satisfies our requirements but we briefly evaluate them in order to better understand our challenges.

Stationary wireless sensor networks for wildlife monitoring that are typically deployed today have small to medium scale (tens to hundreds of sensors), span small to medium geographical distances (tens to hundreds of square miles) and are short to medium term (hours to months). The sensor data is typically archived in a powerful server geographically collocated with the sensors that is usually fully replicated on the pre-determined powerful servers in the labs e.g. The Great Duck Island [7]. An architecture for the long term, large scale self organizing sensor network for monitoring Brazilian rain forest has been proposed [8] but, to the best of our knowledge, not deployed. Our work differs from the stationary wireless sensor networks in terms of mobility of the nodes and their capabilities. In particular our sensor nodes have larger storage space, greater processing power and weaker energy constrains.

Mobile wireless sensor networks for animal monitoring deployed today have typically small scale (up to 100 nodes) e.g. ZebraNet project [9] for monitoring zebras at the Mpala Research Centre in Kenya. In ZebraNet, devices mounted on zebras were transferring all their measurements to all other devices in their range. This approach was not scalable due to limited storage space of the devices and was feasible only because of the small number of monitored animals. The retrieval of the aggregated measurements from animals required approaching them by humans, which increased the maintenance costs. Another approach to retrieving data from the animal mounted sensors utilizes GSM telephony [10]. Such collars are already available on the market [11]. In case of monitoring of large number of animals this approach can be financially challenging due to the cost of GSM communication, i.e. the maintenance costs. Another disadvantage is that GSM transceivers consume large amounts of energy. That can potentially lead to a considerable effort necessary to replace the batteries every few days. These limitations were addressed [10] by putting GSM, in particular GPRS transceivers only on a subset of animals. The devices without GPRS transceivers transfer their measurements through the GPRS-enabled devices. Our work differs from the typical mobile wireless sensor network for animal monitoring in terms of much larger scale (millions of devices rather than tens).

RFIDs are an established approach for tracking domestic animals [12]. They allow reading an animal identifier by a fixed reader that provides information that a given animal has been present at certain time at a given location. The advantages of this approach include: (1) low price, (2) small size – can be mounted in an animal as an implant, (3) long life due to lack of need for energy supply. The disadvantages include limited scope of provided information and unreliable RFID readings (especially in cases when animals move too fast or too many of them happen to be present near the reader).

Archival tags are monitoring devices that can be fitted on animals, can store relevant measurements for the animal and can be read by a handheld device [13]. This is the most reliable approach but expensive in terms of maintenance cost because the animals must be individually approached by stockmen. The scope of monitored factors is potentially large. The cost of deployment is high due to the price of monitoring devices.

In the case of *radio tags,* the measurements from monitoring devices mounted on animals can be read over radio communication. This approach is often used in wildlife monitoring [14]. The ready made collars and receivers are produced by various companies [11, 15-18] and available for ordering. The important limitation of this approach is that the receiver's aerial must be mounted in the place visited by all the animals. Finding such a place can be difficult in case of beef cattle fed by continuous grazing, i.e. roaming on a large meadow without interacting with humans. Then the data from the receiver must be transported to a server, which is not always an easy to automate task and may cause high maintenance costs.

There has been research done in *controlling animals* [19], which involved mounting devices on animals. These devices were producing sounds whenever an animal tried to leave a virtual paddock. The referenced paper concentrates on the automatic control aspect of the proposed application rather than on the utilized wireless communication.

In general most of current approaches to animal monitoring either do not address the large scale, low cost deployment or make simplifications that affect realism of their studies. We aim at improving this realism by deeper understanding of the cattle monitoring application.

III. Requirements and Challenges

In this section, we identify the requirements for a nationwide cattle monitoring system. The cattle can be contaminated not only in the farms buildings or on the pastures but also when they are transported between farms. Therefore, it is necessary to monitor the animals throughout their life regardless of their current owner. A potential user should be able to remotely query data about a particular animal identified by its ear tag or all animals from a particular pasture or enterprise. This data should comprise any detected animal diseases, ear tag ids of other animals which were in proximity of the queried animal and timestamps of their last proximity.

The monitoring equipment should be mounted on an animal all its life and store all the measurements to allow successful detection of abnormal behaviour of the animal suggesting a disease and evaluation of the products acquired from the animal. The major challenges include achieving:

Decreased maintenance and deployment costs – the nationwide farm monitoring system is only feasible when its costs are kept low. This includes costs of hardware, involvement of workforce in maintenance and deployment, particularly the workforce with certain technical skills and costs of the third party communication services such as GPRS connectivity.

Increased Reliability – the major objective of this system is providing users with relevant data. Therefore probability of successful retrieval of measurements in case of disconnections should be maximized.

Impact on the animals - the animal mounted devices should not cause any excessive annoyance to the animals in terms of their construction, weight and placement. Otherwise the

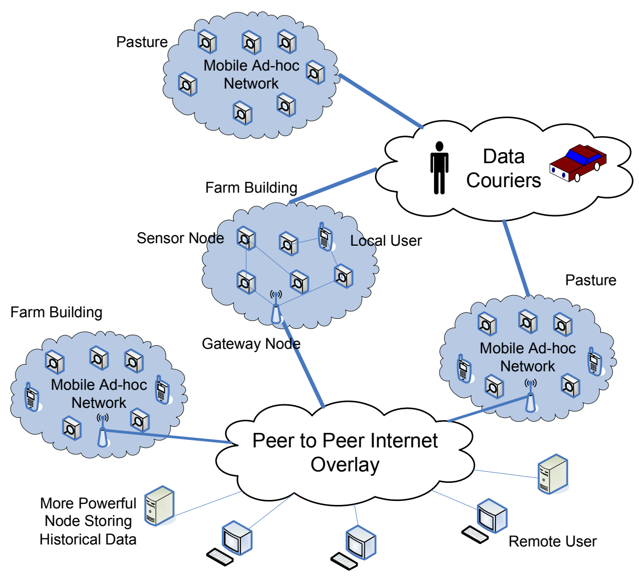

Fig. 1. Nationwide Peer to Peer Monitoring Architecture

animals will be given an incentive to destroy the devices.

Security - the described monitoring system should be tamper proof to prevent dishonest farmers or traders from altering the measurements.

IV. National Peer to Peer Monitoring Architecture

This section gives a high level overview of the novel self-organised architecture for nationwide cattle monitoring system that has no centralised component, utilises commodity equipment and has very low deployment and administration costs. This architecture aims to provide real time data about animals.

The proposed architecture is shown in Fig. 1. It comprises multiple mobile ad hoc sensor networks that contain mobile sensor nodes that can measure, store and cache measured and queried data. This architecture has no centralised storage or processing facilities. The processing of the collected data is performed by sensor nodes.

The animal mounted sensor devices monitor and store information related to animals' health and location. They are typically mounted on animals as collars [9, 20] and in the basic form they have only a microcontroller processing the measurements, a radio transceiver, flash memory for storing the measurements and a solar power source [9, 10]. These devices can be easily extended with pluggable modules providing additional means of communication such as GPRS [10] or additional sensors that proved to be appropriate for animal monitoring such as GPS tracking animal's position [9, 10, 12], a pedometer monitoring its activity level [10, 12, 20], intra-rumenal sensors monitoring its physiology [10] or accelerometers monitoring grazing or mating behaviour [10]. The devices are capable of processing the collected data in order to detect animal diseases.

The *state* of the animal comprises detected events such as diseases, ear tag ids of other approached animals, times stamps of these meetings etc. The most recent *state* of every animal is cached on a number of other animal mounted nodes in a self-organized, network-aware manner that is described in section V. By exploiting nodes' mobility, packet overhearing and self-organization we significantly improve availability of data in conditions of high mobility and disconnections as shown in section VI.

The users of the proposed system (e.g. stockmen, farmers, slaughter workers, people involved in the animal trade, veterinaries) issue queries concerning a particular animal, animals on a certain area or animals that were met by a particular animal. We assume that the users can be mobile and directly query the devices mounted on animals over the wireless communication. Data can be also accessed by office users over Internet. The mobile nodes carried by people or vehicles have a microcontroller, power sources, wireless transceiver, user interface and GPS. They can be used as client devices to access the measurements.

Each mobile sensor device can participate in the routing and forwarding of the packets. If two nodes cannot communicate directly, intermediate nodes aid in forwarding packets between them. The communication between the islands of mobile ad hoc sensor networks can be performed over Internet or using people or vehicles as data couriers. Individual mobile ad hoc sensor networks can be connected to the Internet by gateway nodes (e.g. supplied with GPRS modules or connected to DSL modems). Any of the animal mounted nodes can become a gateway node when it is supported with a GPRS module. The utilization of data couriers can provide interconnectivity with remote and disconnected regions or can decrease the GPRS communication costs. The gateway nodes and users accessing the system over Internet (further called remote users) are connected using peer-to-peer overlay hierarchical protocols on which we are currently working.

More powerful nodes connected to Internet can store historical states and measurements. These nodes acquire data by querying a given subset of the topology. In particular they query gateways which flood the query to the nodes within their islands of connectivity.

A query coming from a remote user is routed by the peer to peer overlay to a more powerful node storing states and measurements if it is available. Otherwise the query is routed to a relevant gateway node on the pasture. If a queried sensor node is within the island of connectivity of a gateway node a remote user can query all the historical measurements and results of processing stored by the sensor device. Otherwise the only available data is the most recent *state* of the queried animal cached by sensor nodes within the gateway node's island of connectivity.

We assume that a user located on a pasture (further called local user) will most frequently issue queries related to sensor devices on the pasture but if the user's device is within the island of connectivity of a gateway node, the user has access to the same data as a remote user. Otherwise she has access only to: (1) full historical measurements and results of processing of

Node Id 10233102

Leaf set	SMALLER	LARGER	
10233033	10233021	10233120	10233122
10233001	10233000	10233230	10233232

Routing table			
-0-2212102	**1**	-2-2301203	-3-1203203
0	1-1-301233	1-2-230203	1-3-021022
10-0-31203	10-1-32102	**2**	10-3-23302
102-0-0230	102-1-1302	102-2-2302	**3**
1023-0-322	1023-1-000	1023-2-121	**3**
10233-0-01	**1**	10233-2-32	
0		102331-2-0	
		2	

Fig. 2. Leaf set and routing table of a hypothetical node with the id 10233102

devices within her island of connectivity, (2) most recent *state* of animals from the pasture where she is located or a set of pastures interconnected with data couriers. In the latter case the *state* is cached by sensor devices within the user's island of connectivity.

To address tamper resistance problem, we use the Public Key Infrastructure mechanisms by digitally signing the query responses and disseminated data.

V. Distributed Data Storage and Routing

This section proposes a novel storage and routing protocol that supports our architecture. In the proposed architecture this protocol organizes wireless communication between sensor nodes, gateways and users on the pasture accessing directly sensor nodes. The scope of this protocol is a pasture or a set of pastures interconnected with data couriers. The gateway nodes and users accessing the system over Internet are connected using peer-to-peer overlay hierarchical protocols on which we are currently working.

The problem of accessing measurements collected by nodes and the disseminated data can be generalized as a distributed storage of key-value pairs and addressed using Distributed Hash Tables (DHTs). A DHT is a scaleable data structure for distributed, self-organized storing of pairs (key, data) which allows fast locating of data when a key is given [21]. DHTs can be made robust in the face of failures, attacks and unexpectedly high loads. DHTs were repeatedly shown to have excellent performance for efficient look-up for a single key, range keys and can also be network aware. In contrast with the wired Internet counterparts, DHTs for wireless ad-hoc applications are still in their infancy.

Existing research [21, 22] has shown that for any mobile ad hoc network (MANET) environment having application-level DHT on the top of existing routing protocols (DSR [23], AODV [24]) is less efficient than tight integration of DHT with these protocols (i.e. DHT substrate) in terms of percentage of successfully answered queries and network overheads. The existing DHT substrates do not address the problem of high disconnection rate and their percentage of successfully answered queries (i.e. success ratio) drop significantly with increased mobility e.g. 70% for the node mobility of only 5m/s [22].

We propose a novel DHT substrate called Artemis that extends existing MANET DHT substrates (e.g. Ekta [21]) with our technique of the self-organized, network-aware caching mechanism and show that it is possible to increase success ratio in conditions of high mobility and disconnections.

A. *Routing*

Each node has a unique 128-bit id and every routed message has an associated 128-bit key. The objective is to route the message to a node with an id numerically closest to the key associated with the message. In our scenario the ids are generated by hashing the node id (e.g. animal, vehicle or person). If necessary we can provide different id spaces for all the classes of nodes, e.g. we can prefix gateway node ids and user ids with 1 and 2 respectively.

Similarly to Ekta [21], in our network comprising N nodes, a message can be routed to any node in less than $\log_{2^b} N$ steps on average. As in Ekta and Pastry [25] *b* is a configuration parameter responsible for the number of rows, columns in the routing table and maximal number of routing steps. In Pastry and Ekta it is chosen to be 4 as a compromise between the size of the routing table and the maximal length of the paths. We follow that compromise.

Every node has the leaf set and the routing table. They are used for routing messages and are set of node ids with associated multi-hop source routes. The routing table is a two dimensional array having $\lceil \log_{2^b} N \rceil$ rows and $(2^b - 1)$ columns. At a row *n* are placed ids sharing first *n* and only *n* digits with the present node's id. The *n*+1 digit of an id in the routing table defines the column where it is stored. Each entry in the routing table can store more than one id. The leaf set contains ids numerically closest to the id of the present node: *L*/2 greater and *L*/2 smaller, where *L* is a configuration parameter. The size of the leaf set (*L*) is chosen to be 16.

The leaf set and the routing table of an example node with the id 10233102 is shown in Fig. 2 (adapted from [25]). For simplicity, *b*=2 and *L*=8. All numbers are in base 4. The top row of the routing table is row zero. The shaded cell in each row of the routing table shows the corresponding digit of the present node's id. The ids in each entry have been split to show the *common prefix with 10233102 - next digit – rest of node id*. The associated source routes are not shown.

The routing table and the leaf set are used to find a next node on the route of the message being forwarded. Such a node should share with the message's key more digits than the id of the forwarding node or be at least numerically closer to the key than the id of the forwarding node. The nodes located closer to the forwarding node (in the sense of network distance) are preferred. Unlike Ekta we also give priority to the freshest routes from the shortest ones. If a lookup to the leaf set and the routing table returns a node to which the route is not known the node performs flooding based route discovery. The flooding is controlled in a similar manner as proposed in the DSR MANET routing protocol [23]. The source routes are collected either by the flooding based route discovery, or acquired from the forwarded or overheard packets. Whenever a node fails to forward a packet along the assigned source route, it sends back the *RouteFailure* message, so all the nodes that forwarded the packet can drop the broken route.

When a new node *a* wants to join the network, it starts with generating its own id. It routes then a *Join* message to a node *b,* which has an id numerically closest to *a*. Node *b* sends to the node *a* the *JoinComplete* message containing the *b*'s leaf set. The node *b* then notifies all the nodes from its leaf set about the arrival of the node *a*.

B. Self-organized Caching

We can generalize the notion of data disseminated by a node as its *state*, which is the status of the monitored animal related to its health and location such as ear tag id, detected diseases, ear tag ids of other approached animals, times stamps of these meetings etc. Every node stores its own state and up to *k* states acquired from other nodes, where *k* is a system parameter depending on the hardware abilities of the nodes. Every state is associated with an id of its producer and a timestamp describing when the state was sent by its producer. The timestamp of the local state is always equal to the current time. The priority is given to store states that have ids numerically closest to the local node. There are two reasons for that. Firstly, the nodes with numerically closest ids are preferred in routing so such strategy increases the probability that a query encounters a replica before the original. Secondly, that potentially improves distribution of replicas across the topology.

The passive caching is performed as follows. Every node overhearing, forwarding or receiving any packet containing states store these states if they have ids numerically closer to the node's id than the states currently stored or are newer than the currently stored states with the same ids.

The idea behind proactive caching is that single hop neighbours exchange states they hold according to their storage strategy, i.e. giving priority to freshest states of nodes with ids numerically closest to the own id. The proactive caching of state has two reasons: (1) improving dissemination of states in the condition of lack of queries, (2) exploiting mobility of nodes (i.e. a state is transported by a node to a different physical location) and (3) exploring temporary availability of nodes (when two nodes meet for a short period of time they exchange their states).

The important disadvantage of the proactive dissemination is that it affects the network traffic caused by querying which limits the success ratio. It is then necessary to keep the graceful degradation of success ratio to minimum by performing the proactive caching only when there is no query traffic.

We perform proactive caching in the receiver driven way. If there is no network traffic for the time t_1, a node *A* sends a beam packet i.e. a broadcast sent to all nodes within its range, which are not supposed to forward it further. Note that it has been considered more energy efficient to send data over smaller number of long hops rather than over a larger number of short hops [26]. The time t_1 is calculated using a following heuristic formula:

$$t_1 = T_1 + T_2 * S + T_3 * j$$
$$T_1 > T_2 > T_3$$

T_1, T_2 and T_3 are the system parameters selected experimentally according to the node's bandwidth. S is the number of states stored currently by the node and j is a random value ranging from 0 to 1. t_1 is proportional to the number of states a node caches so nodes with lower number of states (e.g. these which joined the network recently) are able to acquire states quicker. The random value j is introduced to prevent multiple nodes from trying to send data at the same time.

The beam packet contains the sender's id and a list of ids of states the sender stores together with their timestamps. All the nodes that receive the beam packet decide if they have states that the node *A* would like to store according to its storage strategy i.e. giving priority to freshest states of nodes with ids numerically closest to the own id. If this is the case, they wait time t_2 and send these states to the node *A*. If any of them has a state already stored by the node *A* but with the timestamp greater by t_{max}, it is also scheduled for sending to the node *A*. Time t_2 is calculated with the following heuristic formula:

$$t_2 = \frac{T_2}{n} + T_3 * j$$
$$T_2 > T_3$$

Where T_2 and T_3 are the system parameters selected experimentally according to the node's bandwidth, n is the number of states the node is going to send; j is a random value from 0 to 1. t_2 is reversely proportional to the number of states a node is going to send so the packets containing larger number of states are sent sooner and these containing smaller number of states might be never sent because of overlapping with packets sent earlier. Note that sending the same data in one packet instead of several produces lower network overhead. The random value j is introduced to prevent multiple nodes from trying to send data at the same time.

All the nodes that overhear the states sent to the node *A* adjust accordingly the states that they are going to send to the node *A* and store the overheard states according to their storage strategy. If nodes send, forward or overhear any traffic not connected with proactive caching between receiving the beam packet and replying to it, they do not reply. They also do not reply if the reply would have to be sent sooner than $t_{min\ gap}$ after any previous answer sent by the node. That limits the extent to which the query traffic is affected by proactive caching. When a node receives or sends a beam packet or replays to a beam packet, it delays sending its beam packet by t_1.

Note that care has been taken to make querying and proactive caching temporarily and locally exclusive. With such an approach, we limit the graceful degradation of success ratio of queries to minimum. On the other hand, during periods of querying proactive caching is not essential, as the caches are populated/refreshed from overheard or forwarded answers to queries. To perform passive and proactive caching no global knowledge about the network is necessary.

Caching and replication are often used in DHT based storage systems dedicated to wired networks. In particular our strategy of giving priority to storage of keys with ids numerically closest to the id of the local node is informed by the PAST storage system [27] based on the Pastry DHT [25]. Pastry and PAST differ from our work because: (1) they operate on the application level and contrary to our approach they do not address network layer routing, (2) they target wired networks not addressing dynamic changes in network topology and disconnections, (3) the storage strategy we utilise is borrowed from PAST but the protocol for proactive caching is our contribution.

Caching has been also used in the mobile disconnection-affected systems with the service oriented architecture. The most common approach is that a client caches the state or responses of the service [28, 29]. Another approach is to support client caching with the dedicated intermediaries [30, 31]. Most of these approaches neither address self organization nor give details about peer-to-peer interactions.

VI. Evaluation

A. Simulation Methodology

This section describes how we evaluate our data storage and routing protocol within the scope of a single pasture. We simulate 100 nodes on the square area of 1 km^2 for the duration of 1200 simulated seconds using the established network simulator ns-2 [32]. Nodes move with the random velocities between 0.1 m/s and 2 m/s with the uniform distribution according to the Random Waypoint Model [23] making no stops. This reflects movement of cattle (up to 1.5 m/s [33]) and human users (walking speed up to 2 m/s [34]).

At the beginning of the simulation nodes join the network – one node every second. At 100^{th} second of the simulation nodes start to perform proactive caching and at 300^{th} second of simulation each of them starts to issue random queries every 10 seconds. The parameters of the simulation are presented in Table I. We compare three approaches: (1) no caching, (2) only passive caching, (3) passive and proactive caching. We measure the percentage of successfully answered queries i.e. success ratio (SR) and total amount of data transferred by all nodes i.e. network overhead (NO). We measure network overhead as it is related to the energy consumption of the nodes.

We alternate following parameters of the simulation to measure their influence on the results: (1) velocity of the nodes, (2) average number of nodes in range, (3) number of nodes with constant density of 100 nodes/km² and constant number of querying nodes - 10, (4) time between issuing queries. In the experiments examining influence of velocities on SR and NO we use constant velocities for the nodes as this shows better the examined dependency. In particular a small change in the overall constant velocity is likely to have more influence than a small change in the maximal random velocity. We used velocities ranging from 0.5 m/s to 20 m/s (72 km/h) to examine if our protocol can cope with network nodes placed on vehicles such as trucks, tractors or off-road cars.

TABLE I
SIMULATION PARAMETERS

Parameter	Value
T_1	15 s
T_2	0.1 s
T_3	0.01 s
t_{max}	1000 s
$t_{min\ gap}$	0.01
Size of a cache (*k*)	10 states
Size of a state	1024 B

B. Simulation Results

Fig. 3a and Fig. 3c show success ratio and network overheads for querying and proactive caching in the conditions of variable velocity. We can see that by using only passive caching we achieve an improvement of the success ratio of up to 40% even though the cache can accommodate only 10 states. Proactive caching gives higher improvement than only passive caching mainly in the conditions of high velocity (6 m/s), such as 44% in comparison to no caching at all. In case of lower velocities a node does not have opportunity to exchange its states with sufficiently many other nodes. Similarly high advantage of proactive caching could be achieved for the lower velocities if the period when nodes can exchange their states lasted longer than 300 seconds, which is highly probable in the real life. Using both passive and proactive caching highly decreases the network overhead for velocities up to 8 m/s and slightly increases the network overhead for higher velocities.

Fig. 3b and Fig. 3d show success ratio and network overhead in the conditions of changing network density. For the densities ranging from average to very low (from 2.1 to 4.8 average nodes in range) passive caching gives almost constant advantage of about 10%. In the very sparse topologies (1.2 to 1.6 nodes in range on average) proactive caching has 1% to 2% of the advantage against only passive caching. The dense topologies (8.6 and more nodes in range on average) are not challenging having success ratio higher than 96% for all the cases. The figures show that using both proactive and passive caching gives an important advantage in decreasing network overheads, particularly for the average densities (average number of nodes between 3 and 20).

Fig. 4a and Fig. 4b show success ratio and network overhead in the conditions of changing number of nodes. We can notice that passive only caching improves scalability. In particular for larger topologies (200 and 300 nodes) passive

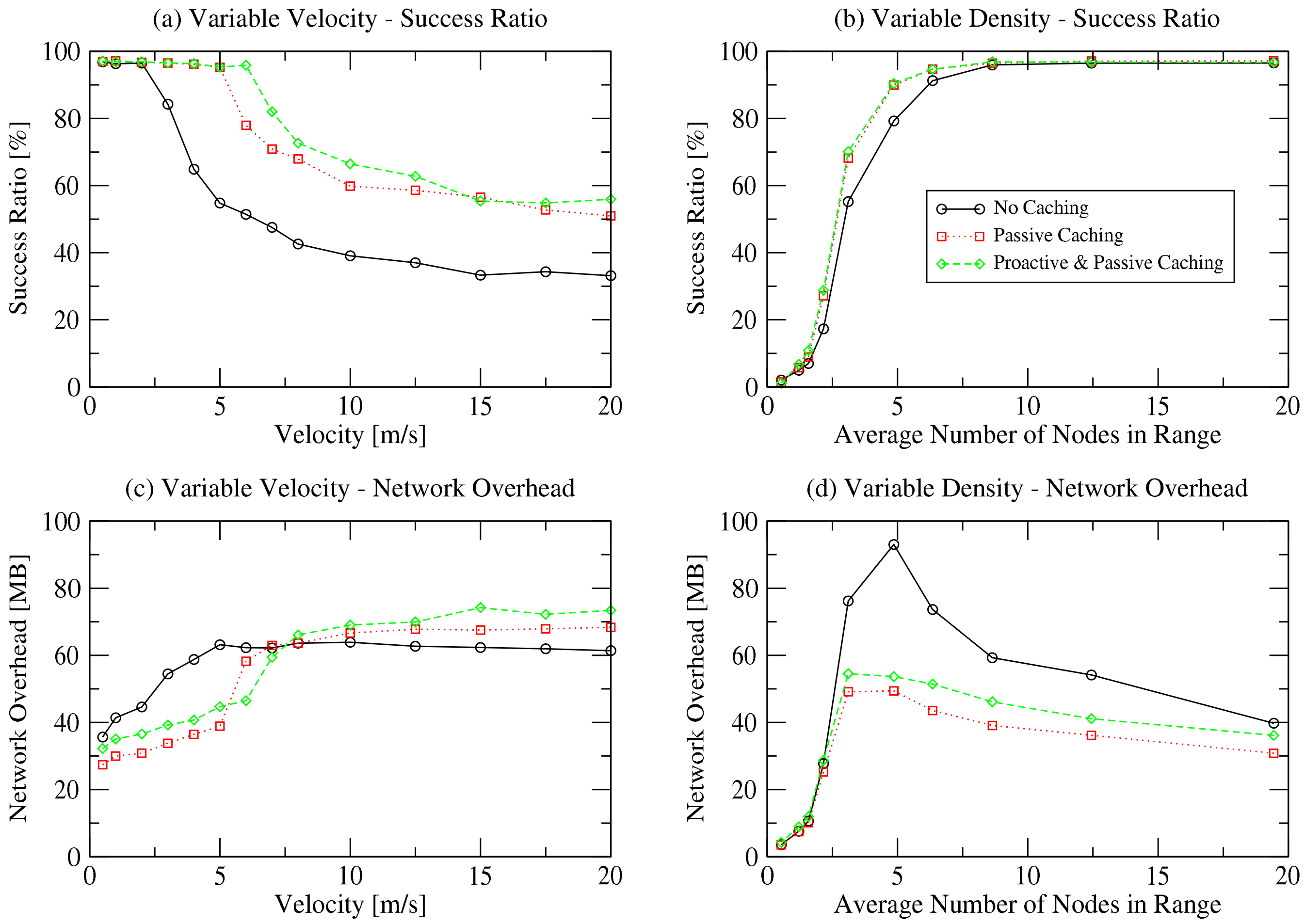

Fig. 3. Success ratio and network overhead in variable velocity and density

caching increases success ratio by 2% to 3%. It also slightly decreases network overhead. Proactive caching gives lower advantage in Success Ratio than passive only caching in larger topologies but higher in smaller topologies (improvement by 1.5% to 4% for topologies with 50 to 200 nodes). Proactive caching slightly increases network overhead in comparison to no caching at all.

Fig. 4b and Fig. 4d present the influence of the increased frequency of queries on success ratio and network overhead. In case of high frequencies of queries, using passive only or passive and proactive caching gives similar improvement in success ratio from 8% to 20% for higher frequencies of querying (pauses between queries shorter than 7 seconds). The increase in network overhead in higher frequencies of queries caused by caching can be attributed to the increased success ratio. More precisely, additional traffic was used for answering queries that would not be answered without the help of caching.

In general, using passive caching always increases success ratio and decreases network overhead in sparse topologies. It also increases scalability. Using proactive caching is appropriate in case of very high mobility or scenarios with anticipated long periods of limited or no query traffic.

C. *Energy Consumption Considerations*

We assume the energy model [35] based on the measurements of WaveLAN radios [36] where idle:receive:send ratios are 1:1.05:1.4. Considering that, additional energy consumption caused by promiscuous mode of wireless network interfaces and packet overhearing required by passive and proactive caching is relatively small.

As passive caching does not require transmitting any additional data and considerably limits network overhead necessary for answering queries, it limits the energy consumption of the nodes. Constant listening performed by the nodes is expensive in terms of energy consumption but it limits the delays of answering queries. Limiting these delays is important as they can negatively influence users' ability to work interactively.

In contrast to passive caching, proactive caching requires transmitting additional data, which increases energy consumption but this is the cost of increasing success ratio in conditions of higher velocities. Energy efficiency of proactive caching can be improved by allowing nodes which are below a certain threshold of remaining battery capacity to restrain from participating in proactive caching. That means neither sending beam packets nor answering to received beam packets.

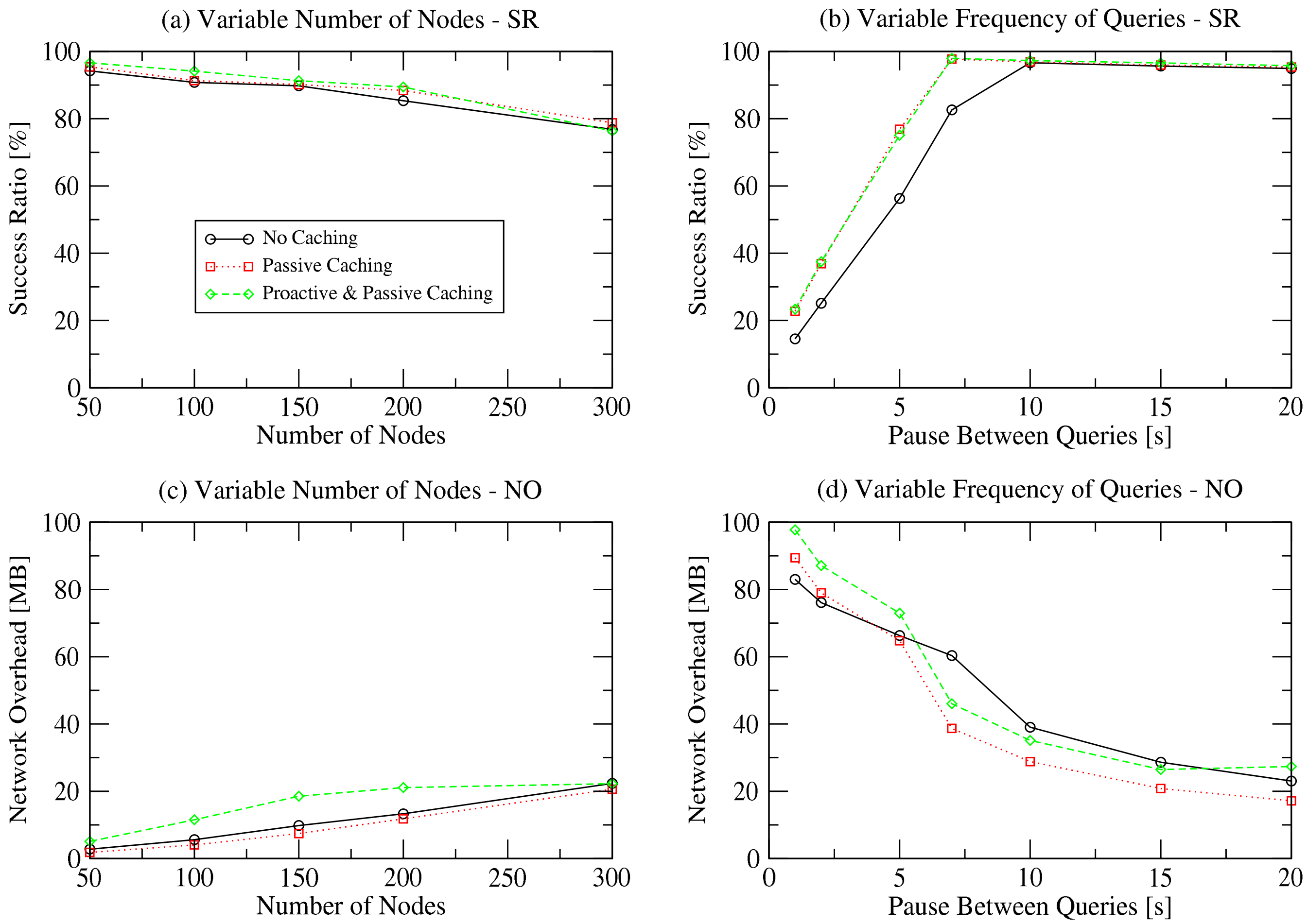

Fig. 4. Success ratio and network overhead in variable number of nodes and frequency of queries

VII. Conclusions and Future Work

In this paper we investigated use of mobile sensor network technologies for a nationwide cattle monitoring system. We proposed an architecture which decreases maintenance and deployment cost due to its self organization and addresses defined security requirements.

For this scenario we proposed a novel DHT substrate for large scale, highly mobile ad hoc sensor networks optimised for disconnections. We showed its improved performance over a wide range of simulations in terms of its success query ratio and network overheads for highly mobile, large scale and disconnected scenarios. The improvement of success ratio increases reliability of the proposed architecture.

Our DHT substrate has much wider applications in addition to the proposed scenario. It can be also used for other large scale monitoring applications with mobile nodes such as monitoring children, elderly people and mentally impaired people [37].

We are currently working on the peer to peer overlay hierarchical protocols addressing Internet connectivity of gateway nodes, remote users and more powerful nodes storing historical data produced by sensor nodes.

References

[1] H. Williams, N. Reeves, D. Jones, and W. Morris, "The economics of lowland beef production in England, 2003 : a report on the results of a special study of lowland beef enterprises in England", Aberystwyth : The University of Wales, Farm Business Survey Unit, Aberystwyth, 2005.

[2] D. P. Davis, "Emerging animal diseases: global markets, global safety : a workshop summary", the National Academies' Board on Agriculture and Natural Resources of the Division on Earth and Life Studies, 2002.

[3] *Cattle keeper's handbook*, British Cattle Movement Service, Workington, Cumbria, UK,

[4] K. Maatje, R. M. d. Mol, and W. Rossing, "Cow status monitoring (health and oestrus) using detection sensors", *Computers and Electronics in Agriculture*, 1997, 16 (3), pp. 245-254(10).

[5] K. Fall, "A Delay-Tolerant Network Architecture for Challenged Internets", In *Proc. of SIGCOMM*, Karlsruhe, Germany, 2003.

[6] S. Jain, M. Demmer, R. Patra, and K. Fall, "Using Redundancy to Cope with Failures in a Delay Tolerant Network", In *Proc. of SIGCOMM*, Philadelphia, Pennsylvania, USA, 2005.

[7] A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler, and J. Anderson, "Wireless Sensor Networks for Habitat Monitoring", In *Proc. of WSNA*, Atlanta, USA, 2002.

[8] L. B. Ruiz, T. R. M. Braga, F. A. Silva, H. P. Assunção, J. M. S. Nogueira, and A. A. F. Loureiro, "On the Design of a Self-Managed Wireless Sensor Network", *IEEE Communications Magazine*, 2005, 43 (8), pp. 95-102.

[9] P. Zhang, C. M. Sadler, S. A. Lyon, and M. Martonosi, "Hardware Design Experiences in ZebraNet", In *Proc. of SenSys*, Baltimore, Maryland, USA, 2004.

[10] K. Mayer, K. Ellis, and K. Taylor, "Cattle Health Monitoring Using Wireless Sensor Networks", In *Proc. of The 2nd IASTED International Conference on Communication and Computer Networks*, Cambridge, Massachusetts, USA, 2004.

[11] "Environmental Studies", URL: http://www.environmental-studies.de/.

[12] A. Frost, C. P. Schofield, S. A. Beaulah, T. T. Mottram, J. A. Lines, and C. M. Wathes, "A review of livestock monitoring and the need for integrated systems", *Computers and Electronics in Agriculture*, 1997, 17, pp. 139-159.

[13] "Wildlife Computers", URL: http://www.wildlifecomputers.com/.

[14] R. E. Kenward, *A Manual for Wildlife Radio Tagging*, Academic Press, 2001.

[15] "Habit Research", URL: http://www.habitresearch.com/.

[16] "Lotek Fish and Wildlife Monitoring Systems - Radio, Acustic, Satellite & Archival Telemetry Solutions", URL: http://www.lotek.com/.

[17] Televit, "GPS collars", URL: www.televit.se.

[18] "Advanced Telemetry Systems", URL: www.atstrack.com.

[19] Z. Butler, P. Corke, R. Peterson, and D. Rus, "Dynamic Virtual fences for Controlling Cows", In *Proc. of Ninth International Symposium on Experimental Robotics*, Marina Mandarin Hotel, Singapore, 2004.

[20] W. J. Eradus and M. B. Jansen, "Animal identification and monitoring", *Computers and Electronics in Agriculture*, 1999, 24, pp. 91–98.

[21] H. Pucha, S. M. Das, and Y. C. Hu, "Ekta: An Efficient DHT Substrate for Distributed Applications in Mobile Ad Hoc Networks", In *Proc. of WMCSA*, English Lake District, UK, 2004.

[22] T. Zahn and J. Schiller, "MADPastry: A DHT Substrate for Practicably Sized MANETs", In *Proc. of Applications and Services in Wireless Networks*, Paris, Fance, 2005.

[23] D. B. Johnson and D. A. Maltz, "Dynamic Source Routing in Ad Hoc Wireless Networks": Kluwer Academic, 1996.

[24] C. E. Perkins and E. M. Royer, "Ad-hoc On-Demand Distance Vector Routing", In *Proc. of WMCSA*, 1999.

[25] A. Rowstron and P. Druschel, "Pastry: Scalable, Decentralized Object Location, and Routing for Large-Scale Peer-to-Peer Systems", In *Proc. of Middleware*, Heidelberg, Germany, 2001, pp. 329-350.

[26] M. Haenggi and D. Puccinelli, "Routing in Ad Hoc Networks: A Case for Long Hops", *IEEE Communications Magazine IEEE*, 2005, 43 (10), pp. 93-101.

[27] A. Rowstron and P. Druschel, "Storage management and caching in PAST, a large-scale, persistent peer-to-peer storage utility", In *Proc. of ACM SIGOPS*, 2001.

[28] D. B. Terry and V. Ramasubramanian, "Caching XML Web Services for Mobility", *ACM Queue*, 2003, 1 (1), pp. 71-78.

[29] A. Kureshy, "Architecting Disconnected Mobile Applications Using a Service Oriented Architecture", *MSDN Library*, 2004, September.

[30] A. Hampshire and C. Greenhalgh, "Caching OGSI Grid Service Data to Allow Disconnected State Retrieval", In *Proc. of GADA Workshop*, Agia Napa, Cyprus, 2005, pp. 284 – 293.

[31] A. Hampshire and C. Greenhalgh, "Supporting OGSI and WSRF Specific Functionality in Intermittently Available Network Environments", In *Proc. of UK eScience All Hands*, Nottingham, UK, 2005.

[32] L. Breslau, D. Estrin, K. Fall, S. Floyd, J. Heidemann, A. Helmy, P. Huang, S. McCanne, K. Varadhan, Y. Xu, and H. Yu, "Advances in Network Simulation", *IEEE Computer*, 2000, 33 (5), pp. 59-67.

[33] C. J. C. Phillips, *Cattle behaviour*, Farming Press, Ipswich, UK, 1993.

[34] S. D. Pinna, *Forces and Motion*, Raintree Steck-Vaughn Publishers, Austin, Texas, USA, 1998.

[35] Y. Xu, J. Heidemann, and D. Estrin, "Adaptive Energy-Conserving Routing for Multihop Ad Hoc Networks", USC/ISI, 527, 12 October 2000.

[36] M. Stemm and R. H. Katz, "Measuring and Reducing Energy Consumption of Network Interfaces in Hand-Held Devices", *IEICE Transactions on Communications*, August 1997, E80-B (8), pp. 1125-1131.

[37] G. Záruba, F. Kamangar, M. Huber, and D. Levine, "CONNECT: A Personal Remote Messaging and Monitoring System to Aid People with Disabilities", *IEEE Communications Magazine*, 2005, 43 (9), pp. 101-109.