



# Kent Academic Repository

Tateson, J. and Marshall, Ian W. (2004) *A Novel Mechanism for Routing in Highly Mobile Ad-hoc Sensor Networks*. In: Karl, H. and Willig, W. and Wolisz, A., eds. *Wireless Sensor Networks*. LNCS, 2920/2 (2920). Springer, pp. 204-218. ISBN 978-3-540-20825-9.

## Downloaded from

<https://kar.kent.ac.uk/14033/> The University of Kent's Academic Repository KAR

## The version of record is available from

## This document version

UNSPECIFIED

## DOI for this version

## Licence for this version

UNSPECIFIED

## Additional information

## Versions of research works

### Versions of Record

If this version is the version of record, it is the same as the published version available on the publisher's web site. Cite as the published version.

### Author Accepted Manuscripts

If this document is identified as the Author Accepted Manuscript it is the version after peer review but before type setting, copy editing or publisher branding. Cite as Surname, Initial. (Year) 'Title of article'. To be published in *Title of Journal*, Volume and issue numbers [peer-reviewed accepted version]. Available at: DOI or URL (Accessed: date).

## Enquiries

If you have questions about this document contact [ResearchSupport@kent.ac.uk](mailto:ResearchSupport@kent.ac.uk). Please include the URL of the record in KAR. If you believe that your, or a third party's rights have been compromised through this document please see our [Take Down policy](https://www.kent.ac.uk/guides/kar-the-kent-academic-repository#policies) (available from <https://www.kent.ac.uk/guides/kar-the-kent-academic-repository#policies>).

# A Novel Mechanism for Routing in Highly Mobile Ad Hoc Sensor Networks

Jane Tateson and Ian W. Marshall

BTexact, Adastral Park, Martlesham Heath, Ipswich IP5 3RE, UK  
{jane.tateson,ian.w.marshall}@bt.com

**Abstract.** This paper describes a novel routing mechanism for a network of highly mobile sensor nodes that routes data over dynamically changing topologies, using only information from nearest neighbours. The preferred forwarding directions of mobile sensor nodes are modelled as vectors, and a scalar trigger is used to determine data forwarding. Simulations have demonstrated that this technique operates successfully in sparse networks, where node movements are unpredictable, and data generation by nodes is non-uniform. The application scenario is a self-configuring network of mobile nodes, floating in the sea, that is tracking the movements of a shoal of fish. The requirements of the technique in terms of memory are minimal, with very few parameters and very little code being needed, as is appropriate for the low-powered microprocessors envisaged.

## 1 Introduction

Scientists often want to monitor processes in the environment, where these processes could be changes in glacier size, seismic activity, or the dynamics of endangered species. Water companies have a particular interest in monitoring the movements of pollutants in rivers and reservoirs. Dredging companies are interested in the effects their activities are having on the coast-line. All these groups have typically had to rely on data gathered from a handful of strategically placed, highly expensive packages, each containing many sensors. Because the environments of interest are often hostile or difficult to access, the sensor packages have had to be housed in heavy-duty casings, to prevent damage or loss. In the case of oceanography, even collecting these sensor packages for data retrieval, by hiring a ship, costs thousands of pounds. Beyond the disadvantage of high cost, the data is only retrievable at long intervals. Even in networked sensor systems[1], localized measurements only represent a tiny fraction of the total area where environmental change is taking place.

What scientists want is near real-time access to data which has been sampled at many points in the environment of interest. In order to achieve this, a much more flexible infrastructure is needed, one that consists of a large number of nodes, each containing one or more sensors, where sensor nodes are able to self-configure dynamically, manage their own resources in the context of the monitoring experiment, and collect and forward data efficiently. This would typically involve multi-hop wireless communication amongst low-powered sensor nodes, in order to forward measurement data to a number of network sinks. The sinks are likely to be higher-

powered devices, with much larger memory than the sensor nodes, and able to interface, perhaps via satellite, with fixed network devices on land. The important thing is that the sensor nodes are cheap enough that they can be deployed in large numbers, and that they operate collectively as an efficient data-gathering network, in spite of node failures and varying topology.

There is relevant work in the literature that refers to dense networks of sensor nodes, and how to manage the wake-up of a subset of nodes from a sleeping state, when an event of interest occurs within the monitoring area of the sensor network. Here the sensor nodes are static, and their major problem is how to conserve power between the occurrence of interesting events, and then how to use the node battery resources fairly. For example, work by Cerpa et al.[2] refers to habitat monitoring as a driver for wireless communications technology, and focuses on power-saving by nodes outside regions where interesting changes could be observed, switching themselves off, and being triggered to switch back on only when interesting activity is detected in their vicinity. Work by Xu et al.[3] again focuses on using powered-down modes for devices to conserve power, based on whether data traffic is predicted or not, and on the number of equivalent nodes nearby that could be used for alternate routing paths. The assumption here is that the underlying routing will be based on conventional *ad hoc* routing protocols such as AODV[4]. Sensor networks, however, typically would require a lighter weight approach to routing, where decisions are based on succinct information from immediate neighbours only.

Work by Heinzelman et al.[5] has as its focus the use of clustering techniques to reduce bandwidth usage by, for example, data aggregation of similar data, and using predictable transmission times, co-ordinated by the cluster heads. This approach saves significant energy, compared with an always-on approach, but the routing side is simplistic and not fully developed. They assume that devices could all broadcast to the base station if they chose to, which would not be realistic for sensor network applications, as a general rule. Other ways to reduce energy usage include the work by Singh et al.[6], who have made a detailed study of power-conservation in ad hoc networks at the MAC and network layers. They include schemes for devices to power-down in between expected transmissions, and they take into account device load as an important factor in power consumption. Their main concern is to prevent network partitioning when gaps appear in the network as a result of devices running out of battery power.

A lot of work has been done at the University of California and the Intel Berkeley Research Lab, to develop operating systems and networks for small *ad hoc* sensor devices, known as the *Smartdust*[7] project, for which *TinyOS*[8] has been developed. Their nodes self-configure into a hierarchical structure to find shortest paths to the sinks, but this process is not power-aware.

Whilst the routing schemes referred to above are appropriate for networks of static nodes, they do not handle node mobility explicitly, and would certainly have great difficulty with highly mobile nodes. *Ad hoc* routing protocols, on the other hand, are designed to cope with node mobility. Many *ad hoc* routing protocols have been devised. Some of the most widely known are DSDV[9], TORA[10], DSR[11] and AODV[4]. DSDV[9] maintains a routing table listing the next hop for each reachable destination. Routes are tagged with sequence numbers, with the most recently determined route, with the highest sequence number, being the most favoured. There are periodic updates of routes and sequence numbers. TORA[10] discovers routes on

demand and gives multiple routes to a destination. Route query and update packets are sent for each destination. Although routes are established fairly quickly, there are often routing loops, leading to dropped packets. DSR[11] uses source routing, rather than hop-by-hop routing, so each packet has a complete route, listed in its header. The protocol uses route discovery and route maintenance, with nodes maintaining caches of source routes that have been learned or overheard. AODV[4] combines route discovery and route maintenance with hop-by-hop routing. Route request packets create reverse routes for themselves back to their source nodes. Hello messages are periodically transmitted by nodes so that neighbours are aware of the state of local links. A comparison of the performance of these protocols[12] has shown widely differing results in the size of routing overhead. The total overhead is worst for TORA, and becomes unacceptable for a network size of 30 source nodes. However, the main problems with using these ad hoc network protocols for a network of mobile sensor devices is that 1) the size of processor and memory required is too large, and 2) the protocols are not energy usage aware.

Sensor networks are envisaged as consisting of very small, very cheap microprocessors, e.g. 16 bit, with 32 kbytes of RAM. They will also have a finite battery supply, which will be difficult, and probably not desirable to replace. It is therefore very important that any communication protocol is energy-efficiency aware, and also pared to a minimum in communication overhead and memory usage.

## 2 Routing Mechanism

The application scenario is a sparsely-populated network of highly mobile, wireless sensor nodes that take measurements from the environment and send data back to one of the network sink devices. Although there has been work published on communication between sensor nodes[6-9], what is unique to this work is its ability to route data well even when nodes are moving rapidly. The method is independent of network scale.

The technique being presented here assumes that nodes know their relative positions. There are many ways that this could be achieved. Relative position can be determined if nodes have a directional antenna, a ranging mechanism, and a digital compass for a reference direction. Nodes can determine even their absolute positions themselves [13,14], using some fixed nodes or GPS-enabled nodes as reference points. And a knowledge of position is generally essential for environmental measurements to be meaningful.

Each mobile data-gathering sensor node needs a forwarding direction to send its data back to a network sink. Because the nodes are moving rapidly, even their nearest neighbours may change between data transmissions. Routing decisions must be made 'on-the-fly', using very recently gathered information. In this work, an analogy is made between the forwarding direction of a node and the co-ordinates of a *polar* bond, associated with that node, that is able to rotate to find its optimal orientation. A *polar* bond has one end that is negatively charged, and one end that is positively charged. Polar bonds will tend to align themselves with their neighbours' polarities. If a chain of real polar molecules formed, you would find alternating positive and negative polarities lined up all along the chain, just as you would find that a set of

magnets, placed together in a line, would seek to have opposite poles touching : + - + - + - .

When a node is ready to transmit data, it can determine its forwarding direction by calculating the 'optimal' orientation of its associated polar bond. This is achieved by combining the alignment influences of the forwarding directions of neighbouring nodes.

Ideally, a node wants to do more than identify a transmission hop in the right direction, it wants to forward according to the best chance of its data getting all the way back to a sink. A complete route back to a sink is given by a chain of nodes, where each node is within forwarding range of the previous node in the chain. Such a structure may only form quite briefly, but, when it forms, it will result in stronger interactions between the 'polar bonds' along the chain. Calculating the strength of an interaction is a natural extension of the calculation of forwarding direction, and gives us a scalar trigger to determine when data forwarding should take place.

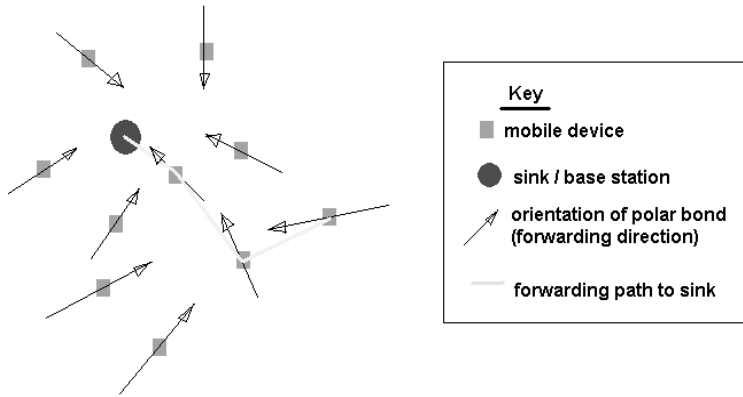
Each polar bond is modelled as two atoms, one atom is positively charged, the other atom is negatively charged. (The charges are equal, but opposite in sign). The network sinks are modelled as unit positive point charges. This results in nodes in the vicinity of a sink pointing their forwarding directions towards this sink. These forwarding directions (polar bond orientations), in turn, have knock-on effects on other nearby nodes that are too far away from the sink to forward to the sink directly. And these nearby nodes influence nodes yet further away from the sink, to aim their forwarding directions so as to make a path to the sink, via the intermediate nodes, and so on, throughout the network. The result is that, irrespective of network topology, all nodes (unless cut-off entirely from regions of the network with paths to a sink) will have a forwarding direction that is likely to result in the multi-hop transmission of data to a sink. Forwarding directions are updated dynamically, so that as soon as a link is re-established, transmission of data can re-start. If the network is very sparsely populated, most data transmissions may only occur when a node comes within direct range of a sink. In densely populated networks, much longer paths, in terms of number of hops, will be common. The important thing is that this method is flexible enough to cope with a wide range of circumstances, in terms of network topology and node speed, without such variations requiring special treatment.

The electrostatic analogy, where *electrostatic* simply refers to the interactions of charges, is a framework to enable the sum of alignment influences of neighbouring nodes to be determined quantitatively. We need to know how far away a node is, and what is its current preferred forwarding direction, but we also need to know how to combine the effects of several nodes, some of which will have conflicting (opposite) influences. Fortunately, there are well-established ways to combine such effects together, one example of which is found in the study of electrostatics. The routing mechanism makes use of these well-known relationships, so that the influences of neighbouring nodes are taken account of correctly. An illustration of how such forwarding directions / polar bond orientations are used to route data to a network sink or base station, is given in Figure 1.

This approach to routing could equally be used to route packets in multi-hop cellular networks to a nearby base station. No sensor-network-specific assumptions are made, e.g. that all nodes generate data. Extension to general *ad hoc* networks, where any node can send to any other, would involve the nodes maintaining a number of forwarding directions, for different target nodes. In order to overcome scaling difficulties, in this case, the nodes could be organised into hierarchical clusters, with

nodes needing to have forwarding directions to cluster-heads and to nodes within their own clusters.

Illustration of Polarization Used For Mobile Device Routing



**Fig. 1.** Illustration of the use of polar bond orientations to give preferred forwarding directions for mobile sensor nodes, forwarding data to a network sink

Note that the type of routing reported here is, for the sake of clarity, in all cases, point-to-point. This simply means that nodes have unique identifiers, quote their identifiers when communicating with their neighbours, and ignore broadcasts that are not meant for themselves, by reading the transmitted data header. For greater assurance of packet delivery, the redundancy of multi-path routing may be desirable, but this issue is orthogonal to the main focus of this work. Also, it may be of interest to compare the hop-by-hop routing presented here to more general Distance Vector methods, for completeness. However, here - despite multiple network sinks - only a single, node-independent forwarding-direction is maintained as routing state.

Each node maintains a tuple representing its location, forwarding direction, quantity of data in its buffer, and a model parameter called the 'induced charge'. When node A has data to send, it broadcasts to see what other nodes are in the vicinity, and each neighbour replies with its tuple. Node A uses the advertised tuples to adjust its own forwarding direction, and determine its forwarding decision. This is most clearly understood as simply a novel form of route table construction that requires less state to be maintained (no addresses), and converges to "good enough" end to end routing solutions rapidly enough to allow very rapid movement to be tracked because no handshake messages are required.

Note that there are 4 different bond types used. The choice of bond is determined by the quantity of data in a device's buffer. Each polar bond has 3 properties: 1) physical reach of influence (bond length), 2) ability to influence other devices (permanent charge) and 3) ability to be influenced by other devices (*polarizability*).

We want the strongest interactions to occur between a device with a full buffer and a device with an empty buffer, for data transfer. We model empty buffers as short bonds with a strong polarising (influencing) effect, and full buffers as long bonds that are easily polarized (influenced). This model requires the devices to have permanent memory of  $(4 \times 3) = 12$  values plus the forwarding threshold parameter, giving 13 values in total, which can be stored as 13 bytes.

Given the tuples from neighbouring nodes, Node A, having data to send, needs to determine its own 'optimal' forwarding direction, i.e. lowest energy polar bond orientation. Rather than finding the absolute minimum electrostatic energy orientation for the polar bond modelled on node A, this polar bond (forwarding direction) is rotated in 45 degree steps, with node A calculating the electrostatic energy 8 times ( $8 \times 45 = 360$  degrees), and choosing the orientation which has the lowest electrostatic energy. (Because the 'bonds' are symmetric about the device - equal but opposite charge on the 'atoms' - only 4 calculations are needed, with the other 4 results being simply opposite in sign.) Calculating this electrostatic energy has the form

$$U_A = \sum_i^{i \in A} \sum_j^{j \notin A} \frac{q_i q_j}{r_{ij}}, \quad (1)$$

where  $q_i$  is the permanent charge on atom  $i$ , which is on device A, and  $r_{ij}$  is the distance between atoms  $i$  and  $j$ , where  $j$  is an atom on a nearby device. Having found the lowest electrostatic energy orientation of the polar bond on device A, this orientation is adopted as the forwarding direction for node A.

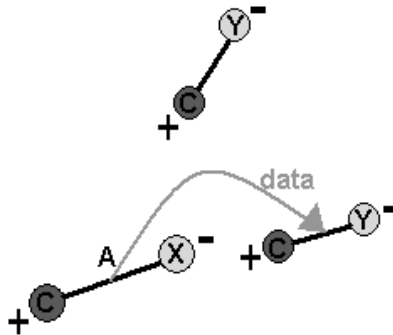
As well as finding the forwarding direction, we also want a way of expressing how strong the interaction (alignment influence) is with neighbouring nodes, for a strong interaction indicates that forwarding is likely to be useful: locally favourable, in terms of node proximity and buffer capacity, and also a good chance of a long-range forwarding chain to the sink. By calculating the electric field at the negative end of the polar bond of node A, we can derive the *induced charge* on the polar bond. (Under the influence of neighbouring charges, the charge distribution of a molecule changes.) If this *induced charge* exceeds a pre-set threshold, then node A decides to forward data to the nearest device to the negative end of its polar bond, provided that this device is within transmission range. (Note that the terms referred to here: *electric field* and *charge* only have meaning in the context of the routing model, and do not refer to the battery levels or other properties of the devices themselves.)

So, the second stage of the calculation is to calculate the electric field at the negatively charged atom  $X$ , on node A, which is given by

$$E_X = \sum_j^{j \in A} \frac{q_X (q_j + \delta_j)}{r_{Xj}^2}, \quad (2)$$

where  $\delta_j$  is the induced charge on atom  $j$ , on a neighbouring node, and  $q_x$  is the charge on the negative atom X. From the electric field, it is simple to calculate the induced dipole moment  $\mu_x$ , where this is given by  $\mu_x = \alpha_x E_x$ , where  $\alpha_x$  is the atomic *polarizability* of atom X. The induced charge on the negatively charged atom of the polar bond on node A,  $\delta_x$ , is taken as equal to  $\mu_{C-X} / d_{C-X}$ , where  $d_{C-X}$  is the length of the polar bond. Note that  $\mu_x$  and the bond dipole moment  $\mu_{C-X}$  are treated as equivalent.

So, when the induced charge,  $\delta_x$ , exceeds a fixed threshold, data is forwarded from device A to the nearest device to the negatively charged end of its polar bond, as long as this device is within broadcast range of A. If all the data has been forwarded from device A, the bond type on device A is re-set to the one appropriate for an empty buffer (short, strongly polarising), and any forwarding means that the induced charge is re-set to zero.



**Fig. 2.** X is the negatively charged end of the polar bond associated with node A, and X has been polarized beyond a fixed threshold by the 'atoms' (poles) associated with nearby nodes. Data is transferred from node A to the node which is nearest to X

Another feature of this approach is the way that nodes respond to changing battery levels. A node's broadcast range is constrained to be no greater than *range*, according to the ratio of (battery power left)/(time left), as follows:

$$\text{range} = \text{range}_{\min} + \frac{B_0 - B}{2(t_{\max} - t)} (\text{range}_{\max} - \text{range}_{\min}), \tag{3}$$

$$\text{range} \leq \text{range}_{\max}$$



where  $B$  is the current battery level and  $B_0$  is the initial battery level, and  $t$  and  $t_{max}$  are the current time and length of the data-gathering experiment respectively. More sophisticated controls could be used, based on the quantity of data left in the device buffer, perhaps, but this simple approach has been shown to give advantage compared with a range that is independent of battery power, and is easy for the primitive devices to calculate.

Lastly, the choice of influencing neighbouring nodes is restricted to exclude the node that has just sent the forwarding device some data. This is to stop data "ping-ponging" backwards and forwards.

### 3 Simulation Results

The main scenario used to test and evaluate the performance of the invention is 20 mobile sensor network devices floating on the surface of the sea, and 3 network sink devices which are fixed, though this need not be the case. The simulation includes a model of water currents including a moving centre of rotation (a whirlpool or gyre) and a model of a shoal of fish with flocking behaviour. Whenever a fish comes within a certain close range of a sensor device, this generates a packet of data at the sensor device. The purpose of this scenario was to model sensor devices moving fast and unpredictably, and subject to unpredictable and unequal load. The network is sparsely populated with the *average* number of nodes within transmission range of any other node during a simulation, being less than 0.2.

There is a lot of interest in the health of fish stocks. Currently, estimates of fish numbers and locations rely on the crude approach of fishermen fishing out tagged fish from their catches and estimating where the tagged fish were found. The scenario simulated here would offer a much more comprehensive coverage of the surface area of the sea. It has been assumed that tagged fish would send out a sonar signal that would enable the floating sensor nodes to detect them.

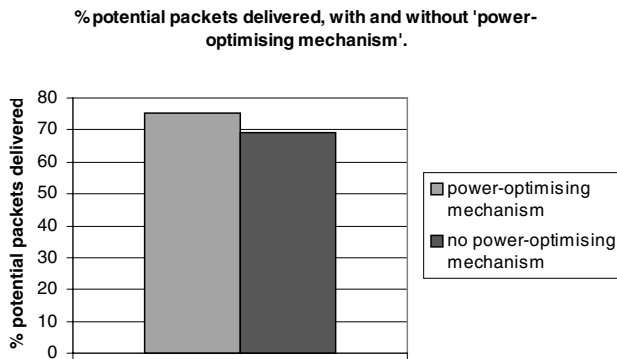
Simulations were carried out with 20 mobile sensor devices and 3 fixed sinks, with a shoal of 30 fish. The transmission cost model used is very simple, being proportional to  $r^2$ , where  $r$  is the inter-node distance. In this work, the receive cost has been neglected; the inter-node distances are assumed to be large, on the scale of 100s of metres. The results presented are each averages of 20 simulation runs. There was a wide range of possible data to collect, between simulations, depending on the movements and interactions of sensor devices with fish. The average number of potential data packets to be collected was 4300, with a standard deviation of 3400. The mobile sensor devices were given equal, finite battery resources, some of which were entirely drained by the data gathering experiment. Results quote the percentage of potential packets of data that are recovered at the sinks by the end of the data-gathering experiment. The number of potential packets is the number of times mobile devices or sinks encountered a fish within a specified range, which would have generated a data packet. This total includes interactions between fish and devices which have exhausted their batteries and are unable to record or forward this data.

No attention has been given to the nodes conserving power by being 'asleep' during periods of inactivity, though this would be an important component of a live system.

The communication costs for exchanging the small amounts of state information needed to choose a neighbour for forwarding, have not been taken into account (except for the experiments whose results are given in Figure 8). But the requirements for the exchange of very short control messages between sensor nodes would be the same for the two methods being compared here. In summary, the technique presented here only covers a subset of the issues involved in designing an operational mobile sensor network, focusing on efficient route-finding alone; but the approach would not be in conflict with other aspects of efficient mobile sensor network design.

First, is presented in Figure 3 a graph that compares use of the power-optimising mechanism given in Equation 3, with simulations that maintain a fixed maximum broadcast range. Note that in both cases, the polarization routing mechanism was used, and the transmission range for forwarding data is variable, with power use per packet sent proportional to  $r^2$ . However, the difference is that, with the power-optimising mechanism, the *maximum* broadcast range that a node can use, even for control packets, is reduced. So, devices with lower battery power remaining are less visible, and so are less likely to be sent data, and are more limited in how far they can transmit data. Use of the power-optimising mechanism results in a greater proportion of packets being collected at the sinks, as can be seen in Figure 3.

To evaluate the polarization approach, a performance comparison has been made with a forwarding criterion that is referred to as 'distance only'. This criterion involves device A forwarding data to a neighbouring device B, if device B is the nearest device within range of A, and device B is nearer to A's nearest sink than A. This is simpler than using the concept of polarization energy, but would require a similar level of notification broadcasts of device positions. The following graphs refer to 4 sets of 20 simulations, where each set has a different forwarding cost constant *bat cost*, which is used to *multiply* the inter-device distance, so can be thought of as decreasing device density, or increasing the cost of transmissions by *bat cost*<sup>2</sup>; no additional battery resource was given. Note that the power-optimising mechanism was used for both 'polarization' and 'distance only'.

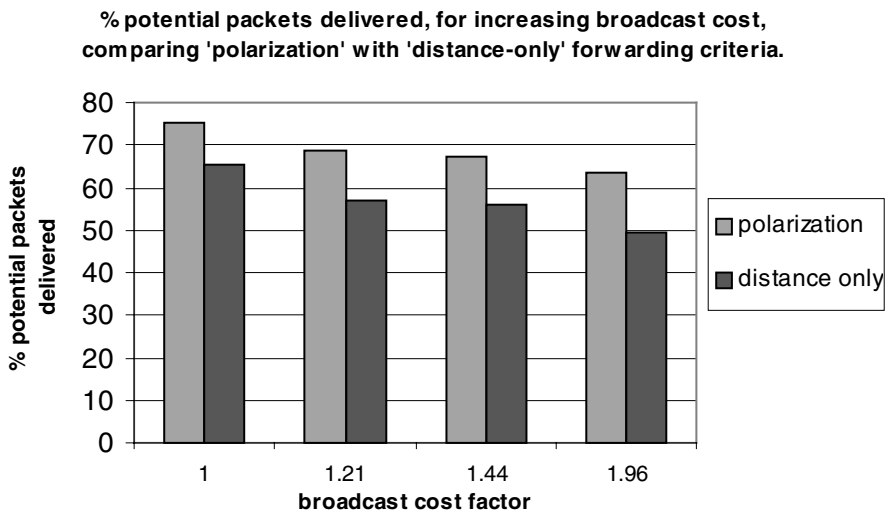


**Fig. 3.** % potential packets delivered on average for 20 simulations using the polarization approach for routing, with and without the 'power-optimising mechanism', which reduces node broadcast range, as relative node battery levels fall

In Figure 4, it can be seen that a significantly higher proportion of packets is successfully collected at the sinks using polarization as the criterion for forwarding data, than distance to sink alone. Similarly, in Figure 5, we see that the average cost of delivering a packet using 'distance only' is as much as 50% greater than when using the polarization forwarding criterion. This is because using the concept of an electrostatic field and polarization triggers make it more likely that data will be forwarded along routes that lead to a sink.

In Figure 6 we see how many fewer hops are needed to deliver packets when the concept of polarization is used. The use of fewer hops is an advantage as it means less processing power being used by devices (this has not been modelled in this work.) These results suggest that if *receive* energy costs were taken into account, as well as transmission costs, that the difference in performance between "polarization" and "distance only" methods would be even greater.

In Figure 7 we see a comparison of the proportion of data forwarding transmissions that occurs when there is a complete routing path to the sink. That is, when a set of devices is arranged in such a way that successive in-range transmissions could result in packets going directly to a sink. This graph shows how much more frequently this occurs when the notion of an electrostatic field is used. This shows that the success of the polarization approach is achieved by conveying long-range structural information using only local transmission interactions, so that transient data forwarding structures can be exploited.

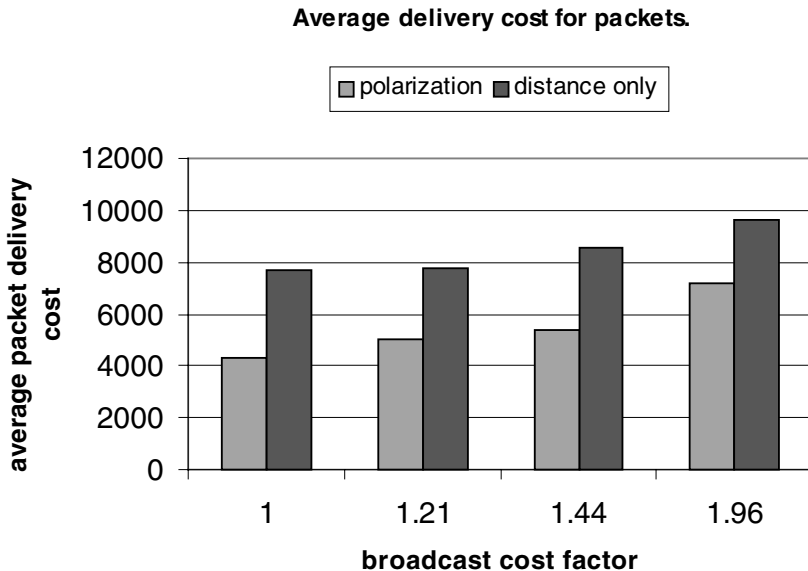


**Fig. 4.** A comparison of % potential packets delivered to sinks in the mobile ad hoc sensor network simulation, for two sets of forwarding rules: 'polarization' and 'distance only'. Each result quoted is the average of 20 simulations; results using 4 different broadcast cost constants are shown

Further work has shown that setting minimum *effective* interaction distances for the electrostatic calculations, improved results for the polarization method further, raising

the average proportion of potential packets delivered (where the broadcast cost factor is 1.0) from 75.4%, to 77.8%. This has the effect of moderating the influence of extremely close neighbouring nodes, which would otherwise unduly dominate the alignment effects. Using this constraint also has the effect of reducing transmissions over very short distances, which would have favourable receive energy cost implications.

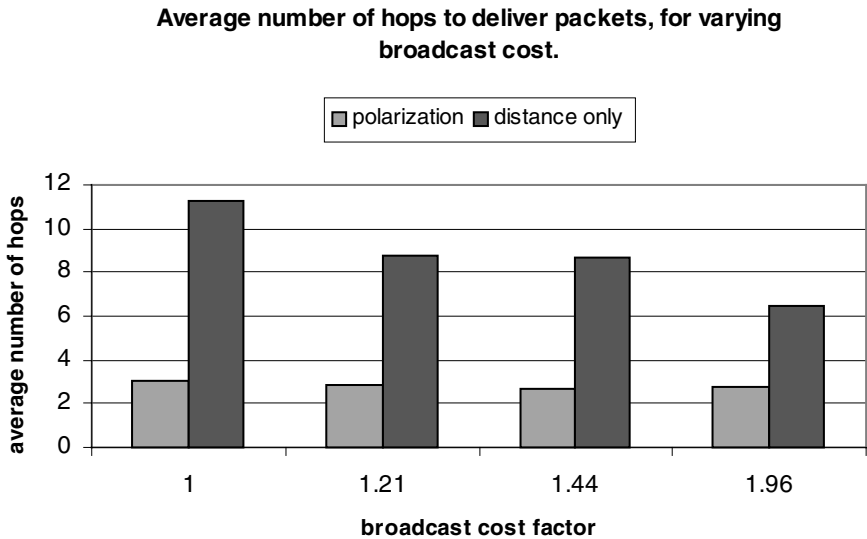
In addition, whereas all results quoted in figures 3-7 were generated with the sensor nodes having effectively infinite buffers, by imposing finite buffers of size 400 packets, the average potential packet delivery rate improved further from 77.8% to 82.3%. This latter improvement is the result of better load-balancing, as can be shown by the fact that the average node deaths (drained node batteries) for simulations fell from 1.25 (out of 20) to 0.5. However, this latter improvement is not specific to the polarization approach, and would be seen for all comparable routing methods.



**Fig. 5.** The relative average delivery costs for packets in mobile ad hoc sensor network simulations, comparing the 'polarization' forwarding criterion with the 'distance only' rule

A final set of simulations was carried out to plot the performance of 'polarization' routing against average node speed. There were 48 mobile sensor nodes, moving randomly, and 2 stationary sinks. Again the transmission cost was as  $r^2$ , and receive energy cost was neglected, but control packet cost was included, being 20% of sending a data packet. The average node movement was one quarter of the maximum possible movement. Movement occurred for every node 'decision-making cycle'. 3 decision-making cycles were needed from when a node checked if it had data to forward, to the point where it had selected a receive node and was now broadcasting

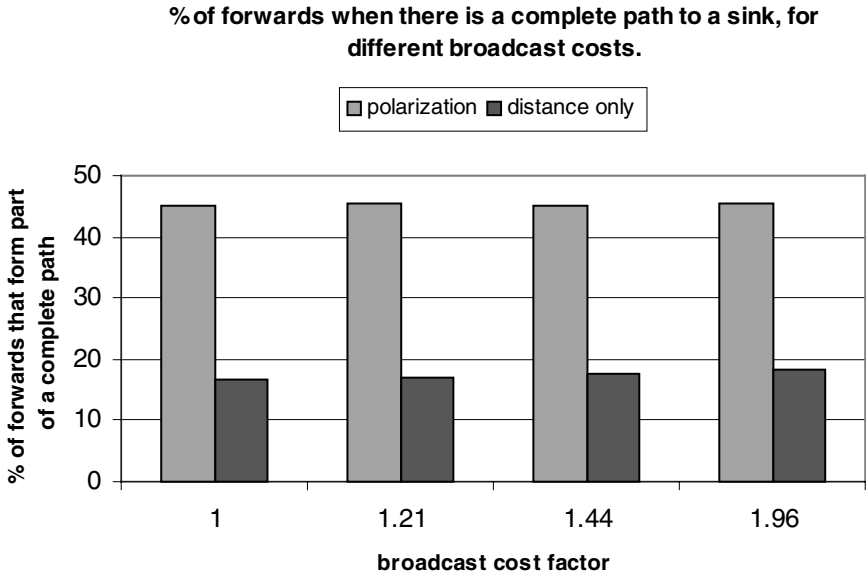
its data. This was to model information from neighbours becoming out of date within the timescale of routing decisions. A comparison was made with a hierarchical routing protocol, similar to the one used for *Smartdust*[7], in which the sinks are at level 0, their nearest neighbours are at level 1, and so on throughout the sensor network, so that each node's level reflects the number of shortest-path hops to its nearest sink. This protocol was implemented in the simulation with appropriate safeguards to prevent erroneous level setting. The results are given in Figure 8. There is an idiosyncrasy to these simulations: the frequency of sensor measurement increases for nodes moving over two regions of the network environment, which means that more data load is put on the network for higher average node speeds. This results in the sensor network performance deteriorating faster with node speed than it would otherwise. However, it can be seen that the 'polarization' approach outperforms the 'hierarchical' approach consistently, especially at higher average node speeds. This is partly because the 'polarization' approach means that the nodes maintain a memory of a reasonable forwarding direction, whereas the 'hierarchical' levels-based approach has nodes relying on routing levels that are changing faster than the level information can be updated.



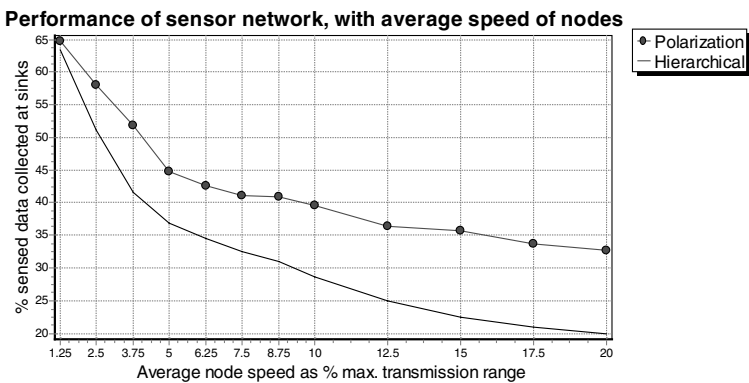
**Fig. 6.** The average number of hops used for packets being delivered to sinks in mobile ad hoc sensor network simulations for the two forwarding criteria: 'polarization' and 'distance only'

Also, the 'polarization' approach drops fewer packets during transmissions, as the forwarding trigger is influenced by receiving node proximity. Rather than just following a simple rule such as 'forward to nearest node at lower level', the polarization approach has a measure of how near is 'near enough'. However, the polarization approach also has an implicit load-balancing component in that receive node buffer capacity is taken into account. (A node with a buffer that is nearly full is less attractive as a recipient of data than a node with an empty buffer.)

Forwarding is also affected by longer-range network structure. This combination of factors enables nodes using polarization routing to make more intelligent forwarding decisions, when node speed is significant.



**Fig. 7.** A comparison of the average % of data forwarding transmissions that occur when there is a viable route all the way to a network sink, comparing the two forwarding criteria: 'polarization' and 'distance only'



**Fig. 8.** Performance of sensor network, with average node speed given as percentage of maximum transmission range, per node 'decision-making cycle', for 'polarization' routing and 'hierarchical' routing

## 4 Conclusion

In conclusion, a novel protocol inspired by modelling the forwarding directions of mobile devices as polarizable bonds, has been used to construct dynamically-updated forwarding paths throughout a mobile device network. It has been shown that this model routes data so that less transmission energy is used for forwarding, more data is recovered, and fewer hops are needed than with a simpler approach that always chooses to forward in the direction of the nearest network sink. With increasing node speed, it has been shown that our proposed approach is more successful at routing than routing that relies on a hierarchical-levels approach. The success of the method rests on its ability to convey longer-range network information than would otherwise be available to the mobile devices, through the interactions of their preferred forwarding directions.

## References

1. <http://www.mysound.uconn.edu/index.html>
2. Cerpa, A., Elson, J., Estrin D., Girod, L., Hamilton, M., Zhao J.: Habitat Monitoring: Application Driver for Wireless Communications Technology. ACM SIGCOMM Workshop on Data Communications in Latin America and the Caribbean (2001)
3. Xu, Y., Heidemann, J., Estrin, D.: Adaptive energy-conserving routing for multihop ad hoc networks. Tech. Rep. 527, USC/Information Sciences Institute (2000)
4. Perkins, C.: Ad Hoc On Demand Distance Vector (AODV) Routing, Internet-Draft, draft-ietf-manet-aodv-04.txt (1999)
5. Heinzelman, W.B., Chandrakasan, A.P., Balakrishnan, H.: Energy-Efficient Routing Protocols for Wireless Microsensor Networks. Proceedings of the 33rd International Conference on System Sciences (2000)
6. Singh, S., Woo, M., Raghavendra, C.: Power-Aware Routing in Mobile Ad Hoc Networks. Proceedings of the Fourth Annual ACM/IEEE International Conference on Mobile Computing and Networking (1998)
7. <http://basics.eecs.berkeley.edu/sensorwebs>
8. Culler, D.E., Hill, J., Buonadonna, P., Szewczyk, R., Woo, A.: A Network-Centric Approach to Embedded Software for Tiny Devices. DARPA Workshop on Embedded Software
9. Perkins, C., Bhagwat, P.: Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for mobile computers. Proceedings of the SIGCOMM '94 Conference on Communications Architectures, Protocols and Applications (1994) 234–244
10. Park, V.D., Corson, M.S.: A Highly Adaptive Distributed routing Algorithm for Mobile Wireless Network. Proceedings of INFOCOM '97 (1997) 1405–1413
11. Johnson, D.B.: Routing in Ad Hoc Networks of Mobile Hosts. Proceedings of the IEEE Workshop on Mobile Computing Systems and Applications (1994) 158–163
12. Broch, J., Maltz, D.A., Johnson, D.B., Hu, Y-C.: A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols. Proceedings of the Fourth Annual ACM/IEEE International Conference on Mobile computing and Networking, Mobicom '98 (1998) Dallas, Texas, USA
13. Robinson, D.P., Marshall, I.W.: An Iterative Approach to Location of Simple Devices in an Ad-hoc Network, Proceedings of the London Communications Symposium, LCS2002 (2002) 65–9
14. Robinson, D.P., Marshall, I.W.: Location of Simple Nodes in an Ad-hoc Network, Proceedings of Location-Based Services. LBS2002 (2002) 19/1–19/4