Context-based Messaging for Ad Hoc Networks

Daniel Cutting, Derek J. Corbett* and Aaron Quigley†

1. Introduction

Ad hoc networks are spontaneously formed networks of devices such as PDAs and mobile phones that are connected through unreliable and slow wireless links. As the adoption into social situations of such devices increases, it is becoming increasingly desirable to support distributed applications over such networks. When dealing with unstable networks it is preferable to improve robustness by having devices interact with one another without explicitly specifying addresses [4]. Since these networks are likely to cover significant geographical regions and comprise devices that are diverse in both capability and purpose, we propose that addressing devices by their contextual situation may be useful for a range of applications.

For our purposes we define context to (non-exclusively) include spatiotemporal information (location, speed, time of day), identity (users and others in vicinity), user models (profile, schedule, preferences), environmental features (noise, light), social settings (meeting, party), and resources (printers, fax, wireless access, network bandwidth).

Scenarios where this messaging style may be applicable include the following: **A sports stadium** has many inherent contextual clusters (such as groups of team supporters, expensive versus low-cost seats, etc.) and provides a dense ad hoc network of devices with applications including messaging social groups based on common interests, and requesting photographs of the action from specific areas of the stadium. **Fighting a forest fire** could be aided with an ad hoc network formed by the fire trucks and firefighters along fire fronts. Relevant context could include fire hotspots, trucks running low on water, etc.

2. Context-based Messaging

We propose a communication paradigm for ad hoc networks called Context-based Messaging which allows messages to be routed not by the address of the recipient nodes (the *targets*) but by their context. It is assumed that the message does not necessarily need to reach all targets, just “enough”, and at a cost far lower than exhaustively flooding the entire network or maintaining global routing table information as with proactive routing protocols. This indirect approach to addressing is not new in general [1, 3], though addressing based on context is a novel suggestion. Our simplified prototype of this concept (FlavourCast) models context as a single attribute, BLUE, that is applied to some clusters of nodes in a network. Sending a Context-based Message is thus a case of sending a message to as many BLUE nodes in the network as possible while minimising the number of transmissions.

---

*School of Information Technologies, University of Sydney, Australia, email: {dcutting,dcorbett}@it.usyd.edu.au. This work was supported by the Smart Internet Technology CRC.

†Department of Computer Science, University College Dublin, Ireland, email: aquigley@ieee.org.

FlavourCast comprises two independent though interacting algorithms. The first is based on local interaction of nodes and is essentially a network cellular automata. It uses the metaphor of a landscape to construct a topographical map of the network where clusters of nodes with similar properties are represented as valleys. If a node has the BLUE attribute, its height becomes one (the global minimum). Otherwise, its height is the minimum of its neighbours plus one. These straightforward rules result in gradients that extend far from clusters into the network, and stabilise very quickly. When distinct clusters are nearby, a ridge forms at the midpoints (figure 1). The idea of topographies for routing in sensor networks has been examined [3, 2], though the method and purpose of their construction is quite different to our approach.

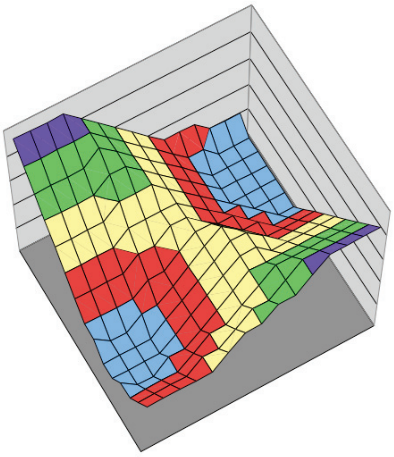

Figure 1. Topographical map.

The second algorithm attempts to deliver a message from a source to as many BLUE nodes as possible. It does this by searching for the minima in the topography generated by the first algorithm. A FlavourCast packet is initially broadcast to all neighbours of the source. Subsequently, each node receiving a message will forward it to at most one neighbour. If the receiver of a message is "lower" than the sender, the message is forwarded downwards to a lower neighbour, ultimately reaching a minimum and BLUE nodes. If the receiver is "higher" than the sender however, the packet is propagated upwards to a higher neighbour until it reaches a peak, after which it is forwarded downwards to an adjacent minimum, in the same manner as described above. This strategy allows messages to be delivered to not only the nearest cluster of BLUE nodes, but also other, more distant clusters. Future work will include improving this algorithm to reach ever more distant clusters while still keeping the algorithm relatively inexpensive in terms of hops.

We have not yet considered mobility in this model, although the reactive nature of the delivery algorithm suggests that it would be reasonably tolerant of nodes moving or failing. However, the topography algorithm is built on the assumption that the general distribution of the network is reasonably stable. A high beaconing frequency would allow the topography to stay up to date, but at high cost; hence the beaconing rate should fall over time when nodes realise they are in stable areas (i.e. when they receive less frequent beacons from known neighbours, and don't receive beacons from new neighbours). If a node knew it was moving, or received beacons from neighbours of which it was previously unaware, it could beacon more frequently.

The FlavourCast prototype so far only simulates nodes with a single attribute. Obviously, a more complete design would need to consider multiple attributes which raises questions of how to designate targets and how to route such messages. Although there are many ways to approach this, one might allow a simple boolean language to designate targets: e.g. send this message to targets that are "BLUE OR GREEN". Unique topographies could be maintained for each attribute in the system and FlavourCast messages could be forwarded to the next node that most fully satisfies the target designation. The topography beacons could be amortised into one for improved efficiency.

3. Results and discussion

Three algorithms were compared to the single-attribute version of FlavourCast. **Flood:** The “worst case”, this algorithm had each node rebroadcast each new message it received ensuring it would reach every target, though at high cost. **SP-multicast:** The “best case”, this took the shortest path from the source to each target guaranteeing delivery to all targets at close to minimal cost. **DR-walk:** Directed random walk (based on [1]) started with a broadcast from the source to all neighbours. At each hop, the current neighbours were appended to a list and the message was forwarded to a random node that was not on the list.

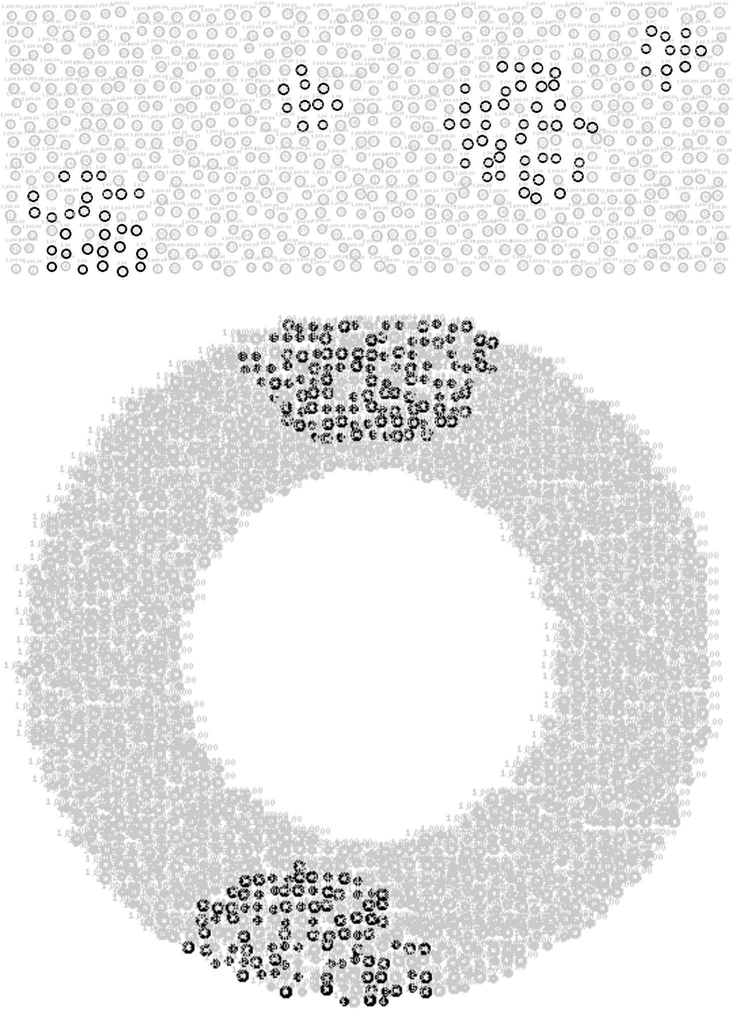

Figure 2. The GRID and STADIUM topologies used in the evaluation.

Using a custom packet-level simulator with simplified physical, link and MAC layers, we tested the algorithms across two networks (figure 2). The first was a rectangular grid of 600 nodes with four BLUE clusters totaling 90 nodes. The second mimicked a stadium of 1,385 nodes with two regions of expensive seating totaling about 210 nodes. In both, each node had approximately six neighbours and were run for five simulated minutes, during which 285 (for GRID) and 260 (for STADIUM) FlavourCasts were initiated from random nodes at a rate of one per second beginning after the beaconing period, which was one beacon per second for the first fifty seconds.

Two metrics were used for evalution: **average % of targets reached** where clearly a high value was preferable and **average number of transmissions per FlavourCast divided by average fraction of targets reached** which gave an overall indication of how “good” an algorithm was. A low value meant that the algorithm reached a relatively high number of targets for the number of transmissions made.

Figure 3(a) shows the average percentage of target nodes reached by each algorithm. As expected, SP-multicast and Flood both reached all target nodes. Interestingly, DR-walk performed much better in GRID than STADIUM. This is because the diameter of STADIUM was greater than GRID, meaning the DR-walk had less opportunity to reach a target node before looping back on itself. FlavourCast performed better in STADIUM than GRID. This was probably because it more frequently found 100% of the targets (since there were only two clusters as compared to four in GRID). Figure 3(b) gives an overall indication of how "good" each algorithm was by finding the ratio of the number of transmissions to the fraction of targets reached. A low value meant it delivered to a relatively large number of targets at low cost. Not surprisingly, Flood performed the worst. Though it always delivered to every target, it did so at high cost. Conversely, SP-multicast had the best score as it always delivered to every target at low cost. According to this metric, FlavourCast performed better than DR-walk on average in both topologies, though the difference was more pronounced in STADIUM where the diameter of the network was greater. FlavourCast also compared favourably to SP-multicast in both topologies.

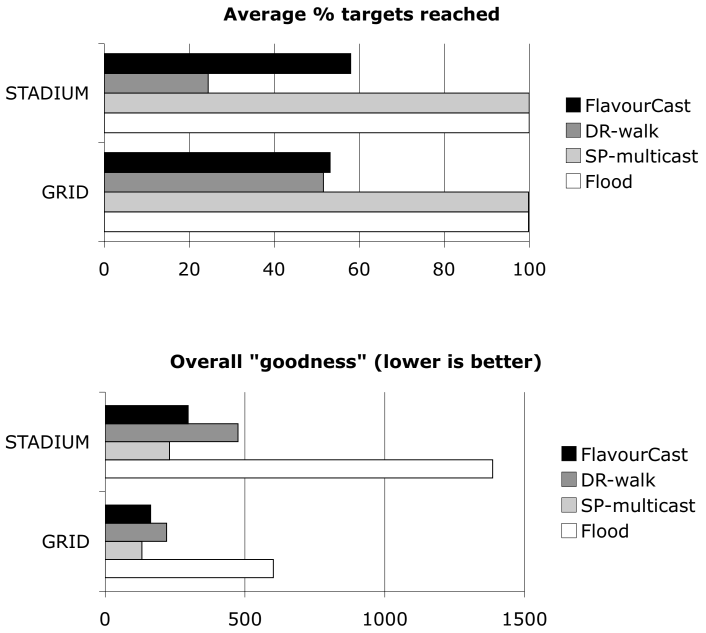

Figure 3. (a), (b). Simulation results.

We expect that Context-based Messaging will be a practical approach to communication within large ad hoc networks and have shown, with a simplified prototype, that it can be at least as competitive as some alternative techniques.

References

[1] D. Braginsky and D. Estrin. Rumor routing algorithm for sensor networks. In *Proceedings of the First Workshop on Sensor Networks and Applications (WSNA), Atlanta, GA*, October 2002.

[2] Jabed Faruque and Ahmed Helmy. RUGGED: RoUting on finGerprint Gradients in sEnsor Networks. Technical report, Department of Electrical Engineering, University of Southern California, 2003.

[3] Chalermek Intanagonwiwat, Ramesh Govindan, Deborah Estrin, John Heidemann, and Fabio Silva. Directed diffusion for wireless sensor networking. *ACM/IEEE Transactions on Networking*, 11(1):2–16, February 2002.

[4] Cecilia Mascolo, Licia Capra, and Wolfgang Emmerich. Middleware for mobile computing (a survey). In E. Gregori, G. Anastasi, and S. Basagni, editors, *Neworking 2002 Tutorial Papers*, pages 20–58, 2002.