An Anycast Service for Hybrid Sensor/Actuator Networks
UNSW Computer Science and Engineering
Technical Report #: UNSW-CSE-TR-0331

Wen Hu
School of Computer Science and Engineering
The University of NSW, Australia
wenh@cse.unsw.edu.au

Nirupama Bulusu
National ICT Australia Limited
nbulusu@cse.unsw.edu.au

Sanjay Jha
School of Computer Science and Engineering
The University of NSW, Australia
sjha@cse.unsw.edu.au

Patrick Senac
ENSICA-LAAS/CNRS
senac@ensica.fr

November 17, 2003

Abstract

This paper investigates an anycast communication service for a *hybrid* sensor/actuator network, consisting of both resource-rich and resource-impoverished devices. The key idea is to exploit the capabilities of resource-rich devices (called micro-servers) to reduce the communication burden on smaller, energy, bandwidth and memory constrained sensor nodes. The goal is to deliver sensor data to the nearest micro-server, which can (i) store it (ii) forward it to other micro-servers using out-of-band communication or (iii) perform the desired actuation. We motivate, propose, evaluate and analyse a reverse tree-based anycast mechanism tailored to deal with the unique event dynamics in sensor networks. Our approach is to construct an anycast tree rooted at each potential event source, which micro-servers can dynamically join and leave. Our anycast mechanism is self-organizing, distributed, robust, scalable, routing-protocol independent and incurs very little overhead. Simulations using ns-2 show that our anycast mechanism when added to Directed Diffusion can reduce the network's energy consumption by more than 50%, can reduce both the mean end-to-end latency of the transmission and the mean number of transmissions by more than 50%, and achieves 99% data delivery rate for low and moderate micro-server mobility rate.

1 Introduction

This paper investigates an anycast service for hybrid sensor/actuator networks. In the last couple of years, sensor networks research has addressed the development of sensor platforms[12], application domains, and algorithms. Because sensor networks depend on multiple nodes cooperating with each other, an effective communication paradigm is of prime importance and has been researched upon[17][13][22][16].

Previous work addressing communication in sensor networks has argued that the key challenge in supporting small, low-powered nodes in ad hoc multi-hop sensor networks, are scalable and energy-efficient mechanisms for data dissemination. Particularly noteworthy amongst numerous proposed data dissemination paradigms are: (i) Directed Diffusion[13], a general purpose, network-oriented approach to data-centric communication in sensor networks (ii) IDSQ[16], an information-oriented approach that combines data routing with information optimization techniques, and (iii) TAG[17], a database oriented approach to address numerous sensors in aggregate by means of SQL queries and gather the data back to a single, central server.

History shows that there are normally different types of network devices in large scale networks. For example, today's Internet combines different devices such as routers, servers and hosts, even the routers can be classified into different categories (e.g., into core routers and edge routers). For the large scale sensor networks that may have thousands of nodes in the future, it is more realistic to have hierarchical models of network devices rather than flat ones.

Previously proposed data routing protocols for sensor networks have not been designed to leverage the capabilities of hybrid devices by exploiting resource-rich devices to reduce the communication burden on smaller, energy, bandwidth, memory and computation-constrained sensor devices. Consequently, they may not be best suited for several applications of such hybrid sensor networks, which involve a multitude of mutually cooperative sinks. Our hypothesis is that an anycast service can provide significant improvements to the aforementioned data dissemination protocols for such applications and networks. The anycast service should be useful to any application involving a hybrid of resource-rich specialized nodes with small sensor devices. The resource-rich nodes provide some service such as (i) long-range data communications, (ii) persistent data storage, or (iii) actuation. Examples of actuation would be re-charging or replacing small nodes whose energy has been depleted, imagers which can take photos or video when activated by sensors, sprinklers which can sprinkle water in badly parched areas etc. The resource-rich node acts as a sink, and we call it a micro-server.[1] The intuition is that you only care for the service, not which

[1] The term micro-server was suggested by Deborah Estrin.

server provides it.

For example, in a battlefield of mobile soldiers; the soldiers may be equipped with more powerful data transmitters (out of band higher-range radios) than sensors. It may be more effective to forward the information (e.g. enemy detection, land mind presence, convoy vehicles) to the nearest available soldier, who can forward it to the other soldiers, instead of to all the soldiers in the field. In a disaster recovery scenario, several biochemical sensors may have been scattered. and multiple imagers (aerial or robotic) may be navigating the terrain. When biochemical sensors detect a toxic plume, this message just needs to go to the nearest imager (rather than a specific imager) which can act accordingly.

We wish to design an anycast service that can extend system lifetime, reduce end-to-end latency and improve network scalability. The design goals for such an anycast service are as follows:

- *Simple:* To accommodate the limitations of small sensor nodes, the anycast mechanism must be computation and memory efficient.
- *Energy-efficient:* The anycast mechanism must incur minimal energy overhead for control communications, while optimizing energy usage for data communications.
- *Self-organizing and Adaptive:*To be responsive to sinks joining and leaving dynamically; and robust to node failures the anycast mechanism must be self-organizing and adaptive.
- *Routing-protocol independent:* To be able to implement on the top of many routing-protocols of sensor networks.
- *Distributed:* To be scalable to large sensor networks, the anycast mechanism must be distributed rather than relying on any central infrastructure.

Contributions: It is challenging to design a routing-protocol independent anycast service that is simple to implement and incurs low overhead, while also being self-organizing and robust. Our approach is to construct an anycast tree rooted at each event source, which micro-servers can dynamically join (by flooding route discovery interests) and leave. Data is delivered to the nearest micro-server on the tree. We motivate the need for anycast service and propose a tree-based anycast mechanism (Section 3). We evaluate our anycast scheme using extensive simulations that demonstrate its benefits in conserving energy, improving latency (Section 4). Finally, we suggest potential strategies for formal analysis for our anycast mechanism (Section 5).

2 Related Work

Numerous data dissemination protocols have been proposed for sensor networks in particular, and more generally for the Internet and ad hoc networks. In this section, we cover research most directly relevant to our approach.

Directed Diffusion: Directed Diffusion (see [13] for more details) is a data-centric, reverse-path based communication paradigm for sensor networks. Sinks flood their *interests* into the network when they join the network. An interest is a query specifying the attributes of the information a sink wants a sensor to collect and respond. Sources in turn flood the first few exploratory data packets into the network. Sinks select and reinforce the best paths and the sources use reverse best paths to deliver data back to the sinks. The sink chooses the neighbor from which the first discovery-packet is received as the immediate upstream node. Therefore, it can minimize the per-node network state maintenance and achieve highly-efficient data dissemination in sensor networks.

The initial packet-flooding is not efficient for dense and large scale networks. Therefore, *one-phase-pull* and *push* [11] algorithms have been introduced to reduce the flooding. Mechanisms to further reduce the scope of the flooding are being studied[9]. The sources choose the best paths by judging the arrival of the interest packets in one-phase-pull algorithm. Sources flood exploration-data and sinks choose the best path by judging the arrival of these data packets in push Directed Diffusion algorithm.

Simulations show that one-phase-pull works well for scenarios involving multiple sources with a single sink, whereas the push mechanism is better suited for multiple sinks with a single source. However, scenarios involving *both* multiple sinks and multiple sources have not been considered. Furthermore, the network still delivers replicated data copies to multiple sinks and thus may introduce unnecessary consumption of precious energy in tiny sensors in some applications. These considerations inspire the design of our algorithm.

Two-tier Data Dissemination (TTDD): Two-tier data dissemination mechanism [22] tries to set up a virtual grid by calculating the distance between sensors and relaying spots. The sensor with minimum distance becomes a relaying point. The sources broadcast their query/interest within the grid and the query/interests are forwarded by the relaying sensors to the sources. The sources transfer the data packets along the reverse path to the sinks. Compared to Directed Diffusion, it can better handle sink mobility because the query/interest is limited in one local grid. However, it may still introduce replicate data packets transmission to multiple sinks.

Manycast: Manycast[5] is a recently proposed group communication scheme for ad hoc networks. They share our objective of providing an efficient communication scheme when many destinations are interested in the same data. However manycast allows a source to communicate with many destinations *simultaneously*. This idea could also be applied to sensor networks; however since data transmissions would occur on the in-band communication channel, it is not very energy-efficient. We believe that our approach of getting data to the nearest sink, and then forwarding it to other sinks using out-of-band communication is more suited to large sensor networks.

Internet anycast: The Internet community has addressed anycast research extensively [10] [14]. However, the environment is radically more dynamic in sensor networks; and sensor nodes have significantly limited resources which must be addressed.

Multi-robot coordination: Within the field of distributed mobile robotics, Daniela Rus *et al* [15] have addressed the problem of maintaining continuous communication to route data amongst mobile robots. Their work is complementary to ours; once data reaches the nearest micro-server using our anycast mechanism; such techniques may be used to forward the data to other mobile nodes.

Summary: Previous research in sensor networks communication has not exploited hybrid device capabilities such as out-of-band communication and not explored anycast services for sensor networks. Previous anycast mechanisms proposed in the Internet and ad hoc networks context, do not work effectively in the sensor networks domain with unique energy and memory constraints and event dynamics (sources are a function of events in the system). Our novel reverse tree-based anycast mechanism for sensor networks is tailored to deal with the unique constraints and event dynamics in sensor networks, which we describe next.

3 Tree-Based Anycast

In this section, we describe the underlying assumptions, design rationale and details of our anycast mechanism.

3.1 Design Rationale

We assume a *hybrid* sensor network which consists of both resource-rich micro-server nodes and low-power sensor nodes. Further we assume that there are multiple micro-servers (sinks) interested in the same data. Sinks could be mobile. Data needs to only reach one sink, thus motivating an anycast service. We assume that sensor network applications can handle small amounts of data loss; and therefore anycast does not need to explicitly provide reliable data delivery.

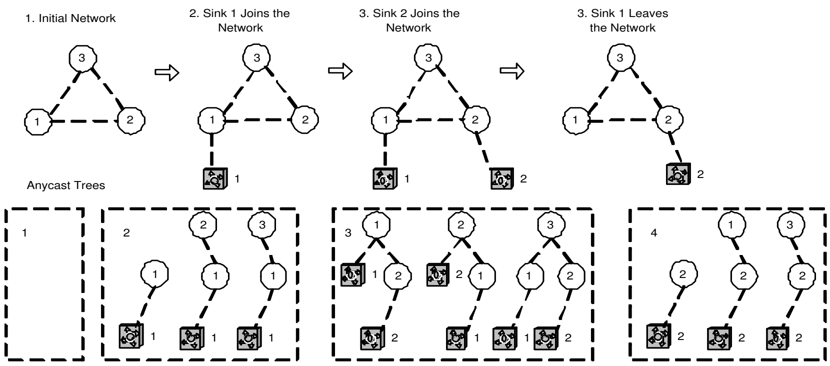

Figure 1: Illustration of the anycast mechanism. The lower, boxed pictures show the structure of each anycast tree as two sinks join and leave a sensor network.

We want to provide an anycast service that is scalable, self-organizing, robust, simple and energy-efficient. A straight-forward approach to implement anycast is using an *expanding-ring search* with feedback from micro-servers. In this approach, the source floods data with a restricted scope. Initially the hop count is set to 1, and incremented in steps of 2 until the source locates a potential sink. This sink sends a feedback message to the source, whereupon the hop-count is retained and flooding is ceased. This is attractive because it is self-organizing and robust, requires minimal network state and can limit the flooding scope in diffusion. On the other hand it is not well suited to handle sink mobility. It incurs high latency and energy overhead if a sink leaves (moves away) because it must discover a route to an alternate nearby sink. Moreover, it may require coordination and synchronization amongst sinks before sending feedback to the event source.

Instead we adopt a shared tree approach. Corresponding to each event source, a *shortest-path tree* rooted at the source is constructed. Sinks form the leaves of the tree. Sinks can dynamically join or leave the anycast tree. Although this approach requires more network state, it is a good approach to handling mobility, as it simultaneously maintains paths to all sinks. By eliminating the need to discover paths to alternate sinks each time a sink leaves, it can reduce worst-case latency and does not require synchronization among sinks.

3.2 Algorithm Details

We now describe the details of our tree-based anycast mechanism. Figure 1 illustrates how the structure of each anycast tree evolves when two sinks join and leave a sensor network.

(Reverse) Tree Formation: Every sensor node forms a potential event source.[2] Therefore, corresponding to every sensor node i in the network, there is an anycast tree T_i rooted at that sensor node. Each anycast tree is built from the leaves to the root. When sink S enters the network, a new branch leading to the sink must be added to each anycast tree. To minimize sensor energy consumption, information for building the branches is *piggybacked* with the route-discovery packet (e.g. interest packets of Directed Diffusion). To calculate the cost of the branch, sink initializes a *cost field* c (e.g. hop-count) in the route-discovery packet. [3] Upon receiving this packet from sink S, each node i updates its anycast table by setting cost $c(i, S)$ of the branch to S to be c. It increments the cost c before forwarding this packet. Eventually, a new branch (with cost) to sink S is added to each tree. To handle the memory constraint of sensors, an upper-bound can be added to limit anycast table size.

Sink Leaves: There are two ways to handle sinks leaving.

- *Explicit Leave:* The sink floods a leave packet into the network when it leaves — the related anycast entry is deleted when a node receives the packet.
- *Timer Based:* Associate a timer with each anycast table entry — an anycast entry is deleted when the timer expires.

Although the explicit leave approach can respond more quickly to sink mobility, it also consumes more energy than the timer based approach. We adopt the latter approach.

Data Delivery and Path Maintenance: After initial set up, when a data packet arrives, a sensor looks up its anycast table for the sink with minimum cost before it forwards the packet. Therefore, the packet will be sent to only the nearest sink instead of to all sinks (as would be the case with Directed Diffusion). Sinks periodically send packets to refresh the anycast table entries. Stale entries are deleted when the related timer expires.

Sink Mobility: A sink may move out of range of its immediate upstream node. Some data packets may be delivered along this route until an (i) alternate route to this sink or (ii) an alternate route to an alternate nearby

[2] If data across multiple sources exhibits spatio-temporal correlation, some aggregation mechanisms based on application characteristics can be exploited to transmit data from a single node.

[3] We use hop-count as the metric for computing the shortest-path tree. However, other metrics like sensor-energy level and path-energy-consumption can be used as per application requirements.

sink are found during next periodic route refresh. Our anycast algorithm does not implement delivery reliability explicitly because most sensor applications are inherently loss-tolerant to small amounts of loss. Reliability can always be implemented at application layer if necessary.

Scalability: The size of the anycast table in each sensor node is independent of the number of sources. Nodes always deliver packets to the nearest sink regardless of the source. The anycast table size would increase linearly with the number of sinks in the network. We observe that since we are only interested in delivering packets to nearest sink, it is not necessary to maintain paths to all sinks in every anycast table. Therefore, the size of the table can be limited. We limit the size to 3 in our simulations.[4] This enables our algorithm to be scalable in terms of the number of sources and the number of sinks. Moreover, it enables us to accommodate the memory constraints of small sensor devices. The sensor network has shorter paths and less number of packet-transmissions with anycast. This can reduce packet-collisions and delivery-latency.

Differences from Internet Anycast: We briefly summarize the key differences between our tree-based anycast mechanism and those proposed previously for the Internet. The servers (leaves) join before the client/host (root) in the Internet. Therefore, the anycast tree is built when the leaf joins the tree. In sensor networks, it is more dynamic as event sources (root) could join the tree earlier or later than micro-servers. The tree is built via a reverse approach of route discovery (interest flooding) in our algorithm. Moreover, the hop-count information is stored in the routing tables of Internet routers. A tree can be built by routing table look up — this is one of the reasons Internet anycast is not scalable and motivates research (GIA [14]) to reduce the router state information. In sensor networks, hop-count information must be built from scratch. We piggyback this information with the route discovery packet (Interest) to save energy. Finally, in sensor networks, the size of the anycast routing table must be limited due to memory and computation constraints of small nodes; which is not a consideration in Internet anycast.

4 Evaluation

To evaluate the performance of anycast algorithm, we implemented it in the ns-2 [1] simulator and compare it to diffusion without anycast. The results of the simulations show that anycast algorithm can significantly reduce overall network energy consumption and transmission latency. Although we

[4]A future extension would be for the nodes to dynamically learn how many entries to keep.

choose to implement our algorithm on the top of Directed Diffusion in our simulations, it is not difficult to implement it on the top of other protocols like TTDD[22].

4.1 Goals, Metrics and Methodology

The goals of our simulation-based evaluation are to study whether anycast can (i) lead to significant energy savings in comparison to Directed Diffusion (ii) improve the end-to-end latency in data transfer (iii) improve network scalability, and (iv) handle moderate sink mobility.

We use several metrics for evaluation.

- *Mean energy consumption:* We study this metric as a function of time. This metric characterizes the mean energy consumed per node at any given instant of time. Ideally, the energy consumption should be as low as possible.
- *Jitter in energy consumption:* The jitter percentage for each node is the positive or negative percentage difference between the energy consumed by the particular node and the mean energy consumption. Ideally, the jitter should be close to zero so as to load balance energy consumption equally across all nodes. We study jitter across all nodes.
- *End-to-end latency:* We study this metric as a function of network size. This metric characterizes the cumulative latency for data to reach from its source to its destination. Ideally, this metric should be as small as possible to indicate timely data transfer.
- *Mean Path Length:* We study this metric as a function of network size. This metric denotes the mean number of hops traversed by a data packet. Ideally, this metric should be as small as possible for lower energy consumption across small nodes.
- *Data delivery rate:* We study this metric as a function of sink speed. This metric characterizes the percentage of event source data packets successfully delivered to at least one sink. Ideally, this should be 100%.

4.1.1 ns-2 implementation

We extend the Directed Diffusion implementation in ns-2 to incorporate the tree-based anycast mechanism. Our extension enables Directed Diffusion to build and maintain the anycast tree and deliver the data packet to exactly one of the sinks. We added an anycast filter to Directed Diffusion routing in ns-2. The anycast filter is implemented by the following four C++ classes: anycast_routing class handles the anycast routing functions and anycast_attr class handles anycast attributes (e.g sinks, costs) anycast_tools class handles

Nodes	Area	Sinks
50	350m × 350m	2
100	500m × 500m	2
150	600m × 600m	3
200	700m × 700m	4
250	790m × 790m	5
300	860m × 860m	6

Table 1: Simulation Parameters. The area size is set such that there are two sensors in each communication unit.

the anycast table manipulations and path_log class handles the logging of data packet transmissions.

4.1.2 Simulation parameters

To understand how our anycast algorithm can affect the above network metrics, we simulated topologies with varying network sizes — 50, 100, 150, 200, 250 and 300 sensor nodes. Sensors are randomly deployed in a two dimension area. Three sources are chosen randomly for each scenario. [5]

Directed Diffusion is chosen as the routing algorithm and IEEE 802.11 is chosen as the MAC level protocol. The transmission range is 100m; the initial energy in the sensors is 1000 Joules.[6] Other simulation parameters are summarized in Table 1.

To study the impact of sink mobility, 100 sensors (with 3 sources) and three sinks are randomly deployed in an area of 500m * 500m. Sinks move at speeds of 1m/s, 3m/s, 5m/s, 8m/s, 10m/s, 15m/s, 20m/s, 30m/s, 40m/s, 50m/s and 100m/s respectively.

Simulation time is 1002 seconds (which is sufficient to characterize protocol trends). Within the first 20 seconds, sources and sinks publish or subscribe the same interest in a random sequence. After publication, a source generates a data packet every three seconds. Sensor energy levels are logged every 10 seconds.

4.2 Results

Figures 2, 3, 4 plot the mean energy consumption as a function of time. Energy consumption with anycast increases at a much slower rate; and is at least 60% lower with anycast (for 300 nodes) when simulations end. Anycast savings are more significant for larger network sizes (upto 60% for 300 nodes

[5] We use fewer sources in our simulation as clustering mechanisms such as [22] will be able to select a representative for data aggregation.

[6] We intend to study the performance of our algorithms with special MAC protocols customized for sensor networks in the future, for example, S-MAC.

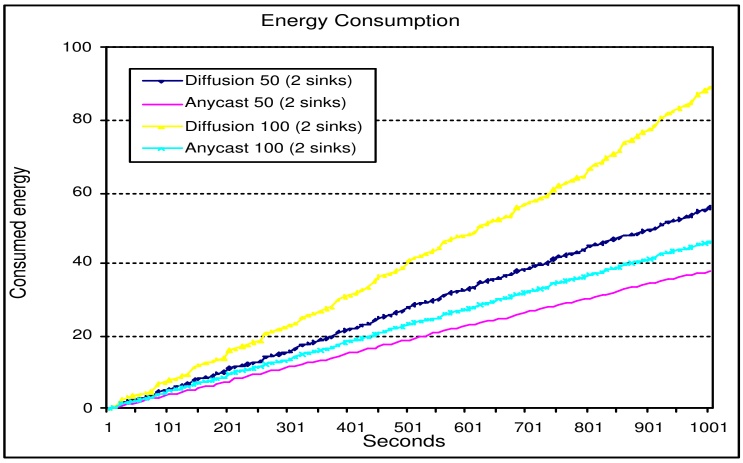

Figure 2: Energy Consumption of 50/100 nodes in 350 * 350/500 * 500 areas (2 sinks and 3 sources).

scenario) than for smaller network sizes (upto 25% for 50 nodes scenario) . The results show that anycast mechanism enables energy savings for two reasons: (i) data is delivered to only one of the sinks instead of all the sinks (ii) the data delivery path is shorter so that fewer data transmissions per path are required.

Figure 5,6 plot the jitter with Directed Diffusion and anycast respectively across all nodes. Jitter is significantly lower with anycast (mainly between −15% and 10%) than with Directed Diffusion (mainly between −40% and 20%) because anycast forwards the data traffic locally to the nearest micro-server for each source; thereby distributing the load more evenly across the network when compared to Directed Diffusion where data traffic is global and burdens nodes in the middle of the network more heavily.

Figure 7 plots the end-to-end latency (transmission delay) as a function of network size. As expected, the end-to-end latency for both Directed Diffusion and anycast increases with the network size. However, the increase is significantly less with anycast because the mean path length is smaller as data is only forwarded to the nearest sink. For about 300 nodes, the end-to-end latency with anycast is nearly 60% lower than with Directed Diffusion.

Figure 8 plots the mean path length as a function of network size. Again, the mean path length (which is related to the end-to-end latency) for both anycast and Directed Diffusion increase with network size. However, the increase is more gradual for anycast. The mean path length of Directed Diffusion grows to 15 hops for 300 nodes; whereas for anycast the corresponding number of hops is just 6 (again nearly 60 % lower). These results show that the major contribution to reducing mean end-to-end latency are the reduced

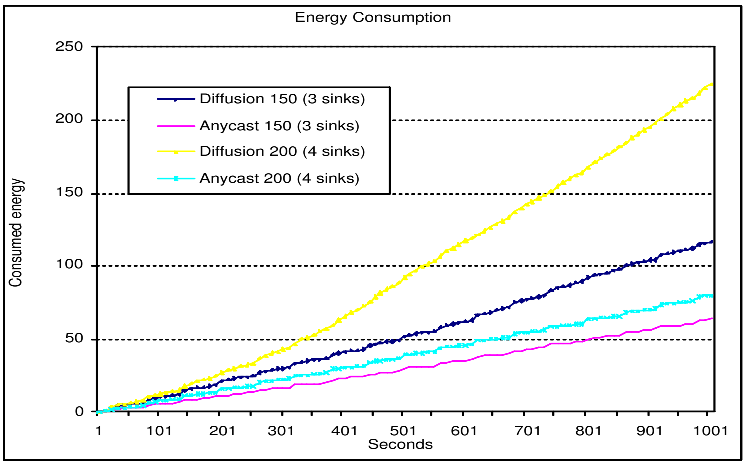

Figure 3: Energy Consumption of 150/200 nodes in 600 * 600/700 * 700 areas (2 percent sinks and 3 sources).

mean path lengths. Figure 9 plots the data delivery rate as a function of sink speed for a network topology of size 100 nodes. We expect human and robotic mobility for many sensor network applications to be approximately 1 m/s (3.6 kmph). Our current anycast algorithm achieves 99% data delivery rates for this mobility regime, and can therefore accommodate it. However, the data delivery rate for anycast drops off to 73% as the sink speed increases to 20m/s. We conjecture that the subsequent increase in data delivery rate for higher sink speeds is an artifact of limited-size terrain. A complete study of mobility effects on anycast is beyond the scope of this paper. However, we are exploring how to extend our scheme to accommodate high-mobility micro-servers.

4.3 Discussion of Results

To summarize, our simulations show that our anycast service when added to Directed Diffusion can: (i) significantly reduce end-to-end latency (ii) significantly reduce energy consumption (iv) balance network load (energy consumption) more evenly by forwarding data traffic locally rather than globally, and (iv) handle low to moderate sink mobility with minimal extensions (as evidenced by the high data delivery rates achieved) but may require further modifications to handle higher mobility rates.

There are two important caveats to our simulation results. The first is that while radio links work very well in simulation; they can be notoriously lossy in practice. These results need to be further validated experimentally. The second is that while simulations show that anycast improves network

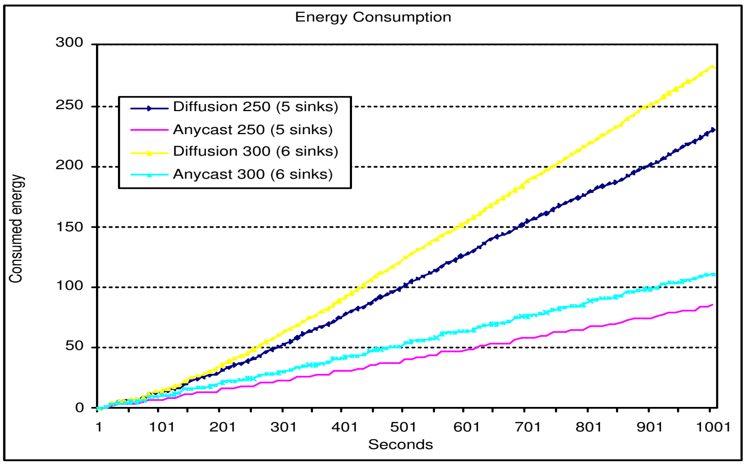

Figure 4: Energy Consumption of 250/300 nodes in 790 * 790/860 * 860 areas (2 percent sinks and 3 sources).

scalability and end-to-end latency; they cannot completely verify reachability properties. This requires formal analysis techniques, which we address next.

5 Formal Analysis

The complexity of communication protocols such as the anycast protocol introduced in this paper which combine a potentially large number of distributed entities that have to satisfy synchronization and communication constraints on top of an error prone and asynchronous network service underline the limits of the classical simulation approach for validating the fundamental properties of this type of protocols. Indeed the simulation approach, while able to give some helpful hints on the behaviour and performances of a protocol exposed to a wide variety of network conditions, is of little help when one need to verify exhaustively the fundamental properties of a protocol. Conversely, the role and contribution of formal modelling in verification and simulation of distributed systems is not questionable [8]. Formal techniques by enabling to focus on an abstract model based on a sound mathematical foundations allow one to verify liveness, reachability and boundness properties (among others) that are of great concern for the anycast protocol introduce in this paper. However, when confronted with the modelling of a large scale distributed system, such as a sensor network, a formal model is exposed to the generation of a combinatorial explosion of its underlying state graph making difficult if not impossible an exhaustive analysis of the system. Even in these adverse situations formal approaches are

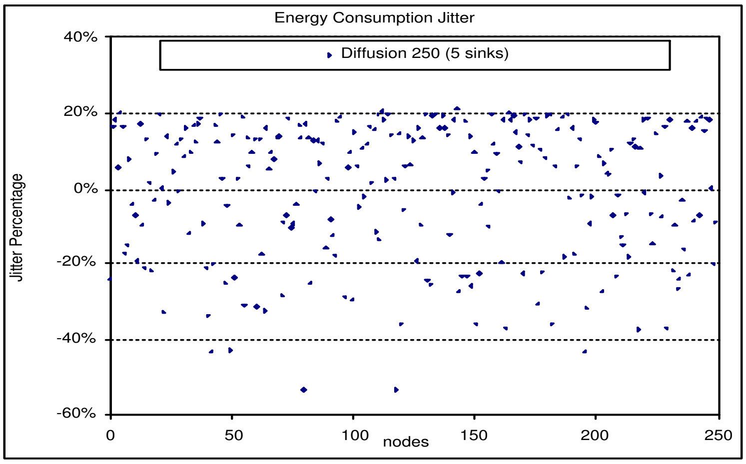

Figure 5: Energy Consumption Jitter of 250 nodes with Directed Diffusion in a 790 * 790 area after 1002 seconds (5 sinks and 3 sources).

able to perform simulation techniques at high abstraction level that makes easier identification of protocol flows.

Nevertheless, while composed of a large number of nodes, the nodes of a sensor network perform elementary and similar tasks that make possible state graph reduction by applying appropriate reduction-projection and folding techniques [18][4].

Petri net and process algebra such as Lotos are the two great families of formal methods that have been intensively used for protocol modelling and verification[21][6]. We have demonstrated in previous works the contribution of these two approaches for the modelling and verification of application and transport layer protocols[19][20]. These contributions have been accompanied by the definition of methodologies and the design of toolkits that offer a powerful support to the specification and verification phases[7][3].

We plan to apply these two approaches to the modelling and design of the considered anycast protocol. More generally, we aim to define a formal methodology for the modelling and verification of protocols for sensor networks.

6 Future Work

There remain several areas of future work.

High mobility: Our current anycast service can accommodate micro-servers with low and moderate mobility rates. While it is not clear at this stage as to which sensor applications will involve very high mobility, we are exploring how to extend our scheme to accommodate high-mobility micro-

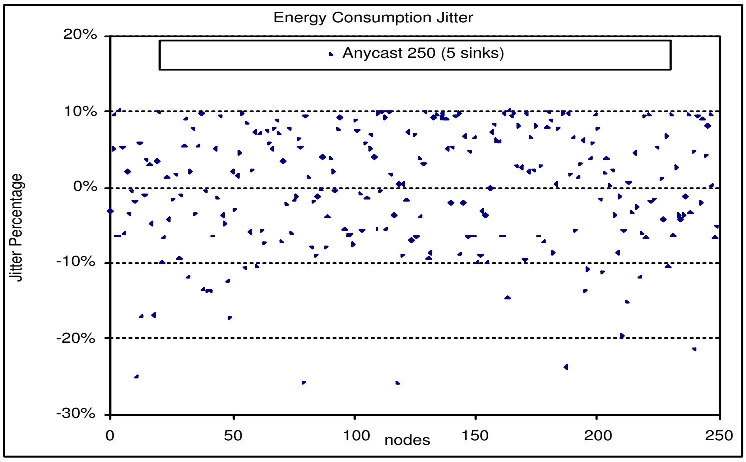

Figure 6: Energy Consumption Jitter of 250 nodes with anycast in a 790 * 790 area after 1002 seconds (5 sinks and 3 sources).

servers.

Metrics for Path Selection: We have used hop-count as the network path metric in our implementation. It would be desirable to make the service energy-aware since energy is such an important issue in sensor networks.

Source Anycast to Data Anycast: Our anycast mechanism in its current form provides support for source anycast, *i.e.,* when data originates at a single source. It is possible for information to be correlated among multiple sources. We are exploring extensions to address this issue.

Interaction with other data dissemination protocols: We have studied how anycast interacts with Directed Diffusion. To fully understand anycast benefits in a sensor network scenario, we wish to study how anycast interacts with other data dissemination protocols such as Two-Tier Data Diffusion and compare anycast performance to Manycast over lossy links.

Linear Programming Formulation: Since anycast depends on exploiting resource-rich devices such as microservers which function as data sinks, we are looking at a linear programming formulation with a multi-objective function (eg., energy, latency) for objective placement of micro-servers.

Formal and Experimental Validation: Finally, we are working on a more detailed formal analysis of the anycast service based on the methodology proposed in Section 5. Since theory and simulations show substantial improvements with anycast, we are planning to implement it with a hybrid mix of motes[12] and PLEB[2] devices in Australia.

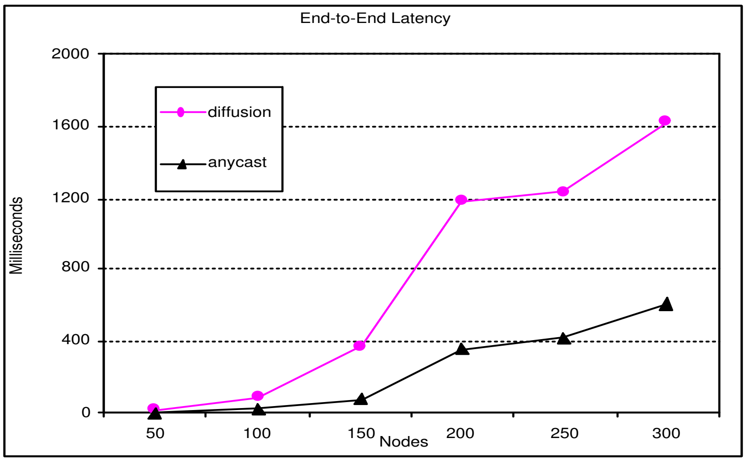

Figure 7: Mean end-to-end transmission delays.

7 Conclusions

In this paper, we proposed and evaluated a tree-based anycast mechanism for applications of *hybrid* sensor networks that is protocol-independent, self-organizing, distributed, robust, scalable, and incurs very little overhead. The key idea was to construct an anycast tree rooted at each event source, which micro-servers can dynamically join (by flooding route discovery interests) and leave. Data is delivered to the nearest micro-server on the tree. We exploit the out-of-band communication of micro-server nodes to forward the data to other micro-servers, if necessary.

Our evaluations demonstrate the significant benefits of an anycast service. In particular, we noticed a significant reduction (over 50% for the simulated scenarios) in end-to-end latency, mean energy consumption, and number of data transmissions. Moreover, anycast maintains relatively high data delivery rates for low and moderate sink mobility speeds corresponding to the situations in which these sensor networks will be deployed.

Anycast could potentially improve scalability of protocols like Directed Diffusion especially in large scale sensor networks with hybrid devices. Although we have compared our anycast service with Directed Diffusion, the mechanism itself is protocol-independent and can be applied generally.

8 Availability

For verifiability, simulation code and traces will be made available shortly at:
`http://www.cse.unsw.edu.au/~wenh/anycast`

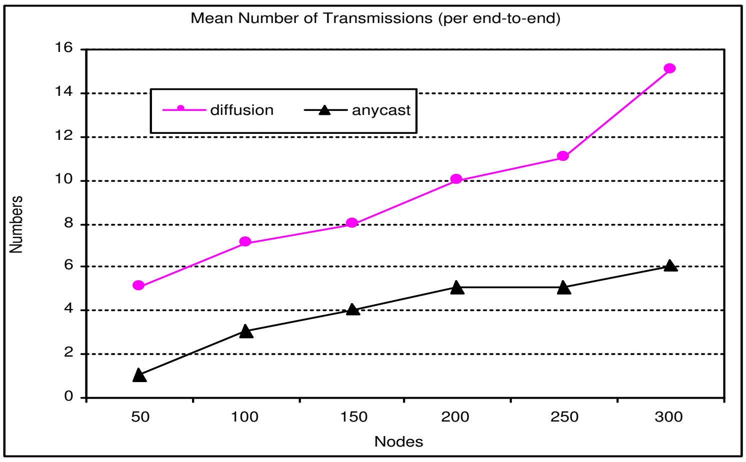

Figure 8: Mean number of transmissions per end-to-end path (Mean path length).

9 Acknowledgements

The authors wish to thank Dr. Huan Pham, Michael Liu, Tatiana Bokareva and William Lau for their feedback. This research was supported by the Australian Research Council.

References

[1] The network simulator - ns-2.

[2] The pleb project.

[3] B. Berthomieu, P-O. Ribet, and F. Vernadat. L'outil tina – construction d'espaces d'tats abstraits pour les rseaux de petri et rseaux temporels. In *Modlisation des Systmes Ractifs, MSR'*, 2003.

[4] B. Berthomieu and F. Vernadat. State class constructions for branching analysis of time petri nets. In *9th International Conference TACAS*, 2003.

[5] Casey Carter, Seung Yi, Prashant Ratanchandani, and Robin Kravets. Manycast: exploring the space between anycast and multicast in ad hoc networks. In *Proceedings of the 9th annual international conference on Mobile computing and networking*, pages 273–285. ACM Press, 2003.

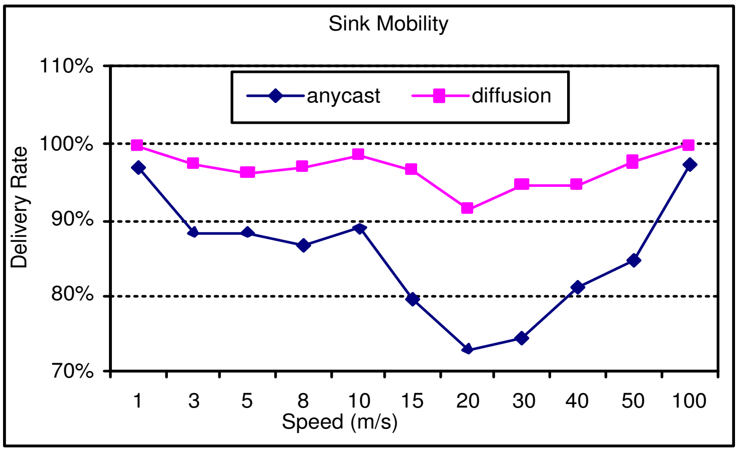

Figure 9: Delivery rate vs mobile sinks' speed with 100 nodes in an area of 500 * 500

[6] J.P. Courtiat, M. Diaz, R.C. De Oliveira, and P. Snac. Formal models for the description of timed behaviours of multimedia and hypermedia distributed systems. *Computer Communication*, 1997.

[7] Pierre de Saqui-Sannes, Ludovic Aprvrille, Christophe Lohr, Patrick Snac, and Jean-Pierre Courtiat. Uml and rt-lotos an integration for real time system validation. *European Journal of Automated Systems*, 2002.

[8] Michel Diaz. Protocol specification, testing and verification v. In *Proceedings of the IFIP WG6.1 Fifth International Conference on Protocol Specification, Testing and Verification*, 1985.

[9] Deborah Estrin and Henri-Dubois Ferreire. Personal communication. 2003.

[10] Network Working Group. Host anycasting service. *IETF RFC 1546*, 1993.

[11] John Heidemann, Fabio Silva, and Deborah Estrin. Matching data dissemination algorithms to application requirements. In *Proceedings of the First ACM Conference on Embedded Networked Sensor Systems (SenSys).*, 2003.

[12] Jason Hill, Robert Szewczyk, Alec Woo, Seth Hollar, David Culler, and Kristofer Pister. System architecture directions for networked sensors. In *Proceedings of the ninth international conference on Architectural*

support for programming languages and operating systems, pages 93–104. ACM Press, 2000.

[13] Chalermek Intanagonwiwat, Ramesh Govindan, Deborah Estrin, John Heidemann, and Fabio Silva. Directed diffusion for wireless sensor networking. *IEEE/ACM Transactions on Networking (TON)*, 11(1):2–16, 2003.

[14] Dina Katabi and John Wroclawski. A framework for scalable global ip-anycast (gia). In *Proceedings of the conference on Applications, Technologies, Architectures, and Protocols for Computer Communication*, pages 3–15. ACM Press, 2000.

[15] Qun Li, Michael De Rosa, and Daniela Rus. Distributed algorithms for guiding navigation across a sensor network. In *Proceedings of the 9th annual international conference on Mobile computing and networking*, pages 313–325. ACM Press, 2003.

[16] Juan Liu, Feng Zhao, and Dragan Petrovic. Information-directed routing in ad hoc sensor networks. In *Proceedings of the 2nd ACM international conference on Wireless sensor networks and applications*, pages 88–97. ACM Press, 2003.

[17] Samuel Madden, Michael J. Franklin, Joseph M. Hellerstein, and Wei Hong. Tag: a tiny aggregation service for ad-hoc sensor networks. *ACM SIGOPS Operating Systems Review*, 36(SI):131–146, 2002.

[18] Pierre-Olivier Ribet, Frangois Vernadat, and Bernard Berthomieu. On combining the persistent sets method with the covering steps graph method. *FORTE*, 2002.

[19] L. Rojas, E. Chaput, L. Dairaine, P. Snac, and Michel Diaz. Video transport over partial order connections. *the Journal of Computer Networks and ISDN Systems*, 1998.

[20] L. Rojas, P. Snac, L. Dairaine, and M. Diaz. Towards a new generation of transport services adapted to multimedia applications. *Annals of Telecommunication*, 1999.

[21] P. Snac, M. Diaz, A. Lger, and P. de Saqui-Sannes. Modeling logical and temporal synchronization in hypermedia systems. *the IEEE Journal on Selected Areas on Telecommunication*, 1996.

[22] Fan Ye, Haiyun Luo, Jerry Cheng, Songwu Lu, and Lixia Zhang. A two-tier data dissemination model for large-scale wireless sensor networks. In *Proceedings of the eighth annual international conference on Mobile computing and networking*, pages 148–159. ACM Press, 2002.