REPORT DOCUMENTATION PAGE	*Form Approved* **OMB No. 0704-0188**

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Washington Headquarters Service, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington, DC 20503.

PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.

1. REPORT DATE *(DD-MM-YYYY)* NOVEMBER 2009	**2. REPORT TYPE** Conference Paper Postprint	**3. DATES COVERED** *(From - To)* May 2008 – October 2008

4. TITLE AND SUBTITLE AN APPLICATION SPECIFIC ROUTING FRAMEWORK FOR WIRELESS SENSOR NETWORKS	**5a. CONTRACT NUMBER** IN HOUSE
	5b. GRANT NUMBER N/A
	5c. PROGRAM ELEMENT NUMBER 62702F
6. AUTHOR(S) Mukundan Venkataraman, Mainak Chatterjee, and Kevin Kwiat	**5d. PROJECT NUMBER** 4519
	5e. TASK NUMBER 22
	5f. WORK UNIT NUMBER 49

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

AFRL/RIGG
525 Brooks Road
Rome, NY 13441-4505

University of Central Florida
Electronic Engineering and Computer Science Dept.
Orlando, FL 32816

8. PERFORMING ORGANIZATION REPORT NUMBER

N/A

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

AFRL/RIGG
525 Brooks Road
Rome NY 13441-4505

10. SPONSOR/MONITOR'S ACRONYM(S)

N/A

11. SPONSORING/MONITORING AGENCY REPORT NUMBER

AFRL-RI-RS-TP-2009-58

12. DISTRIBUTION AVAILABILITY STATEMENT

Approved for public release; distribution unlimited PA# WPAFB-2008-3568 Date Cleared: 6-June-2008

13. SUPPLEMENTARY NOTES

This paper was submitted and accepted for publication in the Proceedings of the 2nd IFIP: International Symposium on Wireless Communications and Information Technology in Developing Countries, Pretoria, South Africa, 6-7 October-2008. One or more of the authors is a U.S. Government employee working, within the scope of their Government job; therefore, the U.S. Government is joint owner of the work and has the right to copy, distribute, and use the work. All other rights are reserved by the copyright owner.

14. ABSTRACT

Numerous routing protocols have been proposed for wireless sensor networks, each of which is highly optimized for a certain class of traffic, like real time, reliable sense and disseminate network reprogramming, energy efficiency and so on. However, a typical deployment demands an arbitrary communication pattern that generates multiple traffic types simultaneously. Arguably, no single routing protocol can completely cater to a deployment's various flavors. In this paper, a dynamic routing framework is proposed that can replace the traditional routing layer with a collection of routing decisions. Application packets carry a two-bit preamble that uniquely describes the nature of communication sought for. The framework dynamically wires the appropriate routing component from a set of well-defined suite.

15. SUBJECT TERMS

Wireless Sensor Networks, Routing,

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON** Kevin A. Kwiat
a. REPORT U	**b. ABSTRACT** U	**c. THIS PAGE** U	UU	5	**19b. TELEPHONE NUMBER (*Include area code)*** N/A

An Application Specific Routing framework for Wireless Sensor Networks

Mukundan Venkataraman*, Mainak Chatterjee* and Kevin Kwiat†
* School of Electrical Engineering & Computer Science, University of Central Florida,
Orlando, FL 32826, USA
† Air Force Research Laboratory, Information Directorate,
Rome, NY, USA

Abstract—**Numerous routing protocols have been proposed for wireless sensor networks, each of which is highly optimized for a certain class of traffic, like real time, reliable sense and disseminate, network reprogramming, energy efficiency and so on. However, a typical deployment demands an arbitrary communication pattern that generates multiple traffic types simultaneously. Arguably, no single routing protocol can completely cater to a deployment's various flavors. In this paper, we propose a dynamic routing framework that can replace the traditional routing layer with a collection of routing decisions. We allow application packets to carry a two-bit preamble that uniquely describes the nature of communication sought for. The framework dynamically wires the appropriate routing component from a set of well-defined suite. We conduct extensive simulation experiments that generates a concurrent mix of different traffic types – each having its own, and often conflicting, communication demands. For such an application, we show that we could meet each traffic types demands for reliability, delay, path distribution, link losses and congestion losses. We also show that service differentiation can indeed be met successfully, and practical deployments can be an imminent reality.**

I. Introduction

Various routing protocols for wireless sensor networks have been proposed: protocols for reliable routing [1], [3], [8], [9], real time communication [4], [5], [10], energy aware communication [11], [12], load balanced communication, aggregation centric approaches [6] and so on to name a few. Each such protocol typically optimizes a certain set of chosen parameters, and is claimed to work well for a particular type of network traffic. This means that a deployment that adopts any given protocol has to build its entire deployment logic using the traffic type for which the protocol is optimized. This limits the possibilities of designing exciting applications built on top of arbitrary communication patterns. Emerging classes of applications mostly demand a concurrent mix of various traffic types to make the deployment meaningful. For example, a simple application such as habitat monitoring would mostly demand all of the following activities: periodic network reports using reliable sense and disseminate, critical real time alerts when anomaly is detected, aggregation to suppress duplicates, network reprogramming, and best effort communication to transfer redundant information. Naturally, no *single* routing protocol can sufficiently handle all of the above modes of communication. One way to overcome this is to replace the conventional routing layer with a host of possible routing protocols designed for various traffic types. These protocols need to be dynamically wired based on specific application requirements. We let the application describe its nature of communication to the layers beneath, while the routing substrate dynamically picks the most appropriate routing protocol for it.

Pressed by scarcity of energy and a need to focus on performance, protocols have been developed with little thoughts to modularity and interoperability. In general, and as Culler *et.. at.* note [2], a framework for testing, integrating and proposing protocols is largely missing. Protocols do not see a common framework where they can both fit in and be evaluated for direct comparison.

In this paper, we present a unified routing framework that is easy to deploy and configure. The framework allows applications to specify, in just two bits, the *intention* of the packet in question. The intention of the packet reflects communication demands in terms of loss-intolerace and delay sensitivity of data. Our routing framwork replaces the conventional routing "layer" with a host of protocol decisions for various types of traffic. With application data publishing its intent, the routing substrate dynamically rewires itself to select the best component protocol to match a packets need. This accommodates varied and conflicting requirements gracefully. Our framework is generic enough to seamlessly integrate the other routing protocols that have been designed for various traffic types. We previously exposed an outline of a dynamic framework [7], but an evaluation of such a framework is unknown. This work provides a brief proof of concept for a dynamic routing layer.

II. Protocol Description

We add a two bit preamble to the application payload, and the bits of this preamble are set using two API calls by the application programmer. These bits describe application requirements along two lines: delay sensitivity of the payload (real-time/elastic) and criticality of payload (reliable/unreliable transmission mode).

As each packet enters the system, the framework examines the payload preamble (Fig. 1). By a simple lookup of packet requirements and available protocols, the framework "wires" a routing decision by selecting appropriate protocols (i.e., R1, R2, R3, and R4). The component protocols in the framework share the nodes resource manager and a universal neighbor lookup table. With necessary fields set (like next hop, transmission cycle etc.), the packet is passed onto the scheduler for transmission.

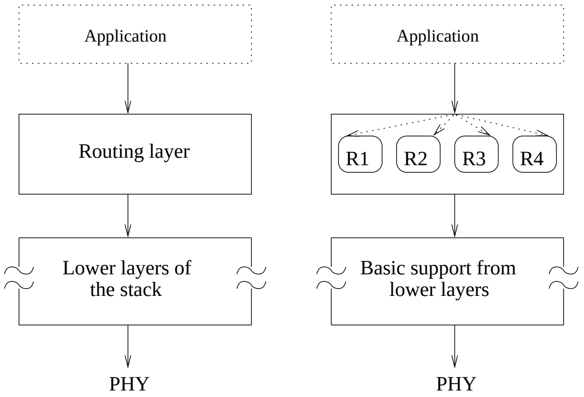

Fig. 1. Overview of the communicating framework. Preamble is decoded to dynamically decide the most appropriate routing component for a packet.

Real-Time(1)/ Non-Real-Time(0)	Reliable(1)/ Un-Reliable(0)	Inference
0	0	Unreliable, non-Real Time Packet
0	1	Reliable, non Real Time Packet
1	0	Time Critical Packet
1	1	Critical Packet

Fig. 2. Combinations of the preamble bits and their inference

A. Preamble: A "ticket" to the system

Most of the approaches to differentiate sensor data are mostly based on priority alone. We argue that application data need not to be differentiated on relative importance, but rather, the differentiating metric should be reflective of the communication requirements. This can enable protocols at various layers to optimize their performance to best meet application demands. Our two-bit scheme, taken in combination, characterizes loss and delay intolerance of application data.

Upon an inspection of the preamble bits, many inferences on the nature and demands of an application can be drawn. A tabular column listing a set of inferences are as shown in Fig. 2. For example, periodic beacons would be of type [0,0], indicating information that is neither time critical nor demanding reliability. Likewise, real time data [1,0] might be offered a shorter, less congested path bearing time criticality in mind. Reliable data [0,1], on the other hand, could be offered paths with high throughput to prevent retransmission, or longer paths that lead to balanced load. Reliable traffic could additionally be tagged for aggregation. An anomalous case of [1,1], where a packet demands both reliability and time critical delivery, is interpreted as an urgent packet that needs to make it to the destination and in a short time duration. In general, the bits serve a higher purpose: With data becoming self identifying, application programming is agnostic to the lower layers of the stack. Since the preambles are not protocol dependent, the scheme is guaranteed to work even when the mapping between the preamble and a particular protocol change over time.

B. Exchanging state information

Since the routing layer is now a collection of possible routing decisions, each such component protocol would require a different set of network state information. We build a shared neighbor table, and we populate this table with a host of information about each neighbor. A global shared neighbor table across protocols solves two problems: (i) gracefully handles conflicting data structure requirements of various protocols, which eliminate interface assumptions; and (ii) makes efficient use of limited residential memory. Specifically, we seek to house the following attributes: (i) One bit for remaining energy, set to high if more than 75% of battery life is available; (ii) One bit congestion indicator, set to high if a more than 75% of a nodes transmission queue is full; (iii) Expected transmission count, which is a ratio of number of beacons received from a neighbor to the number that should have been heard from that neighbor. This gives a good indication of link quality to that neighbor; and, (iv) Shortest path to base station is a field that advertises the minimum number of hops from that node to the base station.

At the start of a network, a node enters into its routing table all beacons that it receives for a maximum of 40 entries. With every passing beacon interval, the node calculates the number of missed beacons, and also continues updating other parameters like congestion, energy, shortest path to sink and ETX to that node. An entry is evicted if a node has considered a neighbor for 100 beacon cycles and finds that more than 70 expected beacons are missed (i.e., the estimated link quality to this neighbor is less than 0.3). Evicted nodes are entered into a blacklisted pool, where beacons from such nodes are not considered as potential neighbor table entries for the next 500 beacon intervals. This prevents stale entries reappearing into the table to be considered for a further 100 beacon cycles, and provides an opportunity for other potential candidates in the nodes neighborhood. A node keeps a good blend of potential neighbors in its table. In effect, it consists of a mix of neighbors with shorter paths to sink but weak links, nodes with very strong links but higher hops to base station, and in general, nodes with varying levels of energy and congestion. This form of routing also ensures an even distribution of traffic to various kinds of neighbors instead of a few select neighbors based on one particular routing cost metric.

C. Decoding the preamble ticket

The bits in the ticket trigger protocol actions on application data. Refer to the table (Fig. 2) for all possible combinations of the preamble bits, which are discussed further below:
Preamble bits (0,0): A packet that is not real time, and does not demand reliability. Such packets could include status updates to specific destinations, or periodic beacons that are simply broadcasted to all neighbors. The routing substrate does little to this packet and forwards it to a particular host or broadcasts it. The network tries nothing to correct a lost packet and applies a best effort model to deliver the packet. We refer to this type of traffic as **Type 1**.
Preamble bits (0,1): A packet that demands reliability, but can tolerate delay. This means that a node would retransmit the

Traffic Type	Routing Optimization
Type 1	Anycast packet, avoiding nodes with congestion and low energy
Type 2	High throughput paths with strong links, with a minimum link quality of 0.75
Type 3	Shortest path to sink
Type 4	Shortest paths with link quality 0.3 or higher, and three copies per transmission

Fig. 3. Routing optimizations used for various traffic types

packet if it is lost in transit. Since retransmission are expensive in terms of bandwidth and energy, the routing substrate identifies neighbors with very strong links, and preferably no congestion. We refer to this type of traffic as **Type 2**.

Preamble bits (1,0): A time critical packet, that demands speedy delivery, but can tolerate loss. Such packets simply demand short delivery times. Since they are agnostic to reliability, they exhibit some redundancy since multiple nodes would be reporting a similar phenomena. The routing protocol hence tries to ensure minimum delay in transit as it chooses neighbors with the *shortest path* to the base station. Since loss is tolerable, the routing protocol does not differentiate potential neighbors based on link quality. We refer to this type of traffic as **Type 3**.

Preamble bits (1,1): This type of packet is contradictory, since it demands both reliability as well as speedy delivery. Hence, the routing module selects shortest paths but sends *multiple* copies to thwart link losses. In our implementation, three copies of the same packet are transmitted. This type of traffic would be generated in the event of a critical alert, where a piece of information has to both make it to the destination, and in as short a time as possible. We refer to this type of traffic as **Type 4**.

III. Simulation Results

To better understand the performance of real world applications, we generated a synthetic traffic pattern of a typical deployment. Nodes in the deployment generate data traffic based on a Poisson process (on a 10% activity interval), and generate control traffic in a deterministic fashion (at an interval of 50 secs with random seeds). All traffic are destined for the sink. The traffic types are evenly mapped onto one of four classes that we discussed.

A. Profiling delivery ratio: attributing causes of packet loss

Delivery ratio is defined as the fraction of packets received successfully at the sink to the number generated at the source. The plot for this is shown in Fig. 4. Delivery ratio decreases with rising number of nodes. Because of the nature of neighbor selection, delivery ratio is much higher for reliable traffic (Type 2), and is surprisingly comparable for critical traffic which takes shortest paths. Delivery ratio drops significantly for the other traffic types (Type 1 and 3) with rising number of nodes. This overall trend indicates that application logic can be dynamically met.

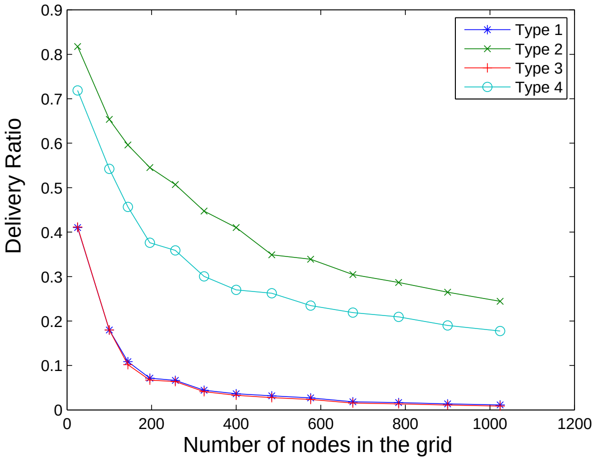

Fig. 4. Delivery ratio

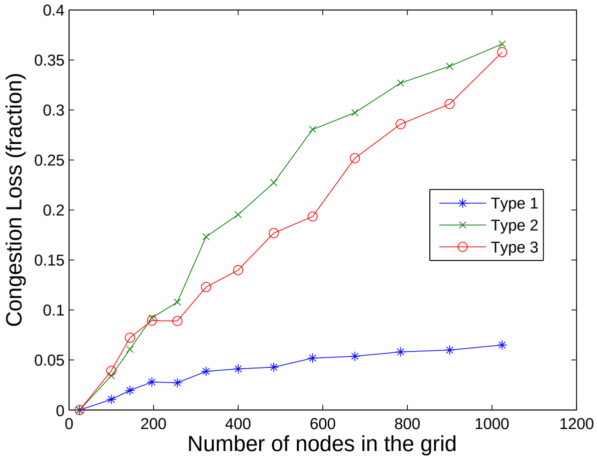

Fig. 5. Congestion loss experienced by various traffic types

Congestion experienced by the various traffic types are also interesting (Fig. 5). Reliable traffic (Type 2) experiences maximum congestion losses of all traffic types. This is largely because of the numerous short hops the packets take, which maximize chances of encountering congested nodes, especially closer to the sink. Type 3 traffic, which simply picks shortest hops, also experiences significant congestion losses.

B. Delay experienced by traffic types

We measure end-to-end delay for the four traffic types (Fig. 6). Intuitively, delay increases with growing network size. While reliable traffic (Type 2) witnesses a big jump in delay values largely because of numerous short hops with long queue waiting times, the other types of traffic witness delays almost five magnitudes smaller. Interestingly, critical packets (Type 4) witness the *least* delay, yet have delivery ratio's comparable to reliable traffic.

C. Path distribution

Finally, we seek to characterize the path length distribution (Fig. 7) for the traffic types for a case of 1024 nodes in the

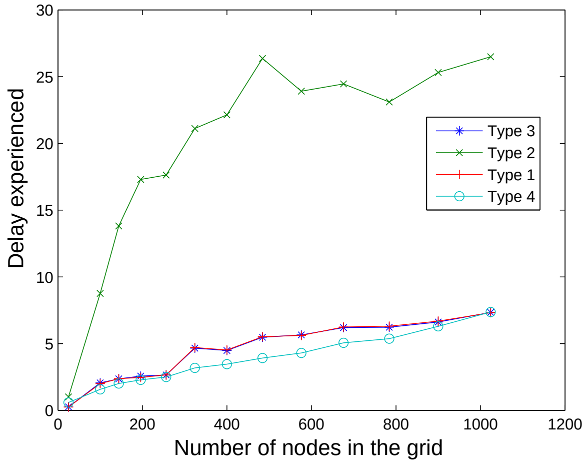

Fig. 6. End to end delay experienced by various traffic types

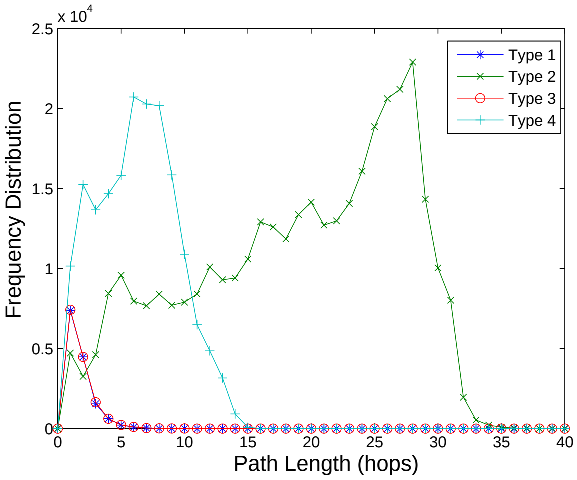

Fig. 7. Frequency distribution of path length taken by various traffic types

network. We measured the frequency of selecting a path with a given hop count. Real time traffic would mostly crowd around short path lengths, while reliable traffic would tend to pick long paths to the sink. As shown in the plot , reliable traffic crowds around 30, while real time traffic is dominant for hop length of less than 5. It is evident that reliable traffic take many hops to make it to the sink, while the other traffic types show a much smaller profile. Critical packets (Type 3) take a slightly longer route to the sink mostly because they seek short paths with link estimates of more than 0.3. This is not the case with real time traffic, which on an average takes shorter number of hops to destination

IV. Conclusions

Monolithic protocol per layer has enjoyed immense popularity in a domain like the Internet, largely because of the Internet's end-to-end philosophy coupled with the paradigm of “dumb“ network core. Differences in networking challenges between the Internet and sensor networks have been sufficiently emphasized in the past, and it is well established that end-to-end models do not work very well in sensor networks. A shift towards a smart network core that can dynamically switch its behavior and communicate wisely is the only arguable means to achieve application fidelity and conserve scarce resources.

Our results firmly establish the fact that traffic in a network can be differentiated with just two bits, and these bits provide an excellent insight into the requirements for various traffic types. With the establishment of such traffic types, it is now relatively easy to deploy complex and complete deployment logic, and achieve application fidelity not seen before. As a basis, we believe this is an excellent starting point to make deployments practical and meet the stated goals.

On the other hand, we have now exposed a framework into which protocols can be easily developed and integrated. The vast breadth of contributions already made in sensor networks find their right place and recognition in the stack. The fact that a given protocol can only cater to a particular traffic type is now utilized by mapping that sort of traffic to the protocol. We have also shown the co-existence of multiple routing logic, and their success in carrying out a deployment logic. We conclude this paper with the hope that the framework promotes synergy, helps meet deployment logic, and becomes an active platform for both contributing and testing protocols for sensor networks.

References

[1] Q. Cao, T. Abdelzaher, T. He, and R. Kravets, “Cluster-Based Forwarding for Reliable End-to-End Delivery in Wireless Sensor Networks”, *IEEE Infocom'07*, May 2007.

[2] D. Culler, P. Dutta, C. T. Ee, R. Fonseca, J. Hui, P. Levis, J. Polastre, S. Shenker, I. Stoica, G. Tolle, J. Zhao, “Towards a sensor network architecture: Lowering the waistline”, *HotOS X'05*, June 2005.

[3] D. S. J. De Couto, D. Aguayo, J. Bicket, and R. Morris, “A High-Throughput Path Metric for Multi-Hop Wireless Routing”, *ACM Mobicom'03*, Sept 2003.

[4] T. He, J.A. Stankovic, C. Lu and T. Abdelzaher, “SPEED: A Stateless Protocol for Real-Time Communication in Sensor Networks”, *Proc. ICDCS'03*, May 2003.

[5] C. Intanagonwiwat, R. Govindan, and D. Estrin, “Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks”, *ACM/IEEE Mobicom'00*, Aug 2000.

[6] S. Pattem, B. Krishnamachari, and R. Govindan, “The Impact of Spatial correlation on Routing with Compression in Wireless Sensor Networks”, *ACM/IEEE IPSN'04*, April 2004.

[7] M. Venkataraman, K. Muralidharan and P. Gupta, “Designing New Architectures and Protocols for Wireless Sensor Networks: A Perspective”, *IEEE Secon'05*, Sept 2005.

[8] A. Woo, T. Tong, and D. Culler, “Taming the Underlying Challenges of Reliable Multihop Routing in Sensor Networks”, *ACM Sensys'03*, Nov 2003.

[9] C. Y. Wan, A. Campbell, L. Krishnamurthy, “Pump-Slowly, Fetch-Quickly (PSFQ): A Reliable Transport Protocol for Sensor Networks”, *IEEE Journal on Selected Areas in Communications (JSAC)*, Vol. 23(4), April 2005.

[10] F. Ye, A. Chen, S. Liu, L. Zhang, “ A scalable solution to minimum cost forwarding in large sensor networks”, *Proc. of the 10th International Conference on Computer Communications and Networks (ICCCN)*, AZ, Oct 2001.

[11] M. A. Youssef, M. F. Younis, K. Arisha, “A constrained shortest-path energy-aware routing algorithm for wireless sensor networks”, *Proc. of IEEE WCNC*, Orlando, FL, March 2002,

[12] W. R. Heinzelman, A. Chandrakasan, H. Balakrishnan, “Energy-efficient communication protocol for wireless microsensor networks”, *Proc. of 33 Hawaii International Conference on Systems Science (HICSS)*, Hawaii, Jan 2000.