

# M<sup>2</sup>RC: Multiplicative-increase/additive-decrease Multipath Routing Control for Wireless Sensor Networks

HANY MORCOS

IBRAHIM MATTA

AZER BESTAVROS

{hmorcos, matta, best}@cs.bu.edu

Computer Science Department, Boston University

**Motivation:** A routing protocol for wireless sensor networks is expected to operate in a challenged environment. Powered by batteries, sensor nodes have to conserve as much energy as possible to increase the lifetime of the sensor network. Furthermore, node failures and packet losses are expected to be common in many sensor networks, *e.g.* military scenarios. These failures/losses could be *transient* in nature, for example due to temporary wireless interference. Thus, a routing protocol for such challenged networks should be capable of adapting to constantly changing network conditions while saving as much power as possible.

GRAB [1] is a routing protocol recently proposed for sensor networks to enhance the reliability of data delivery. Unlike previous approaches, *e.g.* Directed Diffusion [2], which are adaptive *single-path* routing protocols, GRAB is *multi-path* subject to some energy budget constraint on every delivered packet.

Under Directed Diffusion, from the multiple paths available to the sink, the source or an intermediate forwarding node selects *one* path as the primary path based on the quality (*e.g.* loss, delay) of data delivery—data is delivered on secondary paths at a lower rate to keep them alive and determine their quality. Thus, based on local changes in network conditions, a node may switch its primary path to another one. This local reaction to local changes is however limited by the inherent delay of such single-path routing protocol to switch to and establish a new loop-free primary path in case of local error conditions.

To be more proactive, GRAB is multi-path—a sensor node forwards a data packet (report) to several neighbors as long as the total power consumed by that packet is still within a certain given budget. The number of neighbors to which an under-budget packet is forwarded is *static* (*e.g.* three neighbors), regardless of the level of local errors. The only adaptation of GRAB is end-to-end—for example, if the average success rate of data reports falls below a certain threshold, the sink resends an advertisement (ADV) packet to re-establish the routing state in the *whole* network, thus unreliable/failed nodes/links may be avoided. This end-to-end approach to local error conditions is problematic since it is very costly especially if errors are transient. It may be more energy-efficient to react to such temporary error conditions in a localized way.

**Our Contribution:** We propose a new approach to routing in wireless sensor networks. Our approach is inspired by control-theoretic adaptations similar to those widely used

in the Internet, *e.g.* additive-increase multiplicative-decrease (AIMD) of TCP for reacting to congestion conditions. Our protocol, we call M<sup>2</sup>RC, uses a multiplicative-increase additive-decrease (MIAD) controller to adapt the number of neighbors to which a packet is forwarded. This current number of neighbors is maintained at each node in a control variable, we call *branching factor* (*bfac*). *bfac* is controlled based on locally observed error (packet loss) conditions.

In brief, *bfac* is multiplied by two if the last forwarded packet is not correctly received by a neighbor leading to the sink; *bfac* is decremented by one upon successful packet delivery. By aggressively increasing *bfac* upon local errors, M<sup>2</sup>RC expands its routing reach to more neighbors that are further away to quickly overcome *local* unreliability, at the expense of consuming more power. Whenever local error conditions subside, M<sup>2</sup>RC shrinks its routing reach conservatively, finally forwarding data packets (reports) to a single neighbor that is closest in distance to conserve energy.

To summarize, M<sup>2</sup>RC has the following salient features:

- Unlike existing routing schemes, M<sup>2</sup>RC is neither single-path nor multi-path, rather each node adapts the multiplicity of paths based on its observed (local) error/loss conditions;
- Under M<sup>2</sup>RC, only for high error levels, a node uses high-powered transmissions to reach farther neighbors. For low error levels, a node adapts to low-powered transmissions. Thus, M<sup>2</sup>RC judiciously consumes power based on local error conditions, which maximizes the lifetime of the network and minimizes the cost of the power consumed per successfully delivered data report; and
- M<sup>2</sup>RC is unique in its use of an adaptive MIAD controller, which aggressively probes for possible routes to deliver data packets, thus reacting quickly within unreliable areas of the sensor field.

We implemented M<sup>2</sup>RC in the ns-2 network simulator. We compared it against GRAB; *min-hop routing*, where each node sends to the farthest neighbor toward the sink using maximum power; and *min-power routing*, where each node forwards data packets along the least-energy path toward the sink. Our simulation results confirm the premise of M<sup>2</sup>RC, which is found to outperform all other schemes—M<sup>2</sup>RC achieves over 90% successful delivery rate using around 30% less power than that of GRAB.

**M<sup>2</sup>RC Details:** M<sup>2</sup>RC has two phases: the first phase is the mesh establishment phase and the second one is the data for-

warding phase. During the *mesh establishment* phase, the sink broadcasts an advertisement (ADV) packet. This ADV control packet serves two purposes: (1) it assigns a cost to each forwarding node, where a node’s cost is the minimum amount of power needed to reach the sink from this node. Thus the established costs increase as we go further from the sink; (2) the ADV packet gives each node a chance to discover its neighbors toward the sink. Each node  $N$  maintains a list of neighbors whose cost is less than that of  $N$ , *i.e.*, closer to the sink than  $N$ . To ensure loop-freedom,  $N$  can forward packets only to any of those neighbors to reach the sink.  $N$  also maintains its list of neighbors ordered based on the physical distance between itself and each neighbor— $N$  uses these distances to calculate the amount of power needed to broadcast/forward data packets.<sup>1</sup> This mesh establishment phase is similar to that of GRAB.

During the second phase of *data forwarding*, the source gathers sensing reports and forwards them to the sink. The source attaches to each packet a certain amount of “budget.” This budget represents the total amount of power that could be expended by this packet along the way to the sink. A node drops a data packet if the packet’s budget has been depleted. When forwarding a data report, a node includes its cost in the header of the packet. Upon receiving a data packet from node  $Y$ , node  $X$  can only rebroadcast this packet if its own cost is less than the cost of  $Y$ —this ensures that M<sup>2</sup>RC is free of routing loops.

Each node maintains a control variable called the *branching factor* (henceforth denoted by *bfac*), which indicates the current number of neighbors (from closest to farthest in distance) to which this node should broadcast data packets. The higher *bfac* is, the more power the node uses to send packets and thus the more reliable the communication is. When node  $X$  needs to broadcast a data packet,  $X$  looks up its neighbor list for the distance between itself and its neighbor numbered *bfac*.  $X$  then calculates the amount of power needed to send the packet to that neighbor. We next explain how *bfac* is adapted using an MIAD controller.

Every node initializes *bfac* to one. This means that a node initially broadcasts data packets only to its closest neighbor, thus requiring the least power. After sending a data packet, node  $X$  waits for an acknowledgment (ACK). If  $X$  hears an ACK for the data packet within a predefined timeout interval,  $X$  doesn’t do anything else concerning this packet. Otherwise,  $X$  doubles the current value of *bfac* and retransmits at a higher power to reach its neighbor numbered *bfac*. This process continues until either  $X$  hears an ACK for the packet or the value of *bfac* reaches its maximum, which is determined by the total number of neighbors that  $X$  learned about during the mesh establishment phase. Upon receiving an ACK,  $X$  starts to decrease its *bfac* additively (once every a certain number of acknowledged data packets) to a minimum

<sup>1</sup>We assume power is related to distance according to a relationship of the form  $\text{power} \propto \text{distance}^\beta$ , where  $\beta > 1$ .

value of one.

In sending an ACK, node  $Y$ , upon receiving (hearing) a data packet from node  $X$ , estimates its distance from  $X$  based on the strength of the received signal. If the distance over which  $Y$  will next forward that packet is at least equal to its distance from  $X$ , then  $X$  implicitly receives an ACK by overhearing  $Y$ ’s transmission. Otherwise,  $Y$  explicitly sends a small ACK packet to  $X$ .

**Performance Evaluation:** In our ns-2 simulations, we did not include Directed Diffusion as it was shown to be inferior to GRAB [1]. We evaluated GRAB, *min-hop routing* and *min-power routing*, over a sensor field of size  $3200 \times 800$  meters with 256 nodes located uniformly in the field. We simulated both node failures and packet losses. All nodes experience a packet loss probability of 0.1. In our node failure model, in a specified subarea(s) of the sensor field, each node stays up for an exponentially distributed time with an average of 300 seconds, then the node goes to an exponentially distributed (temporary) blackout period with an average that ranges from 60 to 210 seconds.

We vary the ratio of blackout-to-uptime to study the performance of the various routing protocols. Our main performance measures are: *throughput*, defined as the percentage of successfully delivered data reports; and *delivery cost*, defined as the power consumed per successfully delivered data report.

Our results show that M<sup>2</sup>RC outperforms all other protocols. The *min-power routing* scheme fails to successfully deliver any data reports due to its long paths. The *min-hop routing* scheme is inferior to both M<sup>2</sup>RC and GRAB due to high collisions resulting from its high-powered transmissions. Finally, M<sup>2</sup>RC performs better than GRAB—this superiority is more pronounced when the unreliable subarea is located in the middle of the sensor field. In this scenario, because of its MIAD-based routing control, M<sup>2</sup>RC behaves as min-power (single-path) routing outside the unreliable subarea, whereas it behaves in an *adaptive* multi-path routing fashion inside the challenged subarea. M<sup>2</sup>RC achieves higher throughput than GRAB with up to 20% less power consumption per successfully delivered report.

**Future Work:** We are currently investigating the performance of M<sup>2</sup>RC over a wider range of conditions. We are also developing mathematical models that capture the dynamics of routing in sensor networks, and of M<sup>2</sup>RC in particular.

## References:

- [1] F. Ye, G. Zhong, S. Lu, and L. Zhang. GRADient Broadcast: A Robust Data Delivery Protocol for Large Scale Sensor Networks. To appear in *ACM Wireless Networks (WINET)*, Vol. 11, No. 2, March 2005.
- [2] C. Intanagonwiwat, R. Govindan, and D. Estrin. Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks. In *Proc. of MobiCOM*, August 2000.



# M<sup>2</sup>RC: Multiplicative-increase/additive-decrease Multipath Routing Control for Sensor Networks



Hany Morcos, Ibrahim Muta, Azet Bestavros

## Routing in Sensor Networks

- Hard task !!! WHY ??
  - Sensors are battery powered → limited power supply
  - Harsh environments: packet losses and node failures
  - Fast changing routing conditions
- ### Tasks of A Routing Protocol
- Save as much energy as possible
  - Counteract the effect of the environment
  - Adapt to current routing conditions

## M<sup>2</sup>RC

### Mesh Establishment Phase

- Sink sends ADV packet, which serves two goals:
  - Establish cost field in forwarding nodes;
  - node's cost is the minimum amount of energy to reach the sink from this node
- Using ADV packet, nodes discover neighbors (only towards the sink). For any node  $N$ , its list of neighbors is ordered based on increasing distance from  $N$

### Data Forwarding Phase

- Source sends data packets with limited power budget
- Each node  $N$  keeps a variable "branching factor", which is the target number of neighbors to get this packet
- Using its branching factor, any node  $N$  determines the distance of the farthest neighbor to which packet is sent
- Starting with a value of 1 for the branching factor variable,  $N$  sends the data packet to its least-cost (energy) neighbor. Then wait for an ACK from any neighbor with less cost
- Control the branching factor variable using MIAD:
  - When no ACK, multiply by 2; else decrement by one

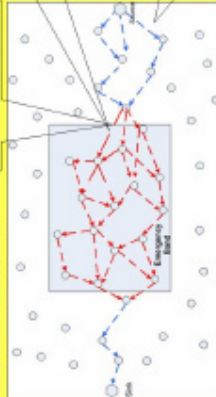
## Why is M<sup>2</sup>RC different ?

- Uses more power to send packets in problematic areas only. Otherwise, it just uses minimum power to save battery life
- Handles local changes of routing conditions locally
- Adaptive to changes in routing conditions using an MIAD controller
- Outperforms min-hop (high-powered) routing, min-power routing and static mesh routing (GRAB of UCLA), in terms of throughput and energy consumed per delivered packet (report) to the sink

## Evaluation

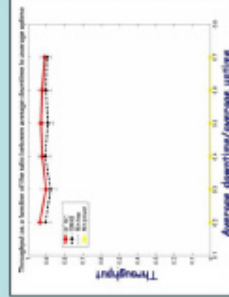
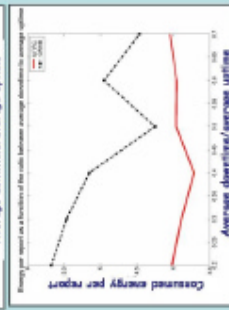
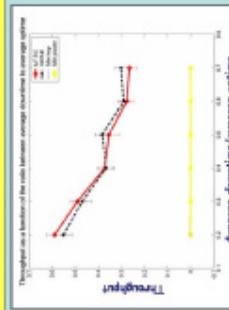


### Network Model



high-powered transmissions in less-reliable areas  
minimum power transmissions in reliable areas

## Results



Unreliability over the whole sensor field

Unreliability in the middle of the sensor field