MicroRouting: A Scalable and Robust Communication Paradigm for Sparse Ad Hoc Networks

Saumitra M. Das Himabindu Pucha Y. Charlie Hu
School of Electrical and Computer Engineering
Center for Wireless Systems and Applications
Purdue University
West Lafayette, IN 47907
{smdas, hpucha, ychu}@purdue.edu

***Abstract*— Mobile ad hoc networks (MANETs) are self-organizing networks that provide rapid network connectivity in infrastructureless environments. Most routing protocols designed for MANETs assume connected networks. Such a restriction directly limits the application domains of MANETs. In this paper, we study the problem of providing time-critical data delivery in sparse ad hoc networks where network partitions can last for long periods, without imposing any restrictions on the node mobility. Supporting real-time communication with unconstrained mobility is important to many mission-critical applications such as battlefields and search and rescue in large-scale disaster areas.**

In this paper, we propose microrouting networks consisting of tiny nodes similar to sensors but without transducers (called microrouters) as a substrate for time-critical data delivery in sparse MANETs. We describe the microrouting protocol for the resulting hybrid network which exploits the fact that microrouters are stationary, but are constrained by energy and memory. We demonstrate the viability of the microrouting network architecture via detailed simulation evaluation. Our results show that microrouting networks running the microrouting protocol efficiently extend the connectivity of sparse MANETs and provide high packet delivery ratios.

I. Introduction

A mobile ad hoc network (MANET) consists of a collection of wireless mobile nodes dynamically forming a temporary network without relying on any existing network infrastructure or centralized administration. A fundamental challenge in MANETs is the design of scalable and robust routing protocols that can provide any-to-any communication among its participants. Most routing protocols designed for MANETs assume connected networks, i.e., a multi-hop path exists between any two nodes. Such an assumption restricts the geographic area of operation to be a function of the number of nodes and their radio transmission range, and thus directly limits the application domains of MANETs.

In this paper, we study the problem of supporting real-time data delivery in sparse MANETs in which network partitions happen frequently and can last for a long period, without imposing any restrictions on the mobility of the participating nodes. One can envision many practical application scenarios in which a small number of mobile nodes (participants) are deployed in large geographic regions. Such sparse deployment scenarios can arise due to budgetary limitations, the nature of the applications, or physical constraints of the environment. For example, in a battlefield, several units of soldiers can be dispersed in a large combat zone. As another example, in a large disaster area, a large number of small teams of personnel can be roaming around performing search and rescue operations. Any mechanism supporting data delivery in such applications needs to satisfy the following three criteria:

- Frequent network partitioning: The mechanism needs to be able to operate under frequent network partitioning, since the area covered by the mobile nodes may be large and any fixed infrastructure is not available or may have been damaged.
- Real-time communication: The mechanism needs to support real-time communication as the tasks being carried out such as disaster relief are often time-critical.
- Unconstrained mobility: The mechanism should not impose any controlled mobility as certain imposed movement of nodes may not be feasible due to the inhospitable terrain, enemy fire, obstacles due to a disaster, etc. Additionally, in such critical application scenarios, the tasks to be accomplished can be fundamentally more important than ensuring connectivity, i.e., the mobile nodes may not be able to simultaneously perform their duties efficiently and adjust their movements to ensure connectivity. For example, in disaster relief, it is more important to search a large area quickly than to stay connected but search a small area.

In summary, many critical applications of MANETs like disaster relief and military operations are characterized by sparse deployment, real-time communication, and unconstrained mobility which require a new communication paradigm that is flexible, efficient, and easily deployable.

There have been several recent work on supporting data delivery in sparse or disconnected networks via exploiting the mobility of nodes in MANETs [11], [18], [5], [6], [22]. Such mobility-based protocols either exploit the existing mobility of the nodes to buffer and deliver messages across network partitions [18], [5], or require nodes to move in a controlled manner to ensure network connectivity [11], [6], [22]. Relying on existing node mobility for message delivery can suffer from high delays and potentially unreliable data delivery. Altering the movement of mobile nodes is limited to application scenarios where such altered movement is feasible and does not interfere with the tasks being performed. Therefore, these mobility-based approaches are unsuitable for the class of mission-critical applications that we envision.

In this paper, we propose to use *microrouting networks* con-

In *Proceedings of the 5th IEEE International Workshop on Algorithms for Wireless, Mobile, Ad Hoc and Sensor Networks (WMAN), Denver, CO, Apr 4-8, 2005.*

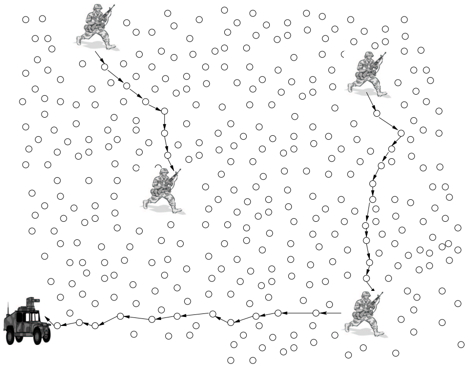


Fig. 1. A micr::routing network. Soldiers and military vehicles use the routing substrate to communicate.

sisting of tiny nodes similar to sensors but without transducers (called *microrouters*) as a substrate for data delivery in sparse MANETs. Like sensor nodes, such microrouters can be easily and quickly deployed to cover a geographic region in which a small number of mobile wireless nodes will be sparsely deployed to perform short-term or time-critical missions. For example, in Figure 1 a military unit consisting of soldiers and vehicles deployed in a large area uses the underlying microrouting network to communicate and coordinate their activities. Such a substrate is easily deployable in inhospitable terrains or disaster stricken areas. At the same time, it decouples the mobility of these nodes (soldiers, rescue personnel, military vehicles, rescue robots) from the task of maintaining connectivity amongst them and thus the mobile nodes can optimize their mobility solely based on the need of the mission. Our study demonstrates that this substrate provides high data delivery rates with low overhead and delays for such sparse MANETs. We believe many mission-critical applications can greatly benefit from this new communication paradigm.

A primary challenge in using microrouting networks is to design a multi-hop routing protocol that provides efficient and reliable data delivery to its mobile participants. The key characteristics of such a protocol are stateless operation, energy efficiency, and robustness. The protocol ideally should be stateless and robust since the microrouters are likely to be intermittently available due to frequent wake/sleep cycles for energy savings and due to external interference in hostile environments. Additionally, microrouters are energy- and memory-constrained devices which can only support minimal routing protocol capabilities.

The rest of the paper is organized as follows. Section II describes the characteristics of microrouting networks and compares it with traditional ad hoc networks and sensor networks. Section III provides a detailed description of the microrouting protocol for microrouting networks. Section IV presents simulation results. Section V discusses previous work on data delivery in sparse MANETs and finally, Section VI concludes the paper.

II. Microrouting Networks

In this section, we discuss the architecture of microrouters, the characteristics of the corresponding microrouting network, and the design principles of the microrouting protocol.

A. Microrouter Architecture

The microrouting networks we propose in this paper consist of microrouters that are similar to sensor nodes in sensor networks except they do not contain any transducers for performing sensor operations in order to reduce costs and form factor. To be easily deployable, these microrouters need to be low cost and dispensable, and have a small form factor. Each microrouter would typically be made up of a processing unit, a transceiver unit, and a power unit.

The processing unit is associated with a small memory unit and implements the microrouting protocol (explained in Section III). Like in sensor nodes, the memory unit available to the processing unit is a scarce resource. For example, the smart dust prototype has 512 bytes RAM and 512 bytes EEPROM [10] while the MICA2Dot [3] nodes have 128 Kbytes of program flash memory and 4 Kbytes EEPROM. The transceiver unit performs wireless communication. Typically the radio design of a microrouter can be similar to that used in current MICA motes. The power unit can be a one-time energy source such as a battery when a microrouting network is deployed for short-term missions. It can also be a power scavenging unit in medium-term to long-term deployment scenarios. In either case, microrouters need to be energy-efficient to extend the lifetime of the network.

We assume that the microrouters are not equipped with location finding devices such as GPS for the following reasons as argued in [17]: (1) GPS devices have a high production cost, especially when a large number of microrouters are to be equipped; (2) GPS devices may not be able to function in environments such as indoor locations or locations with dense foliage or obstacles that block the line of sight from GPS satellites; (3) the power consumption of GPS devices would reduce the battery life of microrouters; and (4) a GPS and its antenna would increase the form factor of a microrouter. This inhibits the easy deployment of microrouters. Note that assuming no position information reduces the applicability of sensor routing protocols that can deliver data to mobile sinks [19] in microrouting networks.

In summary, microrouters are memory-constrained, energy-constrained, low cost devices. Since the goal of a microrouting network formed by stationary microrouters is to provide data delivery to mobile nodes in a sparse MANET, the multi-hop routing protocol needs to be supported on both the microrouters and the mobile nodes in the resulting hybrid network. For simplicity, in the rest of the paper, we will refer to the hybrid network as the microrouting network.

B. Characteristics of Microrouting Networks

We discuss the unique characteristics of microrouting networks by comparing them to traditional mobile ad hoc networks and sensor networks. MANETs are envisioned to have typically few hundreds of nodes, whereas sensor networks are much larger in scale. A microrouting network has a large number of microrouters similar to a sensor network to ensure coverage and connectivity. The typical radio range in a MANET is higher than most sensor nodes. In contrast, a microrouting network may have heterogeneous ranges, small for the microrouters and larger for the mobile nodes. Also, the node density of a MANET is generally lower than that of a sensor network. The node density of a microrouting network is similar to that of a sensor network. In a MANET, all the nodes are generally mobile and thus the topology is highly dynamic. A sensor network has typically static sensor nodes. However, a change in topology can still occur due to failure of a sensor or due to sensors that have run out of energy. Thus, sensor networks have a slowly changing topology. In contrast, a microrouting network has nodes that are constantly mobile and microrouters that are static and susceptible to failures and energy constraints similar to in a sensor network. In MANETs and microrouting networks, any mobile node can communicate with any other mobile node. However in a sensor network, the sensors typically communicate only with the sink, resulting in a many-to-one communication pattern. Finally, in a MANET, globally unique identifiers (IDs) are required to identify and communicate with all the nodes. In a microrouting network, unique IDs are similarly required for both mobiles as well as microrouters to enable end-to-end connection establishment and maintenance. However, microrouters can be addressed separately from mobile nodes since they do not initiate or terminate a flow. Sensor nodes need not be individually addressable for typical sensing applications.

C. Microrouting Protocol Design Principles

The unique architecture of microrouters and the characteristics of microrouting networks dictate the following design principles for the multi-hop microrouting protocol.

- *Stateless Architecture*: The protocol for the microrouters should be stateless since microrouters are resource-constrained devices with limited memory and energy source. In addition, they also suffer from intermittent availability due to frequent wake-sleep cycles when using energy saving techniques. Thus, stateless techniques like source routing may be more useful in microrouting networks since they can be efficient in the presence of intermittent availability.
- *Localized Route Repair*: The protocol running on the mobile nodes should minimize the frequency and extent of route discoveries. In large scale networks, such flooding-based route discoveries lead to the broadcast storm problem [13] as well as energy drain. More importantly, any repair technique performed by the microrouters should ideally be stateless.

Previously proposed sensor routing protocols can not be readily used in microrouting networks. The primary goals of sensor routing and data dissemination protocols are data aggregation and efficient dissemination. Most protocols propose the use of data-centric routing in place of node-centric routing required in a microrouting network. Also, the majority of the protocols developed for sensors do not deal with highly dynamic topology caused by mobility of nodes, whereas microrouting needs to handle such mobility. Exceptions are dissemination protocols such as TTDD [19] which handle only sink mobility. However, TTDD assumes the presence of GPS on sensor nodes. Additionally, both endpoints of a flow in a microrouting network may not be resource-constrained and may have replenishable energy resources unlike in a sensor network. This would not be exploited by sensor routing protocols.

Similarly, off-the-shelf MANET routing protocols can not be readily used in microrouting networks. Most of the protocols proposed assume resource rich nodes all along the path which can keep a large amount of state and perform a lot of computation. Hop-by-hop routing protocols like AODV [16] and DSDV [15] need to maintain state on intermediate nodes for forwarding packets and consequently their routing table sizes grow with the network size or the number of active packet sources. For example, in AODV, each intermediate node along the route maintains a routing table for forwarding packets, a packet buffer for local repair, and backward pointers for route error propagation. In the presence of intermittent availability, such state may be lost frequently along paths causing costly rediscoveries. In DSR, each node maintains a cache of source routes that may be a graph [7] or a list of paths [9]. Intermediate nodes use the cache to salvage packets with stale routes and reply to route requests. Additionally, AODV, DSR, and TORA [14] are unsuitable for microrouting networks since they all frequently invoke global flooding of route requests to discover routes to destinations. Finally, protocols developed for MANETs fail to take advantage of the static nature of microrouters in microrouting networks.

III. Microrouting Protocol Design

This section presents the design of the microrouting protocol, μRP, for supporting data delivery between sparsely deployed mobile nodes in microrouting networks. μRP is used as a representative routing protocol to demonstrate the viability of the microrouting network architecture. Conceptually, μRP has two distinct and separate modules. The first module runs on the energy- and memory-constrained microrouters. The second module runs on the mobile nodes which have larger memory and more energy. The two modules are designed to interoperate seamlessly to provide end-to-end connectivity while exploiting the characteristics and capabilities of the devices they run on. Since μRP uses source routing, it is similar to DSR in many aspects and can be though of as an adaptation of the DSR protocol for microrouting networks.

In the description, we use N_r^A to represent the number of microrouters in a region of radius r around node A. We also denote the average speed at which a mobile node as v, and the transmission range as R.

A. Node Addressing

Each mobile node is addressed using an IP address. We assume some address assignment algorithm exists which assigns these to mobile nodes. Since the microrouters do not act as endpoints of communication, we can efficiently encode their addresses with the minimum number of required bits rather than using 32-bit IP addresses. This significantly reduces the overhead of using stateless source routing which encodes addresses in packets. Additionally we assume that the address of a device can be used to distinguish between it being a microrouter or a mobile node. This can be easily achieved, for example, using a predefined prefix for the addresses of the microrouters.

B. Transmission Range

Mobile nodes and microrouters can potentially have different transmission ranges, and this hybrid nature of microrouting networks can result in unidirectional links. The results in [12] show that using unidirectional links does not generally improve performance and should be avoided. Thus, in a microrouting network, the transmission range of the mobile nodes is adjusted to be similar to that of the microrouters. This also conserves the energy of the mobile nodes.

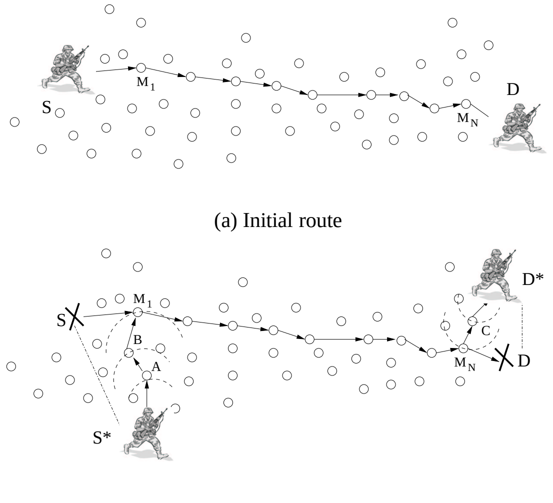
C. Route Discovery

μRP discovers an initial route between a pair of mobile source and destination nodes *reactively*. When a mobile node S needs to send packets to a destination D for which it has not discovered a route, it floods a QUERY packet into the microrouting network in a way similar to route discovery in reactive routing protocols such as DSR and AODV. Each microrouter that has not previously received the QUERY rebroadcasts it after appending its address to the accumulated path in the packet. Because of the sparsity of mobile nodes, any route between a source and a destination will likely consist of multiple microrouters as intermediate hops. To maximize the lifetime of a route, only microrouters forward QUERY packets while mobile nodes along the way ignore such packets unless they are the queried destination node. An initial route discovered in this way will consist of only stationary microrouters as intermediate hops and only the first and the last hops will be affected by the mobility of mobile nodes.

Due to memory constraint, the microrouters along the path of the reply do not cache any route information and consequently no intermediate replies are possible for route discoveries. Once the QUERY reaches D, D sends a QUERY REPLY back to S by reversing the accumulated path. When the QUERY REPLY reaches S, S adds the link information in the route to a graph cache. A *graph cache* stores individual links of paths to build a topological graph of the network. Each link is assigned a timeout value as follows: Links between microrouters are assigned a large timeout since these links are not affected by mobility. Links involving a mobile node are assigned a timeout value based on an adaptive scheme that uses past mobility history of the nodes involved in the link. This adaptive scheme is similar to the graph cache scheme proposed in [7]. The source node then runs a single source shortest path algorithm to discover a route for sending packets to D. To maximize the lifetime of a route, the shortest path algorithm is modified to ignore mobile nodes as intermediate hops in a route. Figure 2(a) shows an example of an initial route discovered that connects the source S and the destination D.

D. Local Route Repair

Since the two end nodes of a route, i.e., the source and destination nodes, are mobile, both the first hop and the last hop of the route will break as the two nodes move. Figure 2(b) shows that the source has moved from S to S^*, which is beyond the transmission range of the first microrouter M_1, breaking the first hop. Similarly, the destination node has moved from D to D^*, which is beyond the transmission range of microrouter M_n. This movement breaks the last hop of the route.



(a) Initial route

(b) Route repair

Fig. 2. μRP route discovery and repair.

μRP uses two different local repair mechanisms to fix breakage of the first hop and the last hop, instead of frequently rediscovering a new route using a network-wide flooding.

1) Source local repair: On the source node side, μRP uses a *reactive* scheme for local route repair. When the first hop breaks, the source node performs a search of its L-hop neighborhood by broadcasting a NEIGHBORHOOD DISCOVERY packet. Each microrouter that receives a NEIGHBORHOOD DISCOVERY for the first time appends itself to the header and rebroadcasts the NEIGHBORHOOD DISCOVERY. When the NEIGHBORHOOD DISCOVERY has traveled L hops, the last microrouter sends a NEIGHBOR REPLY using the encoded L-hop route in the packet back to the source node. The source adds the link information received in NEIGHBOR REPLIES into its graph cache. This L-hop flooding provides the source with a path to M_1 which can then be concatenated with the old route from $M_1 \rightarrow D$. Thus, the source repairs the route by running a single source shortest path algorithm on its graph cache. Figure 2(b) shows that the new route now goes through microrouters A and B before reaching M_1. Note that the repaired path may not go through M_1, for example, if it discovers a shorter or equally long route to some other node on the path, e.g., M_2. If the source repair does not succeed, a route discovery is initiated. The source side local repair involves an L-hop flooding of a NEIGHBORHOOD DISCOVERY packet, followed by the replies from the nodes on the perimeter of the L-hop flooding zone. The total packets transmitted is thus $2N_L^S$ (recall that N_r^A is defined as the number of microrouters in a region of radius r around node A). Assume that the average interval for the breakage of the first link $\frac{1}{\lambda_1}$. This is proportional to $\frac{R}{v}$. Since the packet may arrive less frequently than first link breakage, the total overhead for source side repair is at most $2N_L^S \cdot \lambda_1$. Note that in many cases a repair may be triggered at a much lower rate due to the use of the graph cache at the source which can be used to repair a broken link through another previously discovered neighboring microrouter.

2) Destination local repair: On the destination node side, μRP uses a adaptive *proactive* scheme for local route repair. After replying to an initial route discovery, the destination node D prepares to receive data packets. As it moves, it invokes a beaconing procedure to leave a "trail" in the microrouting network for data packets to follow. Periodically, D broadcasts a TRAIL packet with a TTL value of K. Upon receiving a TRAIL packet for the first time, a microrouter checks if it is within K hops from D and if so, inserts its ID into the packet and rebroadcasts it. Such K-hop broadcast will reach M_N if it is within K hops from D, and M_N stores the partial multi-hop route between itself and D in a trail cache. When M_N discovers that the direct hop to D is broken in trying to send a data packet, for example, as shown in Figure 2(b), it consults its trail cache to find a route towards the destination node. Figure 2(b) shows that microrouter M_N repairs the route using microrouter C to reach the destination at its new position D^*. Additionally, an error packet is sent back to the source whenever a route is repaired by a microrouter. The error packet contains the broken link as well as the repaired local route to the destination node. Upon receiving the error packet, the source node D removes the broken link from and stores the repaired route into its local graph cache and recalculates the shortest path to D when sending the next data packet. The overhead of destination repair is $N_K^D \cdot f_t$, where f_t is the frequency of the broadcast. To ensure that D does not move beyond K hops within each broadcast period, f_t should be at least $K\lambda_p$ where λ_p is the link failure rate of the last hop between the last microrouter and the mobile. f_t is thus proportional to $K\frac{v}{R}$.

When node D moves at high speed, it is possible that it has moved beyond K hops from M_N by the time the next data packet reaches M_N. In this case, the route is repaired and extended recursively till it reaches the destination.

Note that node trails can be used for repairing all connections intended for a particular node from multiple sources. Also, a mobile node initiates the periodic beaconing *only* if it is currently receiving data packets from other nodes, and periodic beaconing from a destination node can repair packets coming to it from multiple sources.

E. Pre-emptive Route Discovery

When S discovers a route to D initially, it records the original hop count of the route (H) in its *localization table*. As time progresses, due to route repair on both sides, this route could potentially shrink or grow in hop length. S keeps track of the hop length growth and if it grows to $H + h$ hops where $h = \alpha \cdot H$, the node performs a *pre-emptive* route discovery. The motivation behind this pre-emptive rediscovery is to prevent repaired routes from becoming arbitrarily long and thus to discover shorter routes to the destination. With an average speed of v, the average interval for the two end nodes to move beyond $h = \alpha \cdot H$ hops away is proportional to $\frac{h \cdot R}{2v}$. The overhead for the source node to discover a route of maximum length $H + h$ is approximately proportional to N_{H+h}^S packets. Thus the total overhead per connection due to pre-emptive route discovery is $N_{H+h}^S \cdot \frac{2v}{h \cdot R}$. α is a system parameter chosen to be 0.75 for reasons explained in Section IV-B.

In order to localize the query, the new route discovery is limited to a radius of $H + h$ hops which will cover the expected zone the destination node D would exist in.

F. Handling Microrouter Failures

Since microrouters have limited energy and can be deployed in hostile environments such as enemy territories or disaster recovery areas, they may fail unexpectedly while serving as intermediate hops for some active routes. μRP uses an intermediate repair mechanism to deal with such failures. When a delivery failure is detected by a microrouter when forwarding a data packet to another microrouter, it changes the packet to broadcast mode and initiates a 2-hop broadcast of this data packet to attempt to reach the next hop node *after* the failed node in the source route. In addition, it sends an error packet to the source to inform it of the link break. If the next hop node after the failed node in the source route receives the broadcast, it changes the data packet mode back to source routing and continues the transmission. Due to the low bandwidth in microrouting networks, congestion may cause failures to occur frequently. To compensate this, intermediate repair retries a unicast of the packet after a random backoff before initiating the 2-hop broadcast discussed above.

G. Memory Requirement

Each microrouter uses a constant amount of memory. It needs to store a sequence number per source node. Additionally each node has a small cache of m source routes for storing *node trails*. m is configured to be as small as 5 to conserve memory resources on the microrouters. Apart from this mini-cache, the microrouters store no other routing state. Note that unlike the DSR route cache in which each path is of length $O(\sqrt{N})$, the routes stored in this mini-cache are of constant length equal to the trail depth K making the total size $m \cdot K \cdot address\_byte\_size$. In the worst case, even if we use IP addresses for the microrouters, this cache only requires 40 bytes of memory for K=2.

IV. PERFORMANCE EVALUATION

In this section, we first describe our simulation methodology and then evaluate the routing performance of μRP.

A. Simulation Methodology

We implemented μRP in the Glomosim [21] simulator. Glomosim is a widely used simulator for wireless networks with a comprehensive radio model.

1) Microrouter Model and Deployment: We model our microrouters after MICA Motes [3] except that they do not have transducers. The simulator implements a 802.11 based MAC layer and the microrouters use this as the MAC protocol. 802.11 has been widely used in simulation evaluation of sensor networks [8], [1], [19]. In this study, we do not assume the use of topology control algorithms during the deployment of the microrouters. The devices are assumed to be sprayed and randomly placed in the target area. The experiments consider a data rate of 200Kbps and a 220m radio range for the microrouters and mobile nodes similarly as in [1]. This range

TABLE I

EFFECT OF DESTINATION REPAIR PARAMETER K. L=3.

K	0	1	2	3
Overhead	391,890	408,979	128,450	158,446
Dest. repair	0	1800	20,327	49,421
PDR	97.7	97.7	99.9	99.9
Delay	0.16	0.17	0.15	0.18
Drops ($M_n \rightarrow D$)	371	372	3	4

can already be achieved by the MICA2Dot [3] motes currently available, while newer 802.15.4 Motes will achieve a rate of 250Kbps.

2) Traffic and Mobility Models: We use the modified random waypoint mobility model [20] to simulate the movement of mobile nodes. Nodes move without pausing at a speed randomly chosen between 1 m/s and 9 m/s. Traffic is generated and received only by the mobile nodes and the microrouters only forward traffic. We assume constant bit rate (CBR) traffic, same as in previous MANET protocol comparison studies [2], [4]. Each mobile node initiates one random CBR connection to another mobile node with a packet rate of one 32-byte packet/sec. The simulation duration is 15 minutes.

3) Energy Model: Efficient usage of the limited energy of microrouters can extend the operation lifetime and connectivity for mobile users. Microrouters consume energy due to computation (packet processing, routing, etc.) and wireless communication. We focus on the wireless communication energy costs in this study. Since our work focuses on the routing layer, we do not assume a lower layer protocol that coordinates wake/sleep cycles of microrouters. Since a microrouter's radio is expected to be similar to that of a MICA Mote, we adopt the radio energy consumption parameters of the MICA2Dot Motes [3]. Specifically, the current drawn for transmission is $25mA$ and that for reception is $8mA$. The transmit and receive durations of each node were logged and the corresponding energy consumption calculated.

4) Metrics: We use the following four metrics to evaluate the performance of μRP: (1) *Routing overhead* — the number of control packets transmitted to discover and maintain routes between mobile nodes; (2) *Packet delivery ratio* (PDR) — the ratio of the number of data packets successfully received to the number of data packets sent; (3) *Delay* — the average time between transmission and reception of data packets. This metric accounts for all possible delays caused by buffering during route discovery latency, queuing at the interface queue, retransmission delays at the MAC, and propagation and transfer times; and (4) *Energy consumption* — the communication energy consumed during transmission and reception, for microrouters only.

B. Effect of Local Repair Ranges

In this section, we evaluate the effects of two important design choices for μRP: the sender repair depth L and the node trail depth K used in destination repair. The evaluation scenario consists of 20 mobile nodes continuously moving in an area of 16 km^2. Microrouters are deployed with a density of 10 microrouters per radio range.

Table I depicts the effect of varying K on the performance of μRP. Apart from the overall overhead, PDR, and delay, we also show the number of packet drops that occur between the last microrouter on a path and the destination mobile node as well as the overhead incurred due to TRAIL packets.

TABLE II

EFFECT OF SENDER REPAIR PARAMETER L. K=2.

L	0	1	2	3	4
Overhead	237,342	182,535	132,381	128,450	140,901
Sender repair	0	1,565	6,674	14,223	23,119
PDR	99.9	99.9	99.9	99.9	99.9
Delay	0.14	0.15	0.16	0.15	0.17
Query	297	201	115	106	108

TABLE III

EFFECT OF PRE-EMPTIVE REQUEST CONTROL FACTOR α. K=2 AND L=3.

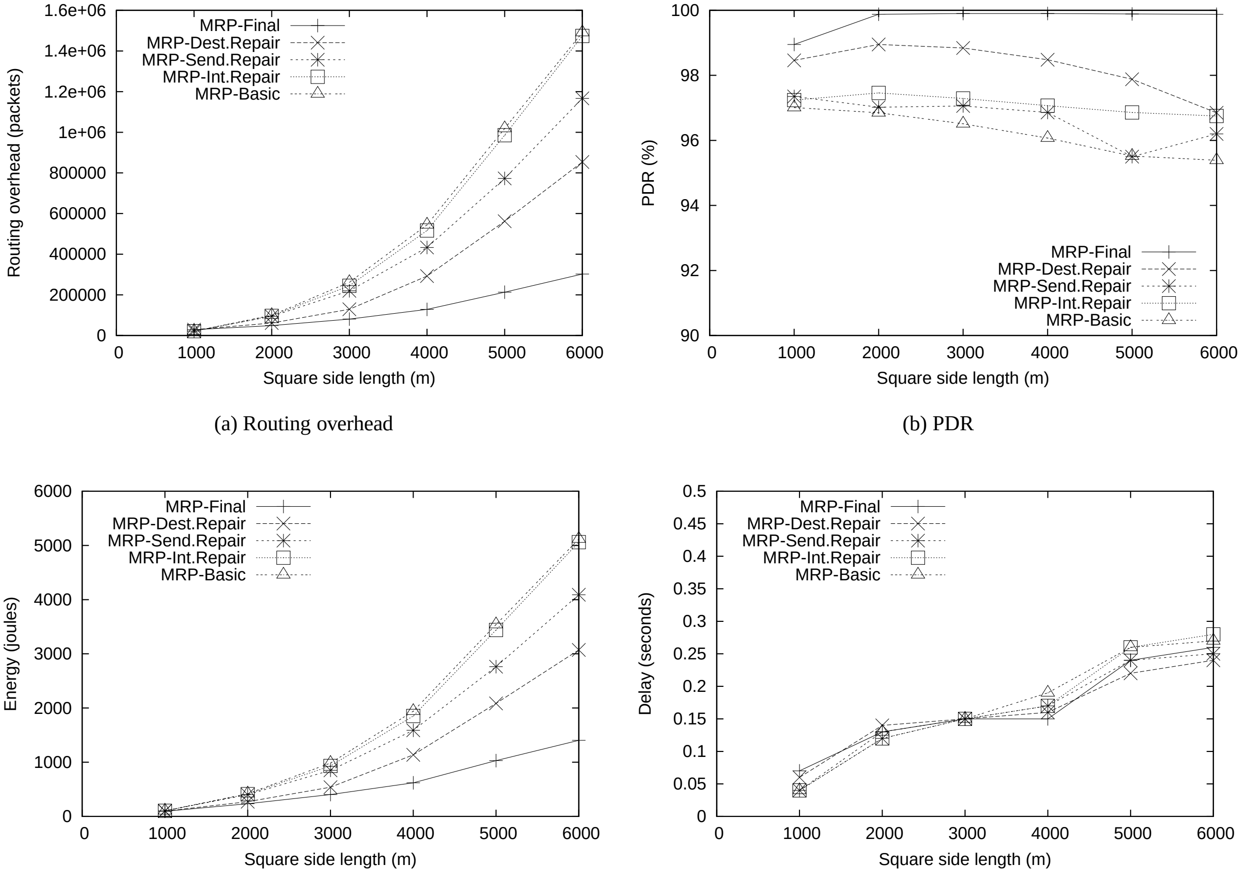
α	0.25	0.5	0.75	1
Overhead	161,467	134,996	128,450	128,035
PDR	99.9	99.9	99.9	99.9
Delay	0.15	0.16	0.15	0.17
PQuery	85	43	25	18
Avg. Path Len.	11.4	11.5	11.7	12.5

When node trails are not used (K=0), 371 packets are dropped due to breaks between the last microrouter and the destination, and μRP incurs an overhead of 391,890 packets. When node trails are used, K=2 is the best choice. At K=2, the number of drops reduce from 371 to 3 while the overall overhead reduces by 67%. This is because after each repair, a special repair error and the repaired local route are sent back to the source of the packet, avoiding costly route discoveries. Additionally, the PDR is increased by 2%. As K is increased further, the proactive beaconing overhead increases without an increase in rescued packets.

Table II depicts the effect of varying L on the performance of μRP. The repair overhead is due to NEIGHBORHOOD DISCOVERY and NEIGHBOR REPLY packets. The number of queries sent reflects the repair success rate. The lower the success rate of repair, the more the queries sent and thus the higher the total overhead.

The results show that a value of L=3 results in the lowest total overhead due to the lowest number of queries sent. Also, an increase beyond L=3 is not beneficial since the repair overhead grows without any increase in the success rate of the repair. Note that sender side repair is more flexible and robust since it is performed by a mobile node which can use its graph cache to repair routes aggressively.

Table III depicts the effect of varying α on the performance of μRP. In the table, PQuery is the total number of pre-emptive QUERY packets transmitted. The results show that a value of α=0.75 provides the best tradeoff between routing overhead, path length, and delay. As α increases, the number of pre-emptive queries sent decreases at the expense of higher path length and more route breaks. Increasing α beyond 0.75 results in increased delays and path lengths while not significantly reducing the overhead. This occurs due to the increased route errors due to longer paths with α=1 which offset the savings in pre-emptive queries. This result indicates that repairing paths for a longer period of time (using a larger value of α) can be counter productive. μRP achieves a good tradeoff between the routing overhead and the path length by using pre-emptive queries.



(a) Routing overhead

(b) PDR

(c) Energy consumption

(d) Delay

Fig. 3. Effect of local repair techniques on routing overhead, PDR, delay, and energy consumption. The x-axis corresponds to the side length of the square terrain in meters, e.g., 5000m corresponds to an area of 25 km^2.

C. Effect of Local Repairs

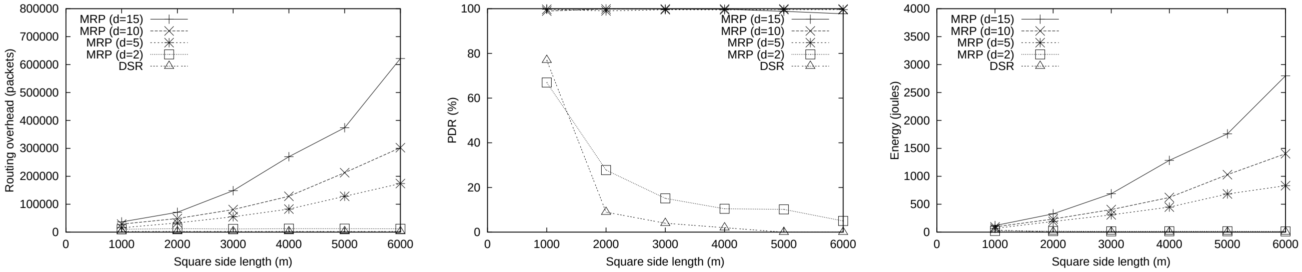
In this section, we evaluate the effectiveness of the three local repair mechanisms in μRP: sender side repair, intermediate repair, and destination side repair. The evaluation scenario consists of 20 mobile nodes continuously moving in terrains with varying areas ranging from 1 to 36 km^2. Microrouters are deployed with a density of 10 microrouters per radio range.

Figure 3 depicts the performance of various versions of μRP. The basic version of μRP depicted has no repair whereas the final version of μRP has all three local repair techniques. The three other curves depict the performance of μRP with each feature individually turned on. The following important observations can be made from the results. Firstly, μRP with all features turned on significantly reduces the overhead (e.g. by 80% in 36km^2) and energy consumption (e.g. by 72% in 36km^2) while always delivering 99% of the packets as the size of the microrouting network increases. This high data delivery rate is essential in critical military and disaster relief applications. The results show that stateless repair techniques used in μRP can significantly reduce the overhead and energy consumption in microrouting networks while keeping the microrouters as simple as possible. Secondly, destination side repair using node trails is the single feature that improves the performance of μRP_{basic} the most. This technique is effective in spite of using a small fixed size cache (to store trails) and not using any packet buffers (for salvaging packets). Thirdly, all versions of μRP have an acceptable delay which grows slowly as the network size increases. Delays are increased due to the limited bandwidth of microrouters (200Kbps) and large distances traveled by the packets. However, these delays are acceptable compared to reactive mobility-assisted schemes which may have unbounded delays.

D. Effect of Deployment Area and Density

This section evaluates the performance of μRP under varying microrouter density and deployment scenarios. These results are useful in determining the number of microrouters required for a given area. We vary the microrouter density d in a given area as 2, 5, 10 and 15 microrouters per radio range. As before, we vary the area of deployment from 1 to 36 km^2. In each chosen area, 20 mobile nodes are randomly distributed.

Figure 4 depicts the performance metrics for μRP as the terrain area is varied and the network becomes increasingly sparse. The results show that as expected with increased network area and minimal microrouting support ($d = 2$), the PDR falls quickly as the network area grows. The results show that a density of 10 microrouters per radio range is sufficient for adequate packet delivery with low overhead. Increasing d to 15 does not improve packet delivery bit increases the overhead and energy consumption. Interestingly, a d of 5 also provides good PDR with low overhead. However, we found that this density



(a) Routing overhead

(b) PDR

(c) Energy consumption

Fig. 4. Effect of microrouter density on routing overhead, PDR, delay, and energy consumption. The x-axis corresponds to the side length of the square terrain in meters, e.g., 5000 corresponds to an area of $25km^2$.

is less resilient to failures. We also show what the performance would be when no microrouters exist. This scenario involves running DSR on each mobile node in a sparse ad hoc network. As expected, due to the sparse connectivity and frequent partitions, DSR has the lowest PDR. The slightly higher PDR of DSR compared to MRP ($d = 2$) in the smallest network size is because the density of mobile nodes is high enough that they can form multi-hop routes.

Thus, the provisioning of microrouters should be based on both the connectivity required as well as the failure rate. Although a reduced density of microrouters may provide adequate connectivity, the resilience of the microrouting network to failure reduces as well.

V. Related Work

Previous research on data delivery in sparse MANETs have largely focused on exploiting the mobility of nodes to buffer data packets during network partitions and forward data packets when the network is connected again. Such mobility-based protocols are thus suitable for delay-tolerant applications such as sensor data collection. These approaches can be classified as *proactive* and *reactive* schemes. Reactive schemes ([18], [5]) typically exploit the existing mobility of the nodes to buffer and deliver messages across network partitions, while proactive schemes ([11], [22], [6]) require nodes to move in a controlled manner to ensure network connectivity.

VI. Conclusions

In this paper, we proposed microrouting as a new paradigm for real-time communication in sparse mobile ad hoc networks without imposing any controlled mobility on the participating nodes. Microrouting employs microrouting networks consisting of low cost and easy to deploy microrouters that are similar to sensors but without transducers to provide the connectivity in a geographic area in which a small number of mobile hosts will be sparsely deployed. To demonstrate the viability of the resulting hybrid microrouting network architecture, we designed and evaluated the microrouting protocol which takes advantage of the immobility of microrouters to improve the robustness of the routes, while maintaining minimal state at the microrouters to satisfy their energy and memory constraints. Our evaluation results showed that microrouting networks running the microrouting protocol effectively extend the connectivity of sparse ad hoc networks.

Acknowledgment

This work was supported in part by NSF grant ANI-0338856.

References

[1] B. Blum. et al., An entity maintenance and connection service for sensor networks. In *Proc. of MobiSys*, May 2003.

[2] J. Broch. et al., A performance comparison of multi-hop wireless ad hoc network routing protocols. In *Proc. of ACM MobiCom*, October 1998.

[3] Crossbow Technology Inc. www.xbow.com.

[4] S. R. Das, C. E. Perkins, and E. M. Royer. Performance comparison of two on-demand routing protocols for ad hoc networks. In *Proc. of IEEE INFOCOM*, March 2000.

[5] J. A. Davis, A. H. Fagg, and B. N. Levine. Wearable computers as packet transport mechanisms in highly-partitioned ad-hoc networks. In *Proc. of IEEE ISWC*, 2001.

[6] D. K. Goldenberg. et al., Towards mobility as a network control primitive. In *Proc. of ACM MobiHoc*, 2004.

[7] Y.-C. Hu and D. B. Johnson. Caching strategies in on-demand routing protocols for wireless ad hoc networks. In *Proc. of ACM MobiCom*, August 2000.

[8] C. Intanagonwiwat, R. Govindan, and D. Estrin. Directed diffusion: a scalable and robust communication paradigm for sensor networks. In *Proc. of ACM MobiCom*, August 2000.

[9] D. B. Johnson and D. A. Maltz. *Dynamic Source Routing in Ad Hoc Wireless Networks*. Kluwer Academic, 1996.

[10] J. Kahn, R. Katz, and K. Pister. Emerging challenges: Mobile networking for smart dust. *J. Comm. Networks*, 2(3):188–196, Sept. 2000.

[11] Q. Li and D. Rus. Sending messages to mobile users in disconnected ad-hoc wireless networks. In *Proc. of ACM MobiCom*, August 2000.

[12] M. K. Marina and S. R. Das. Routing performance in the presence of unidirectional links in multihop wireless networks. In *Proc. of ACM MobiHoc*, September 2002.

[13] S.-Y. Ni. et al., The broadcast storm problem in a mobile ad hoc network. In *Proc. of ACM MobiCom*, August 1999.

[14] V. D. Park and M. S. Corson. A highly adaptive distributed routing algorithm for mobile wireless networks. In *Proc. of IEEE INFOCOM*, April 1997.

[15] C. E. Perkins and P. Bhagwat. Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers. In *Proc. of ACM SIGCOMM*, August 1994.

[16] C. E. Perkins and E. M. Royer. Ad hoc on-demand distance vector routing. In *Proc. of IEEE WMCSA*, February 1999.

[17] A. Savvides, C.-C. Han, and M. B. Strivastava. Dynamic fine-grained localization in ad-hoc networks of sensors. In *Proc. of ACM MobiCom*, July 2001.

[18] A. Vahdat and D. Becker. Epidemic routing for partially connected ad hoc networks. Technical Report CS-200006, Duke University, 2000.

[19] F. Ye, H. Luo, J. Cheng, S. Lu, and L. Zhang. A two-tier data dissemination model for large-scale wireless sensor networks. In *Proc. of ACM MobiCom*, September 2002.

[20] J. Yoon, M. Liu, and B. Noble. Random waypoint considered harmful. In *Proc. of IEEE INFOCOM*, April 2003.

[21] X. Zeng, R. Bagrodia, and M. Gerla. Glomosim: A library for parallel simulation of large-scale wireless networks. In *Proc. of PADS Workshop*, May 1998.

[22] W. Zhao, M. Ammar, and E. Zegura. A message ferrying approach for data delivery in sparse mobile ad hoc networks. In *Proc. of ACM MobiHoc*, May 2004.