E^2R: Energy Efficient Routing for Multi-hop Green Wireless Networks

Ting Zhu, Don Towsley
Department of Computer Science, University of Massachusetts Amherst
{tzhu, towsley}@cs.umass.edu

***Abstract*— Wireless communication consumes a significant amount of energy in wireless networks. Most multi-hop wireless routing protocols are adopted from wired network routing protocols and use *pre-selected* single or multiple minimum cost paths to forward data packets. However unlike wired networks, a wireless network has unique characteristics such as unreliable wireless links and multiple receivers associated with a single packet transmission. These particular characteristics of wireless network make it energy inefficient by using preselected paths to forward data packets. In this paper we propose a scalable, opportunistic, and energy efficient routing protocol (E^2R) that uses an opportunistic forwarding scheme to deliver control messages and data packets in a multi-hop wireless network. To further reduce the overhead of control message, we introduce a novel greedy forwarding algorithm and an efficient self-suppression scheme. Extensive analysis and simulations show that E^2R can provide high packet delivery ratio and effectively reduce redundant packet transmissions which results in less energy consumption.**

I. Introduction

Routing is one of the most important control functions in a wireless network. In order to build green wireless networks, it is very important to design an energy efficient routing protocol. Most multi-hop wireless routing protocols are adopted from wired network routing protocols which use conventional graph theory to *pre-select* minimum cost paths.([19], [10], [20], [16], [12], [24], [13], [14], [9]) In these protocols, wireless links are represented by edges and a path is denoted by a sequence of edges. However, unlike wired networks, wireless communication is broadcast in nature. In other words, after a sender broadcasts a packet, the sender's neighbors all receive the same packet. This broadcast nature provides rich spatial diversity that has not been utilized by most routing protocols to reduce energy consumption in wireless networks. Moreover, wireless links are more unstable and unreliable than wired links. These instabilities and failure characteristics of wireless links can cause poor end-to-end performance (e.g., small end-to-end packet delivery rates), a large number of packet retransmissions, and frequently route rediscovery when pre-selected paths are not available for data delivery.

In order to overcome link instability and utilize broadcast benefit in wireless networks, some opportunistic routing protocols ([5], [3], [2], [26]) have been proposed. However these opportunistic routing protocols pre-select forwarding candidates beforehand and the selection of an appropriate forwarder list is not trivial especially for multiple source-destination pairs in a large scale multi-hop wireless network. Moreover, in order to compute the forwarder list, a large number of control messages are required, which results in high energy consumption.

In this paper we propose a scalable, opportunistic, and energy efficient routing protocol (E^2R) that utilizes an opportunistic forwarding scheme to deliver control messages (e.g., route metric discovery packets) and data packets. Unlike other opportunistic routing protocols, E^2R neither uses pre-selected static paths nor prepares forwarding candidates. Therefore E^2R can reduce control message overhead caused by maintenance of forwarding paths and forwarder lists. Moreover, E^2R also utilizes spatial diversity (i.e., broadcast nature of wireless communication) by introducing an implicit probabilistic multi-path routing mechanism that reduces the probability of route discovery and local route repair due to link failure. E^2R is designed to increase end-to-end performance with low control message overhead. More specifically, our major contributions are as follows:

- To our knowledge, this is the first wireless routing protocol that does not specify any paths to forward the data packets.
- Our design is simple and robust. All operations only require one-hop neighbor information, making our protocol highly distributed and scalable.
- We evaluate our protocol in different settings. The results indicate that our design is reliable and energy efficient.

The rest of the paper is organized as follows. Section II presents the main idea of E^2R. Route metric discovery and data delivery schemes are introduced in Section III. The performance of E^2R is evaluated in Section IV. We discuss related work in Section V and conclude the paper in Section VI.

II. Overview of E^2R

The key idea of E^2R is to exploit spatial diversity (i.e., broadcast nature) in wireless networks rather than specifying data delivery paths (i.e., next hops). In E^2R, the source node does not specify any particular paths. Both route metric discovery packets and data packets are delivered through broadcast. Nodes that have better opportunities to deliver the packets are automatically selected to forward the control packets and data packets. Similar to other wireless routing protocols, E^2R operates in two phases: route metric discovery and data delivery. In this section, we briefly introduce these two phases and discuss the design challenges in these two phases. The detailed design is described in Section III.

978-1-4244-9920-5/11/$26.00 ©2011 IEEE

A. Route Metric Discovery Phase

In this phase, route metric discovery packets are delivered via broadcast. There are two challenges that have to be carefully addressed. The first challenge concerns how to prevent the repeated flooding of route metric discovery packets when the network is first constructed. At the early stage of network construction, route metric discovery (RMD) packets must be delivered to every node inside the network since the source node and intermediate nodes do not know the direction of the destination node. Controlled flooding schemes (e.g., each node always forwards the newly received RMD packet once or multiple times) can be applied. However flooding schemes can cause many unnecessary rebroadcasts of RMD packets especially when the network density is high. To address this challenge, greedy forwarding algorithm is devised for the distribution of RMD packets. The current forwarding node's covered neighbor list is embedded in the control packets. Here, we say a node is *covered* if it has already received the packet. Whenever a node receives the packet it marks whether its neighbors have already been covered based on the covered neighbor list in the packet. Then the node sets a waiting (i.e., backoff) time based on the number of its neighbors that have not been covered. Intuitively, the node with more uncovered neighbors should have higher priority to forward the packet. Therefore, if a node has a large number of uncovered neighbors, its waiting time should be shorter.

The second challenge concerns how to reduce redundant transmissions of the route metric reply (RMREP) packet. At the stage after the destination node receives the route metric discovery (RMD) packets, the destination node tries to deliver the route metric reply (RMREP) packet back to the source node. Similarly, controlled flooding schemes will introduce redundant transmissions. This challenge differs from the first one, because the destination node and the intermediate nodes already know the direction of the source node because they have received the RMD packet initiated from the source node. Therefore, the challenge is to utilize this knowledge to further reduce redundant transmissions. To address this challenge, we introduce an efficient self-suppression scheme (described in Section III-B.2), which suppresses the forwarding of the route metric reply (RMREP) packet based on the route metric.

B. Data Delivery Phase

After the route metric discovery phase, the source node obtains a route metric for the new destination. The source node now needs to deliver data packets to the destination node. The challenge is how to utilize spatial diversity to improve end-to-end performance (i.e., end-to-end packet delivery ratio and delay) and reduce energy consumption (by reducing the total number of packet transmissions inside the networks).

In order to address this design challenge, we introduce a forwarder self-selection scheme, which is similar to the relay selection scheme used in [1]. The source node attaches the obtained route metric to data packets and broadcasts the data packets without designating forwarding nodes. Upon receiving the data packets, the nodes that have smaller route metric value than the attached route metric value are eligible to further forward the data packets. Before these nodes forward the received data packet, they wait for a small amount of time to do back-off based on their own route metric values for the destination. For example, the node with smaller route metric value will have shorter back-off time and selects itself to forward the received data packets. During the back-off time interval, these nodes listen to the channel and suppress the forwarding of received data packets if they overhear data packets forwarded by a node with a smaller route metric value. When the back-off timer fires, the node updates the route metric value in the data packets by attaching its own route metric and forwards the data packets.

III. Design of E^2R

This section describes the detailed design of E^2R which contains maintenance state, route metric discovery, and data delivery.

A. Maintenance State

A node enters a maintenance state after it is deployed. While in the maintenance state, every node maintains its neighboring node information and the route metric from itself to all the other nodes inside the network. Like other wireless routing protocols, every node inside the network periodically sends out HELLO messages to indicate the existence of the node. Moreover, every node uses the HELLO messages received from its neighboring nodes to update its neighboring node set ($N(i)$).

Besides neighboring node information, every node also maintains the route metric from itself to all other nodes inside the network. We note that E^2R is compatible with all other route metrics (e.g., ETX [4] or ETT [6]) that have been proposed. For example, we can give nodes with smaller ETX values higher priority to forward received packets. Without loss of generality, in this paper we use distance vector (e.g., hop count) as the route metric. If a node s needs to route data packets to a destination node and there is no route metric maintained at node s for that node, s will initiate the route metric discovery process. To reduce the transmission of control messages, the establishment and maintenance of route metrics is on demand. It is triggered when the source can not deliver data packets to the destination.

B. Route Metric Discovery

The route metric discovery process includes two stages: the stage of disseminating route metric discovery (RMD) packets and the stage of propagating route metric reply (RMREP) packets.

1) RMD Packets Dissemination: During RMD discovery, the source node originates a new RMD packet if the source node needs to route data packets to a destination node and no route metric is available to the destination node. The RMD packet contains the source node id (s), packet id (Pid), source node's covered neighbor list (CN_s), destination node id (d), route metric (R), and route metric sequence number from source to destination (S_s^d). When a node, i, receives an RMD packet, i processes the RMD packet based on the Greedy

Forwarding Algorithm (shown in Algorithm 1). In the first step, i updates its uncovered neighbor set ($UN(i)$) by using the source node's covered neighbor list which is embedded in the RMD packet (Line 1). Here uncovered neighbor set of i is the set of i's neighbors that have not received the RMD packet.

If i is an intermediate node and this RMD packet is the one that i receives for the first time (Lines 2-4), there are three possible cases:

- **Case 1:** i has a fresher route metric to the destination than the source has. In other words, the sequence number of the route metric from i to the destination (S_i^d) is larger than the sequence number of the route metric from the source to the destination (S_s^d). In this case, i directly returns a route metric reply (RMREP) packet with S_i^d(Lines 5-7).
- **Case 2:** All neighbors of i have received the RMD packet (i.e., $UN(i) = \phi$). In this case, there is no need for i to rebroadcast the RMD packet. Therefore, i drops the RMD packet (Lines 8 and 9).
- **Case 3:** i does not have a fresher route metric to the destination than the source and some neighbors of i are uncovered. In this case, i sets the back-off timer with time interval $T_{backoff}$. Here the value of $T_{backoff}$ is inversely proportional to the size of the uncovered neighbor set $UN(i)$. The larger the uncovered neighbor set $UN(i)$, the smaller the value of $T_{backoff}$. Therefore we allow the node (assume node j) that has a larger number of uncovered neighbors to rebroadcast the RMD packet first. When i overhear j's rebroadcast during i's back-off time interval, i updates its uncovered neighbor set $UN(i)$. If all the neighbors of i are covered, then i drops the RMD packet (Lines 10-14). Otherwise, i will update the covered neighbor set $CN(s)$ with i's covered neighbor set $CN(i)$ and increases the value of route metric R (if the route metric is hop count, then increases the number of hop count by 1) in the RMD packet and rebroadcasts the RMD packet (Lines 15 and 16).

If i is the destination node and this RMD packet is the one that i receives for the first time, i will send back an RMREP packet (Lines 20-23).

2) RMREP Propagation: As discussed in the previous subsection, route metric reply (RMREP) packets are generated either by the destination or by the intermediate node which has a fresher route metric to the destination than the source node. The RMREP contains the destination node id (d), source node id (s), packet id (Pid), route metric (R), and route metric sequence number from source to destination (S_s^d). Here the route metric R is the number of hops from source to destination. Since the RMREP does not contain any intermediate node information, it will be unnecessarily propagated to all nodes inside the network, which will result in a large amount of energy waste. In order to address this issue, we introduce an efficient self-suppression scheme which contains two rules:

- **Rule 1:** If the node has rebroadcasted the source originated route metric discovery (RMD) packet, this node is eligible to forward the route metric reply (RMREP) packet. No other node is eligible to forward the RMREP packet. This rule avoids unnecessary rebroadcasts from nodes far from both the source and the destination.
- **Rule 2:** If the route metric of the node to the destination is larger than the route metric in the RMREP packet, this node is not eligible to forward the RMREP packet. For example, if we use hop count as route metric. If the node has a larger number of hops to the destination than the source has, the source will not use this node to forward the data packet. Therefore, there is no need to let this node forward the RMREP packet.

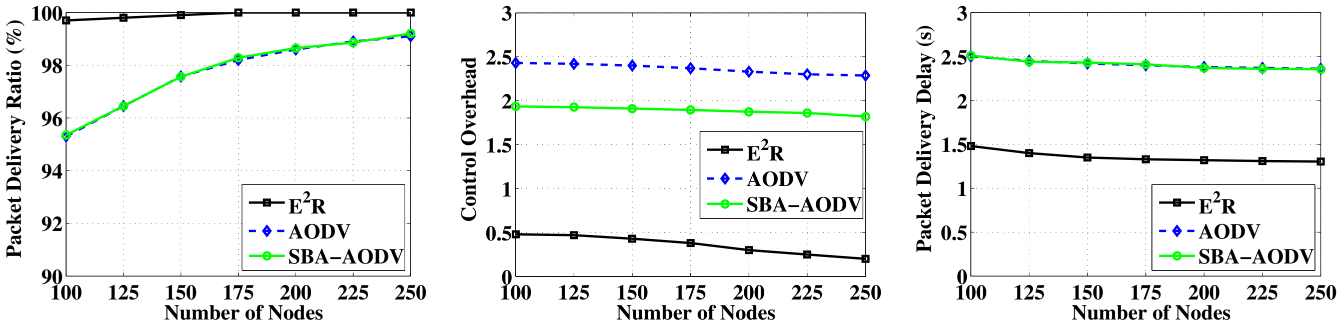
Algorithm 1 `Greedy Forwarding Algorithm`

1: Update $UN(i)$ based on $CN(s)$
2: **if** ($i \neq s$) and ($i \neq d$) **then**
3:   // i is an intermediate node
4:   **if** $newRMD$ **then**
5:     **if** $S_i^d > S_s^d$ **then**
6:       //i has a fresher route metric to destination
7:       i sends RMREP with S_i^d
8:     **else if** all neighbors in N_i received the RMD **then**
9:       drop RMD
10:     **else**
11:       // i has an out-dated route metric
12:       wait for $T_{backoff}$ period assigned based on $UN(i)$
13:       **if** a neighbor forwarded RMD and $UN(i) = \phi$ during $T_{backoff}$ **then**
14:         drop RMD
15:       **else**
16:         $CN(s) \leftarrow CN(i)$, $R \leftarrow R + 1$, forward RMD
17:       **end if**
18:     **end if**
19:   **end if**
20: **else if** $i = d$ and new RMD **then**
21:   // i is the destination node
22:   send RMREP
23: **end if**

C. Data Delivery

After the source receives the RMREP packet and obtains the route metric, the source needs to forward the data packets to the destination node. Unlike other wireless routing protocols (such as AODV) and forwarding methods (such as ExOR), in E^2R, the source node does not need to designate its next hop or a forwarder list within the data packets. The source node only attaches the obtained route metric to data packets and then broadcasts the data packets. In this scheme, the forwarders of the data packets are selected by the intermediate nodes themselves. We call it a forwarder self-selection scheme.

When an intermediate node i receives the data packets, i compares its own route metric value with the value of the route metric embedded in the data packets. If i's route metric value is smaller than the value of the route metric embedded in the data packets, i selects itself to be a potential forwarder of the



(a) Packet Delivery Ratio (b) Control Overhead (c) Packet Delivery Delay

Fig. 1. Impact of Node Density

data packets. However, i does not know whether its neighbors also received the data packets and have smaller route metric values than i. In order to handle this problem, we introduce a back-off mechanism and design the back-off time interval based on the route metric values. The smaller the value of the route metric a node has, the shorter the back-off time this node experiences. During the back-off time interval, i listens to the channel and suppresses the forwarding of the data packets if i overhears that one of its neighbors with a better route metric already forwards the data packets. If i does not overhear its neighbors' forwarding and its back-off timer fires, i updates the route metric in the data packets with its own route metric and then rebroadcasts the data packets.

When the destination node receives the data packets, the destination node returns an acknowledgement to the source node. The propagation of the acknowledgement is similar to the propagation of the route metric reply (RMREP) packet. Due to space constraints, we do not describe it in detail in this paper.

IV. Performance Measurement

In this section we extensively evaluate the performance of E^2R through ns-2 simulations. We compare the performance of E^2R with the following two approaches:

- **AODV** [20]: It is a widely used baseline to represent on-demand wireless routing protocol.
- **SBA-AODV**: We modify the AODV protocol by using the scalable broadcast algorithm (SBA) [17] to do route discovery. In SBA, each node waits for a randomly assigned time interval before rebroadcasting the first time received packet. The waiting time interval is based on the ratio of the maximum degree among its neighbors ($d_{N_{max}}$) to the number of neighbors (d_{me}) this node has (i.e., $\frac{d_{N_{max}}}{d_{me}}$).

Here, we do not compare E^2R with opportunistic routing protocols. This is because most of opportunistic routing protocols (such as ExOR [2]) need to maintain forwarder list and do not work in mobile environment.

Three metrics are used to evaluate the protocols:

- **Packet Delivery Ratio**: It is the ratio of the total number of data packets delivered to the destination nodes to the total number of data packets generated by the source nodes.
- **Control Overhead**: It is the number of control (e.g., routing) packets transmitted per data packet per node during the simulation. Here, control packets include route metric discover packets, route metric reply packets, and route error packets. Since packet transmissions consume energy, lower control overhead indicates lower energy consumption.
- **Packet Delivery Delay**: It is the average duration from the time that the source nodes initiates the data packet to the time that the packet is received by the destination nodes.

A. Simulation Setup

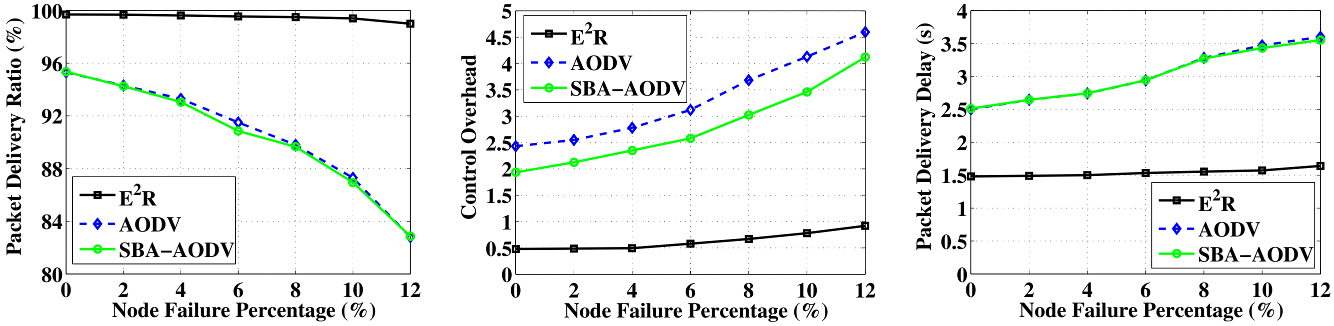
In the simulation, we randomly deployed 100 nodes inside a $1500m \times 1500m$ square field and used the Realistic Mobility Model in which node velocities and directions of movement are based on probability distributions [11]. According to default physical parameters in ns-2, the communication range of each node is 250 meters and the radio propagation model is the two-way ground model [22].

Ten source and destination node pairs are selected. The source nodes send out a 1024-byte data packet every 5 seconds. The total simulation time was set at 2200 seconds. In order to avoid the initialization bias of the system state on the routing operation, the source nodes did not send out the data packets during the first 100 seconds, but only exchange HELLO messages between neighboring nodes to establish neighborhood information. Similarly, to make sure that all broadcast packets propagate throughout the network, the source stops sending out the data packet after 2100 seconds. Every data point on a graph represents the averaged value of 20 runs, and 95% confidence intervals for the data are within $3\% \sim 7\%$ of the mean shown in the graph. For each run, we use the same network topology and traffic pattern for all the routing protocols. Unless explicitly stated otherwise, we used the above default values in our simulation.

B. Impact of Node Density

In this experiment, we analyzed the effect of node density by varying the number of nodes in the field from 100 nodes to 250 nodes.

Figure 1(a) shows that the packet delivery ratio of all the protocols increases as the network density increases. When the node density increases, E^2R always has more than 99.7%



(a) Packet Delivery Ratio

(b) Control Overhead

(c) Packet Delivery Delay

Fig. 2. Impact of Node Failure

reliability, while the mean value of packet delivery ratio of AODV and SBA-AODV varies from 95.3% to 99.2%. This is because AODV and SBA-AODV specifically select nodes to forward the data packets. In a sparse network, when the selected forwarding nodes move out of radio range of each other, the data packets can not be delivered. However, nodes running E^2R protocol select themselves to be forwarding nodes. When a forwarding node moves out of range, other nodes will select themselves to be forwarding nodes. Therefore, the packet delivery ratio only decreases by 0.3% as the network density decreases.

Figure 1(b) shows that the control overhead of all the protocols decreases slightly as the network density increases. The reason is that the control overhead is amortized over the number of nodes inside the network as the number of the nodes increases. Since nodes running AODV have to rebroadcast the first time received route request packet, AODV always has highest control overhead. Compared with AODV, E^2R's control overhead is 90% less.

Figure 1(c) shows that the average packet delivery delay of all the protocols slightly decreases as the network density increases. By allowing the node with better route metric to forward the data packet first, E^2R achieves average smaller packet delivery delays. Compared with AODV, E^2R reduces packet delivery delay by more than 44%.

C. Impact of Node Failure

In wireless networks, nodes may run out of energy which causes node failure. In this section, we study the effects of varying node failures. The percentage of node failure varies from 0% to 12%.

Figure 2(a) shows that the packet delivery ratio of all protocols decreases as node failure percentage increases. Since AODV and SBA-AODV depend heavily on preselected forwarding nodes to forward the data packets, the failure of these forwarding nodes significantly affects the packet delivery ratio. In contrast, the node failure has little effect on E^2R. This is because when a node failure happens, the other nodes will select themselves to forward the data packets. Therefore, the packet delivery ratio of E^2R is always above 99%, even 12% of the nodes inside the networks are failed.

Figure 2(b) shows that when the node failure percentage increases, the control overhead of all the protocols increases. The reason is that as node failure percentage increases, more route discovery packets, route reply packets, and route error packets are generated and these control packets are generated more frequently. Since E^2R does not select any particular nodes or paths to forward the data packets, the control overhead of E^2R is one quarter that of the other two protocols.

Figure 2(c) shows that the packet delivery delay of all the protocols slightly increases as the node failure percentage increases. This is because node failures trigger a new round of route discovery which will introduce the delay of packet delivery. Since the nodes running E^2R automatically select themselves to forward the data packets, node failure has little effect on E^2R. Compared to AODV and SBA-AODV, E^2R reduces the packet delivery delay by more than 41% in all the scenarios.

V. Related Work

Wireless networks is an active research area and many wireless routing protocols and scheduling policies have been proposed ([10], [20], [19], [16], [9], [13], [12], [14], [24], [23], [27], [7], [28]). Most of these routing protocols *preselect* a minimum cost single path or multiple alternate paths and use the preselected static path(s) to forward data packets. At any specific time, these routing protocols use unicast to forward data packets inside a single path even though these protocols discovered multiple alternate paths during route discovery process. Unlike the previous approaches, energy efficient routing (E^2R) protocol neither preselects any single or multiple alternate paths nor maintains next hop information. Instead, E^2R delivers data packets using broadcast and simultaneously utilizes all the neighboring nodes to forward data packets.

On the other hand, broadcasting protocols have already been extensively investigated([15], [8], [17], [21], [29], [18]). The literature in broadcasting protocol designs can be classified into two categories: deterministic approaches and probabilistic approaches. In the deterministic approaches, a fixed node within a connected dominating set is determined as a forwarding node. These approaches are also called fixed-forwarder approaches. In these approaches, the connected dominating set is calculated by using global or local information. In a probabilistic approach, when a node receives a packet, it forwards the packet with probability p. The value of p is determined by relevant information gathered at each node. Simple probabilistic approaches predefine a single probability for every node to rebroadcast the received packet. When running the above protocols in a network with different node densities, the nodes in a dense area may receive a lot of redundant transmissions. More complicated and efficient protocols, such as distance-based and location-based [15] schemes, use either area or precise position information to reduce the number of redundant transmissions.

Despite this rich literature, the existing broadcasting approaches try to disseminate packets from one node to all the other nodes inside the network by using broadcast. However, E^2R tries to use broadcast to forward packets from a source node to a destination node. It is an end-to-end routing protocol design.

ExOR[2] is an influential opportunistic routing protocol for wireless mesh networks. ExOR preselects prioritized forwarding candidates and each packet carries those forwarding candidates list. Only receivers in the forwarding list forward packets in the order of forwarding priority estimated based on the proximity (measured using ETX[4]) to the destination.

Zifei and Srihari[26] has proposed expected any-path transmissions (EAX) system. Westphal has proposed opportunistic routing in dynamic ad hoc networks (OPRAH protocol) [25]. OPRAH prepares multiple paths for a destination and each packet carries forwarding node list as ExOR did. These opportunistic routing protocols either preselect forwarder list or prepare multiple paths and use them to forward data packets. While E^2R does not need to preselect any path. Forwarding nodes are selected by the intermediate nodes themselves at the time of data delivery. Furthermore these protocols mainly focuses on data plane improvement with underlying link state routing protocol or AODV-style control plane protocols, while E^2R simultaneously focuses on the reduction of control overhead and the enhancement of data delivery ratio.

VI. Conclusion

In this paper, we introduce a scalable, opportunistic, and energy efficient routing protocol (E^2R) that does not need to preselect any paths or forwarding nodes. By using greedy forwarding algorithm to disseminate route metric discovery packets and applying efficient self-suppression scheme, E^2R significantly reduces the number of control packets which in turn reduces energy consumption on transmitting these packets. Moreover, E^2R effectively utilizes spatial diversity and uses the forwarder self-selection scheme to select the forwarding nodes and forward data packets using broadcast.

We also performed extensive simulation with various network configurations to reveal the performance of E^2R. The results show that the E^2R protocol can provide high packet delivery ratio, low control overhead, and low packet delivery delay in unreliable environments. Moreover, by reducing the number of packet transmissions, E^2R protocol can effectively reduce the energy consumption and make it an energy efficient routing protocol for multi-hop green wireless networks.

VII. Acknowledgements

This work was supported by NSF under grant #1019343 to CRA for the CI Fellows Project and grant CNS-0721861.

References

[1] D. R. A. L. A. Bletsas, A. Khisti. A simple cooperative diversity method based on network path selection. In *IEEE JSAC, Vol. 24, No. 3, pp. 659-672, March 2006.*

[2] S. Biswas and R. Morris. Exor: Opportunistic routing in multi-hop wireless networks. In *SIGCOMM '05*.

[3] R. Choudhury and N. Vaidya. Mac-layer anycasting in ad hoc networks. *ACM SIGCOMM Computer Communication Review, 34:75–80*, 2004.

[4] D. D. Couto, D. Aguayo, J. Bicket, and R. Morris. High-throughput path metric for multi-hop wireless routing. *In MobiCom '03*.

[5] D. S. J. De Couto, D. Aguayo, B. A. Chambers, and R. Morris. Performance of multihop wireless networks: Shortest path is not enough. In *HotNets-I*, Princeton, New Jersey, October 2002.

[6] R. Draves, J. Padhye, and B. Zill. Routing in multi-radio, multi-hop wireless mesh networks. *In Mobicom '04*.

[7] Y. Gu, T. Zhu, and T. He. Esc: Energy synchronized communication in sustainable sensor networks. *In ICNP '09*.

[8] Z. Haas, J. Halpern, and L. Li. Gossip-based ad hoc routing. In *INFOCOM '02*.

[9] P. Jacquet, P. Muhlethaler, and A. Qayyum. Optimized link state routing protocol. In *IETF MANET, Internet draft*, 1998.

[10] D. B. Johnson and D. A. Maltz. Dynamic source routing in ad hoc wireless networks. In *Mobile Computing*. Kluwer Academic Publishers, 1996.

[11] A. Kamal and J. Al-Karaki. A new realistic mobility model for mobile ad hoc networks. In *ICC '07*.

[12] S. Lee and M. Gerla. Split multipath routing with maximally disjoint paths in ad hoc networks. *In Proceedings of the IEEE ICC*, 2001.

[13] S.-J. Lee and M. Gerla. AODV-BR: Backup routing in ad hoc networks. In *WCNC '00*, 2000.

[14] M. K. Marina and S. R. Das. Ad hoc on-demand multipath distance vector routing. *WCMC '01*, 2001.

[15] S. Ni, Y. Tseng, Y. Chen, and J. Sheu. The broad-cast storm problem in a mobile ad hoc network. In *MobiCom '99*.

[16] V. D. Parka and M. S. Corson. A highly adaptive distributed routing algorithm for mobile wireless networks. In *INFOCOM '97*.

[17] W. Peng and X. Lu. On the reduction of broadcastredundancy in mobile ad hoc networks. In *MOBIHOC*, 2000.

[18] W. Peng and X. Lu. Ahbp: An efficient broadcastprotocol for mobile ad hoc networks. In *Journal of Computer Science and Technology*, 2002.

[19] C. Perkins and P. Bhagwat. Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers. In *SIGCOMM '94*.

[20] C. Perkins and E. Royer. Ad hoc on demand distance vector routing, mobile computing systems and applications. In *WMCSA '99*.

[21] A. Qayyum, L. Viennot, and A. Laouiti. Multipointrelaying: An efficient technique for flooding in mo-bile wireless networks. Technical report, 2000.

[22] T. Rappaport. Wireless communications: Principles and practice. *Prentice Hall*, 1996.

[23] L. Tassiulas and A. Ephremides. Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks. In *IEEE Transactions on Automatic Control, vol. 37, no. 12, pp. 1936-1948, Dec. 1992.*

[24] A. Valera, W. Seah, and S. Rao. Cooperative packet caching and shortest multipath routing in mobile ad hoc networks. In *INFOCOM '03*.

[25] C. Westphal. Opportunistic Routing in Dynamic Ad Hoc networks. *In Proc. of IEEE MASS*, Oct. 2006.

[26] Z. Zhong, J. Wang, S. Nelakuditi, and G. Lu. On selection of candidates for opportunistic anypath forwarding. *Mobile Computing and Communications Review 10(4):*, pages 1–2, 2006.

[27] T. Zhu, Y. Gu, T. He, and Z.-L. Zhang. eshare: A capacitor-driven energy storage and sharing network for long-term operation. *In SenSys '10*.

[28] T. Zhu and M. Yu. A secure quality of service routing protocol for wireless ad hoc networks. In *IEEE GLOBECOM '06*.

[29] T. Zhu, Z. Zhong, T. He, and Z.-L. Zhang. Exploring link correlation for efficient flooding in wireless sensor networks. In *Proceedings of the 7th USENIX/ACM Symposium on Networked Systems Design and Implementation (NSDI '10)*, 2010.