

The Impact of Traffic Patterns on the Overhead of Reactive Routing Protocols

Nianjun Zhou, Huaming Wu and Alhussein A. Abouzeid
 Dept of Electrical, Computer and Systems Engineering
 110 Eighth St, Rensselaer Polytechnic Institute
 Troy, New York 12180, USA
 email:zhoun@rpi.edu, wuhm@rpi.edu, abouzeid@ecse.rpi.edu

Abstract

This paper presents a new mathematical and simulative framework for quantifying the overhead of reactive routing protocols, such as DSR and AODV, in wireless variable topology (ad-hoc) networks. A model of the routing-layer traffic, in terms of the statistical description of the distance between a source and a destination, is presented. The model is used to study the effect of the traffic on the routing overhead. Two network models are analyzed; a *Manhattan grid model* for the case of regular node placement, and a *Poisson model* for the case of random node placement. We focus on situations where the nodes are stationary but unreliable. For each network model, expressions of various components of the routing overhead are derived as a function of the traffic pattern. Results are compared against *ns-2* simulations, which corroborate the essential characteristics of the analytical results. One of the key insights that can be drawn from the mathematical results of this paper is that it is possible to design *infinitely scalable* reactive routing protocols for variable topology networks by judicious engineering of the traffic patterns to satisfy the conditions presented in this paper.

Index Terms

Performance evaluation, modelling, ad hoc networks, reactive routing, routing overhead, AODV, DSR, scalability, simulation

I. INTRODUCTION

Several classes of routing protocols for variable topology multi-hop wireless (ad-hoc) networks have been proposed in the literature. In this paper, we consider the class of *reactive* routing protocols such as AODV [1] and DSR [2], where paths are maintained only when needed. We focus on situations where the topology changes because of node failure. Our objective is to mathematically characterize the various components of

Correspondence Author: Alhussein A. Abouzeid. 110 Eighth St, JEC-6038, Troy NY 12180 Tel: (518)276-6534, Fax: (518)276-4403.

routing overhead under different traffic patterns. We study the scalability of reactive routing protocols in terms of *routing overhead*.

The term “traffic” in this paper refers to routing-layer traffic, which is the pattern of route (i.e. path) requests. The focus of this paper is the inter-dependence between the traffic pattern and the routing overhead, by deriving quantitative measures relating the two. To our knowledge, there exist no evaluations (whether analytical or simulative) of this issue in the current literature. Related work¹ on the mathematical analysis of reactive routing protocols overhead include [3] and [4]. [3] provides an asymptotic (approximate) analysis that does not model the effect of the route-request traffic pattern while [4] provides an upper bound on the overhead of a hierarchical routing protocol.

The results obtained in this paper may have several practical applications. One such application is in the problem of sink-placement i.e. the problem of placing special nodes in the sensor grid for collecting information from “near-by” sensors, such that the average routing overhead is bounded by a desired value. An application for the case of hierarchical routing protocol design could be the selection of the size of the cluster such that the average routing overhead is bounded by some given constant.

Two models of the network topology are considered; regular (Manhattan grid) and random (Poisson field). We view the two different models as abstractions of two possible practical scenarios where we can (or cannot) control or assign the location and wireless coverage of nodes. While the analytical results obtained from the two models are different, the scalability properties of these two models are exactly the same.

It is important to note that, unlike some previous work (e.g. [5]), this work does not attempt to model or compare between specific reactive routing protocols - rather, to capture the essential behavior and scalability limits of this class of protocols by deriving lower bounds on the overhead. The simulation results in this paper are not intended to provide an exact match with our analysis - rather, they are provided as a support of the conclusions regarding overhead scalability. The *ns-2* [6] simulations naturally deviate from the assumptions made in the analysis, and are intended to reflect that the results are indeed reasonable, even though we do not model the complex behaviors of MAC layer or the detailed route caching mechanisms.

This paper is organized as follows. Section II presents the topology, traffic and routing protocol models. Sections III and IV present the results (analysis and simulations) of routing overhead for regular and random network topology, respectively. Section V concludes the paper by highlighting the practical implications of this work and several possible avenues for future extensions.

¹Part of the material in this paper (Section III) was presented at the Ninth Annual International Conference on Mobile Computing and Networking (MobiCom 2003), San Diego, CA, September 2003, and is included in revised form in this paper.

II. MODELLING

A. Topology Model

1) *Regular Topology Model*: In this case, the network is modelled by a regular degree-4 (Manhattan) grid (Figure 1) where nodes are located at the intersections with unit physical distance between neighbors. Two nodes within the range of each other can communicate directly in either direction and are said to be neighbors.

Let (x_i, y_i) denote the coordinates of a node i , where $-\infty < x_i, y_i < +\infty$ are integers. We define the distance $r_{i,j}$ between two nodes i and j as $r_{i,j} = |x_i - x_j| + |y_i - y_j|$. The distance between two nodes is exactly the number of ‘‘hops’’ along the shortest path between them.

Finally, we will refer frequently to the following summation:²

$$f(k) = \sum_{r=1}^{\infty} \frac{1}{r^k} \quad (1)$$

2) *Random Topology Model*: In this case, the nodes are randomly distributed over an unbounded area following a homogeneous Poisson point distribution with node density λ [7]. Therefore the random network is spatially homogenous (Fig. 2). The probability distribution of the number of nodes N located within an area A is

$$p_N(n) = \frac{(\lambda A)^n}{n!} e^{-\lambda A} \quad (2)$$

B. Traffic Model

We define a *new path request* or simply a *new session* as one that is associated by the arrival of a new routing-level session request at a node i with some destination $j \neq i$ for which node i doesn’t already have a path. If the destination is the same as an existing session, a ‘‘new path request’’ will not be generated, and hence we do not count this as a *new session*. New path requests are independent. Throughout the paper, we refer to a *routing-layer sessions* simply by *sessions* or *active paths* interchangeably. Also, we interchangeably use a routing *packet* and routing *message*. Two nodes are said to *communicate* (possibly through multiple hops) if they are the source/destination pair of a session.

1) *Regular Topology*: Let $r_{i,j}$ denote the distance between two nodes i and j . We assume that a node i will need to communicate and hence maintain a (routing-layer) session with node j with probability $p_{i,j}$ given by

$$p_{i,j} = \frac{c}{r_{i,j}^k} \quad (3)$$

where $0 < c < 1$. Because of the symmetry, it will be more convenient to drop the subscripts from the above equation when it is understood. We thus let $p(r)$ denote the probability that two nodes (i and j) will communicate i.e.

$$p(r) = \frac{c}{r^k} \quad (4)$$

²This infinite sum is a famous series known as R-series. The sum of the infinite series has general closed form result when k is an even integer.

where r is the distance between the node initiating the new path request and the destination node.

2) *Random Topology*: In the discussion of random topology, “distance” refers to the “physical distance” and not the number of hops as in Manhattan grid. The reason of choosing geographic distance rather than the number of hops between two nodes is that the number of hops along the shortest path between two nodes is affected by both the geographic distance between the nodes and the random placement of neighboring nodes.

Let (x_i, y_i) denote the physical coordinates of a node i . Let $r_{i,j} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$ denote the physical (geographic) distance between two nodes i and j .

$$p(r) = \begin{cases} \frac{c}{r^k} & ; \text{ for } r \geq r_1 \\ \frac{c}{r_1^k} & ; \text{ for } 0 \leq r < r_1 \end{cases} \quad (5)$$

where $r_0 \leq r_1$ and $0 \leq c \leq r_1^k$. Note that we have omitted the subscripts i, j for convenience.

We assume that the nodes along an active session are roughly distributed along the straight line connecting the source and the destination nodes of the session. Let h denote the number of hops along the shortest path between the source and destination and let r denote the physical length of the straight line from the source to destination. We use following equation to estimate h

$$h \approx \left\lceil \beta \frac{r}{r_0} \right\rceil \quad (6)$$

where β is a positive constant.

3) *A note on the choice of the traffic model*: A discussion of the choice of the shape of the above traffic distribution is necessary. Up to our knowledge, there are no empirically derived traffic statistics for wireless ad-hoc or sensor networks, and, in the lack of such statistics, the assumption is necessary for modeling purposes.

We deliberately did not choose the form $\frac{c}{k^r}$ although this latter form may have proved easier to handle mathematically. The reason is that $\frac{c}{r^k}$ allows us to analyze the effects of r and k more closely than $\frac{c}{k^r}$ since the former decays polynomially in r while the latter decays exponentially in r .

C. Routing Protocols Model

We describe below a generic reactive routing protocol, which we believe captures the essential behavior of many designs and implementations of routing protocols, including DSR and AODV, relevant to our analysis. We do not describe the caching mechanisms here. The effects of caching mechanisms will be observed later in the paper through simulations. The overhead of reactive routing protocols can be associated with two operations: *route discovery* and *route maintenance*, both of which are described below.

1) *Route Discovery*: Route discovery is the mechanism initiated by a node i upon the arrival of a “new path request” (defined earlier) in order to discover a new path to a node j . Similar to other reactive routing protocols, our generic protocol uses a “flooding” technique for control packets. Node i floods the network with Route Request (RREQ) packets. The header of the RREQ packet is initialized by some value called Time to Live (TTL) in the header of the packet. Each node forwards an RREQ packet only once, and decrements the

TTL upon each transmission. If the initial TTL value is large enough, an RREQ packet arrives to j (we assume the network is connected). Node j (or an intermediate node that has the route cached) sends out a Route Reply packet (RREP) to i along the reverse path. Finally node i gets a shortest path source route to node j (in the case of source routing protocols like DSR) or entries in the routing tables are established at the nodes along a path between i and j (in the case of distance vector protocols like AODV).

2) *Route Maintenance*: Route maintenance is the mechanism by which a node i is notified that a link along an active path has broken such that it can no longer reach the destination node j through that route. Upon reception of a notification of route failure, node i can initiate route discovery again to find a new route for the remaining packets destined to j . For example, in Figure 3, there are two source routes. One from node E to node A, another from node D to node A. Suppose now node B fails, if the intermediate node C finds the link to its next-hop destination B broken (by some MAC layer method), C initiates a Route Error (RERR) message. We describe two route notification mechanisms, one mimics source-route routing while the other mimics distance vector routing:

(a) In *source-route notification*, C sends an RERR message to *each source node* that has sent a packet routed through the failed link. Each RERR message will travel along *the reverse route* from the node reporting link breakage to the source node. Thus, if any link on a source route is broken, the original source node is notified by an RERR packet. For the example, node C sends two source RERR packets since two routes pass through the link C-B. One RERR packet is from node C to node D, another RERR packet is from node C through D then to node E. In this scenario, node D receives an RERR packet twice, although each RERR reports the same broken link. After receiving the RERR, a new route discovery process must be initiated by the source, if this route is still needed.

(b) In *distance-vector notification*, node C also sends an RERR message. However, in distance-vector routing, node C doesn't know the source of the packets. Thus the RERR will be forward to the active "next hop" entries in the distance vector routing table. The difference between this and DSR can be indicated in reference to Figure 3, where in this case, node D receives RERR packet only once. In essence, RERR packets follow a spanning tree rooted at the node detecting the failure.

D. Notations and Definitions

This section presents several definition and notations that will be used in the rest of the paper.

- $N(r)$ denotes the number of nodes located within a distance r from a given node.
- R denotes the distance between a pair of source and destination nodes of an active session.
- $p_R(r)$ denotes the probability that a pair of source and destination nodes is a distance r apart.
- Z denotes the number of hops that an active session travels i.e. *session length*. $E[Z]$ denotes the expected value of Z .

- For $0 < \alpha < 1$, $T(\alpha)$ denotes the TTL value necessary for guaranteeing that the packet will reach the destination with a probability at least $(1 - \alpha)$.
- $E[S]$ denotes the average number of active paths terminating at a given node.
- $E[S_i]$ denotes the average number of (routing-layer) sessions (i.e. paths) initiating from a given node.
- $E[S_r]$ denotes the average number of sessions passing through a node. Here a pass-through session (relay session) is one that neither terminates nor starts at the given node.
- $E[S_a]$ denotes the average number of active paths at a given node (i.e. sum of all the above three quantities).
- q denotes the probability that a routing packet at a given node is not terminating at the node. In other words, the probability that the session does not terminate at the given node.
- N_{find} denotes the total number of RREQ packets per new route discovery. Every transmission of the same packet is counted as a separate transmission. $E[N_{find}]$ denotes the expected number.
- N_{off} denotes the total number of packets (cost) needed to notify others about a node failure. $E[N_{off}]$ denotes the expected number.
- Let M denote the set of all the nodes in the network, and let $M \setminus v$ denote all the nodes except node v .

For the case of random network topology, the following notations will also be needed:

- Let g denote the average number of nodes within direct communication range of a given node.
- Let r_0 denote the communication radius of a node, which is same for all the nodes in the network.

And finally, we introduce the following important definition:

Definition: A certain metric is *infinitely scalable* if and only if it grows as $O(1)$ as the network size grows to infinity.

III. ANALYSIS FOR REGULAR NETWORK

A. Some Quantities of Interest

Lemma 3.1: The number of nodes located at a distance r away from a given node is $D(r) = 4r$.

Proof: Without loss of generality, choose node o as the node of interest. For any given positive integer r , there are a total of $(r + 1)$ nodes at a distance r from node o located in each of the four quadrants (Figure 1). Summing over the four quadrants yields $4(r + 1)$. Removing the four double counted nodes located at x and y axis, the total number of nodes is $4r$. ■

It follows from the above result that $N(r) = 2(r^2 + r) + 1$.

Theorem 3.2: For $k > 2$,

$$p_R(r) = \frac{1}{r^{k-1}} = \frac{1}{f(k-1)}; k > 2 \quad (7)$$

For $k \leq 2$, $p_R(r) = 0$.

Proof: Without loss of generality, choose node o as the source node. Let r denote the distance from the source node to destination node. There are $D(r)$ nodes at a distance r away from node o . Each node is communicating with node o with a probability $p(r)$. Hence,

$$p_R(r) \propto (4r)p(r) = \frac{4c}{r^{(k-1)}} \quad (8)$$

By normalizing the above equation, we prove the theorem. The probability distribution is meaningful only if we have $k > 2$, otherwise it becomes zero. ■

B. Average Number of Sessions

Corollary 3.3: The average number of hops that an active session travels is equal to

$$E[Z] = \frac{f(k-2)}{f(k-1)}; k > 3 \quad (9)$$

Proof:

$$E[Z] = \sum_{z=1}^{\infty} z p_Z(z) = \sum_{z=1}^{\infty} z \frac{\frac{1}{z^{k-1}}}{f(k-1)} = \frac{\sum_{z=1}^{\infty} \frac{1}{z^{k-2}}}{f(k-1)} = \frac{f(k-2)}{f(k-1)} \quad (10)$$

■

Please refer to Figure 4.

Theorem 3.4: The average number of active paths terminating at a node is

$$E[S] = 4c * f(k-1); k > 2 \quad (11)$$

Proof: Without loss of generality, choose node o as the destination node. Let $X_{i,o}$ be an indicator function which is 1 if there exists an active path starting at node i and terminating at node o , and is 0 otherwise. Let $p_{i,o}$ denote the probability that node o and i have an active session. Clearly, $Pr[X_{i,o} = 1] = p_{i,o}$. Hence, $S = \sum_{i \in M \setminus o} X_{i,o}$. Taking the expectation of both sides and using the result of Lemma 3.1,

$$E[S] = \sum_{r=1}^{\infty} 4r \frac{c}{r^k} = 4c \sum_{r=1}^{\infty} \frac{1}{r^{(k-1)}} = 4c * f(k-1); k > 2 \quad (12)$$

where $k > 2$ is necessary for convergence of the equation. ■

From symmetry,

Corollary 3.5:

$$E[S_i] = E[S] = 4c * f(k-1); k > 2 \quad (13)$$

Lemma 3.6:

$$E[S_r] = 4c * (f(k-2) - f(k-1)); k > 3 \quad (14)$$

$$E[S_a] = 4c * (f(k-2) + f(k-1)); k > 3 \quad (15)$$

Proof: Without loss of generality, we choose node o as the source node, and let r denote the distance from the source node o to destination node of a session initiated by node o . This session will pass through $(r - 1)$ intermediate nodes. There are total $4r$ nodes r away from node o . The average number of active pass-through sessions observed by other nodes is thus

$$\sum_{r=1}^{\infty} (r - 1) * 4r * p(r) = \sum_{r=1}^{\infty} (r - 1) * 4r * \frac{c}{r^k} = 4c * \sum_{r=1}^{\infty} \left(\frac{1}{r^{k-2}} - \frac{1}{r^{k-1}} \right) = 4c * (f(k - 2) - f(k - 1)) \quad (16)$$

From symmetry, the average number of active relay sessions for a given node o has to be equal to the average number of active pass-through sessions observed by other nodes but initiated by node o . The right side of the above equation gives the average number of relay sessions for a node. Finally, the average of the total number of sessions at a node is the sum of the averages of pass-through sessions, initiated sessions and terminated sessions at the node. ■

The above results indicate that, indeed, the number of sessions passing through a node is larger than the number of sessions generated, on the average. The reason is that some sessions are observed by many intermediate nodes along the communication path.

Theorem 3.7: For $k > 3$,

$$q = 1 - \frac{2f(k - 1)}{f(k - 2) + f(k - 1)} \quad (17)$$

For $k \leq 3$, $q = 1$.

Proof: The result follows directly from Lemma 3.6,

$$q = \frac{E[S_r]}{E[S_a]} = \frac{f(k - 2) - f(k - 1)}{f(k - 2) + f(k - 1)} = 1 - \frac{2f(k - 1)}{f(k - 2) + f(k - 1)} \quad (18)$$

C. Route Discovery Overhead

Claim 3.1: For $0 < \alpha < 1$,

$$T(\alpha) = \min\left\{i \mid \sum_{r=i+1}^{\infty} \frac{1}{r^{k-1}} \leq \alpha f(k - 1)\right\}; k > 2 \quad (19)$$

Proof: From Theorem 3.2, the probability that the destination node is located at a distance larger than a positive integer i can be calculated from (7) as

$$\sum_{r=i+1}^{\infty} p_R(r) = \frac{\sum_{r=i+1}^{\infty} \frac{1}{r^{k-1}}}{f(k - 1)} \quad (20)$$

which proves the claim. ■

Corollary 3.8: The total number of route request (RREQ) packets associated with a flooding event with TTL value $T(\alpha)$ is

$$N_{find} = 1 + 2 * T(\alpha) * (T(\alpha) - 1); k > 2 \quad (21)$$

Proof: Assume that the route discovery is initiated by a node i . The longest distance each packet travels is bounded by $T(\alpha)$. Let $r_{i,j}$ denote the distance between a node j and node i . Node j with distance $r_{i,j} < T(\alpha)$ will transmit the packet once. Therefore, the total packet retransmissions is the same as the number of the nodes within a circle of radius $T(\alpha)$ from node i . From Lemma (3.1),

$$N_{find} = 1 + \sum_{l=1}^{T(\alpha)-1} 4 * l = 1 + 4 * \frac{T(\alpha)(T(\alpha) - 1)}{2} = 1 + 2 * T(\alpha) * (T(\alpha) - 1) \quad (22)$$

■

D. Shortest Path Analysis

We will need to derive the following two lemmas which provide expressions for some interesting quantities regarding the properties of the shortest paths. Those will be used in the following section to derive the minimum average route failure notification overhead.

Let (x_i, y_i) denote the coordinates of a node i . Define $\Delta x_{i,j} \triangleq |x_i - x_j|$ and $\Delta y_{i,j} \triangleq |y_i - y_j|$ then $r_{i,j} = \Delta x_{i,j} + \Delta y_{i,j}$.

Lemma 3.9: Let $L_{i,j}$ denote the total number of shortest paths between two nodes i and j . Consider the case where $x_i < x_j$ and $y_i < y_j$ (Figure 5). The shortest path between the two nodes will consist of exactly $\Delta x_{i,j}$ horizontal hops to the right and $\Delta y_{i,j}$ upward vertical hops on the grid, in any order. Thus, $L_{i,j}$ is the total number of permutations of hops of the two types. It follows from standard counting methods that

$$L_{i,j} = \binom{r_{i,j}}{\Delta x_{i,j}} = \frac{(r_{i,j})!}{(\Delta x_{i,j})!(\Delta y_{i,j})!} \quad (23)$$

Lemma 3.10: Let $L_{i,j,l}$ denote the number of shortest paths between nodes i and j that pass through node l . For any two arbitrary nodes i and j with $x_i < x_j$ and $y_i < y_j$, we need only to consider the nodes that satisfy $x_i \leq x_l \leq x_j$ and $y_i \leq y_l \leq y_j$ since any other node l that does not satisfy this condition cannot possibly lie on any of the shortest paths between nodes i and j . Using the same counting argument as before,

$$L_{i,j,l} = L_{i,l} L_{l,j} \quad (24)$$

Let $a_{i,j,l}$ denote the probability that node l lies along the shortest route between nodes i and j , as shown in Figure 5. Assume that when nodes i and j need to communicate, any of the shortest paths between these two nodes is discovered (through a RREQ/RREP procedure) with equal ³ probability. Then

$$a_{i,j,l} = \frac{L_{i,j,l}}{L_{i,j}} \quad (25)$$

An example is shown in Figure 5.

³This is a reasonable assumption for flooding-based routing protocols.

Let $p_{i,j}$ denote the probability that there is an active session started from node i and terminated at j . Let $b_{i,j,l}$ denote the probability that node l is routing the communication session from node i to node j . Then

$$b_{i,j,l} = a_{i,j,l} p_{i,j} = p_{i,j} \frac{L_{i,j,l}}{L_{i,j}} \quad (26)$$

E. Route Notification Overhead

In this section, we will compute the minimum overhead required to complete a route notification procedure, on the average. The best reactive routing protocol can do no better (in terms of routing overhead) than a hypothetical routing protocol that notifies the intended nodes by communicating over the shortest possible paths from the failed node.

We presented in Section II-C two methods for route notification; source route-based and distance vector-based. In the former, separate messages are sent for each intended node, while in the latter, the same message may be sent along the shared portion of overlapping routes to more than one destination. Another important feature of distance vector route maintenance is the local-repair feature (e.g. see [1]). If an intermediate node fails, neighboring nodes may attempt to repair (or patch) the path instead of notifying the transmitting. Clearly, this works only for the paths that are not terminating at the intended node.

Thus, we split our analysis into three cases; two extreme cases, and an intermediate case, as follows. The first case uses local repair together with distance vector-based notification. We make the optimistic assumption that local repair will succeed for all active paths except for those terminating at the failed node. The third case is for source route based notification where there is no local repair and all affected nodes (whether having active pass-through paths through the failed node or active paths terminating at the failed node) must be notified, each with an independent RERR packet. The second, intermediate, case uses local repair but with source route notification (i.e. separate messages). We will show that in some cases, the first and second cases are equivalent.

1) *Case 1: Local Repair with Distance-vector based notifications:* Without loss of generality, choose node o as the destination node. Node j is another node in $M \setminus o$. Let $c_{j,o}$ denote the probability that there is an active session with node o started from j and at the same time node j is not a routing node for other active sessions having node o as destination node. Let M_j denote the set of nodes that can have node j as its routing node. From any node $k \in M_j$, there is a path passing through node j that is one of the possible shortest paths from node k to node o . For example, for a node j located at $(2,3)$, M_j will be all the nodes located at $\{(x,y) | x \geq 2, y \geq 3\}$ except $(2,3)$. Note that the event “there is an active session with node o started from j ” is independent from the event “node j is not a routing node for other active sessions having node o as destination node.”

$$c_{j,o} = p_{j,o} \left[\prod_{l \in M_j} (1 - b_{l,o,j}) \right] = p_{j,o} \left[\prod_{l \in M_j} \left(1 - p_{l,o} \frac{L_{l,o,j}}{L_{l,o}} \right) \right] \quad (27)$$

Theorem 3.11:

$$E(N_{off}) = \sum_{j \in M \setminus o} (r_{j,o} - 1) c_{j,o} \quad (28)$$

Proof: Define a random variable $X_{j,o}$ for a given node j as:

$$X_{j,o} = \begin{cases} r_{j,o} - 1 & ; A \\ 0 & ; \text{otherwise} \end{cases} \quad (29)$$

where A is the event “session starts at j and ends at o , and also j is not a routing node for other active sessions that end at o .” Hence,

$$N_{off} = \sum_{j \in M \setminus o} X_{j,o} \quad (30)$$

and taking the expectation of both sides and substituting by $E[X_{j,o}] = (r_{j,o} - 1)c_{j,o}$ completes the proof. ■

2) *Case 2: Local Repair with Source Route Based Notification:* In this case, a RERR packet will be sent to the source of each active session that ends at the failed node, independently (i.e. even if some of the sessions might have some portion of the routes that are common). Using the same methodology presented in Theorem 3.4, we have

Theorem 3.12:

$$E[N_{off}] = 4c(f(k-2) - f(k-1)) \quad (31)$$

Proof: Let i denote any node in the network, and let r denote the distance from node i to destination node of a session initiated by node i . If node i fails, the RERR packet for this session will pass $(r-1)$ intermediate nodes. There are total $4r$ nodes having distance r away from node i . The average number of messages is

$$\sum_{r=1}^{\infty} (r-1) * 4r * p(r) = \sum_{r=1}^{\infty} (r-1) * 4r * \frac{c}{r^k} = 4c * \sum_{r=1}^{\infty} \left(\frac{1}{r^{k-2}} - \frac{1}{r^{k-1}} \right) = 4c * (f(k-2) - f(k-1)) \quad (32)$$

3) *Relation between route notification in Cases 1 and 2:* Consider a special case where a network in which the needs of communication are “weak” i.e the coefficient c in (4) is very small. If we only preserve the first-order involving c , then

$$c_{i,j} = p_{i,j} = \frac{c}{r_{i,j}^k}; k > 3 \quad (33)$$

and hence

Corollary 3.13: For a network which is low traffic, i.e $c \ll 1$, route failure notification overhead for both cases 1 and 2 above are identical. Specifically, (28) reduces to (31)

Proof: Without loss of generality, we assume a node o is turned off (or failed) at location at $(0, 0)$. Because the communication needs are weak, we can ignore the terms of c^n in (28) with $n > 1$. Hence,

$$E[N_{off}] = \sum_{r=1}^{\infty} (r-1) * 4r * \frac{c}{r^k} = 4c(f(k-2) - f(k-1))$$

■

4) *Case 3: No local repair and source route based notification:* In this case, an RREP packet will be sent to each source that has an active session passing through or terminating at the failed node.

Theorem 3.14:

$$E[N_{off}] = 2c(f(k-3) - f(k-2)); k > 4 \quad (34)$$

Proof: Observed from any given node m , whether a session passes through or terminates at the node m can be uniquely presented as a tuple (j, m, i) . Here, j is the session source node, i is the session destination node. If a node m is turned off, the total number of RERR packets that will be transmitted to node j to notify it about the failure of the node m is $(r_{m,j} - 1)$.

For any given node i and a given session that ends at node i with session length (distance from source node to destination node) r will generate $(r - 1)$ tuples described above. Each tuple specifies a pass-through or terminate node m for this session. From the symmetric construction of the network, for a given node m and a given distance r and a given positive integer $l < r$, the average number of sessions with session length r and distance l from pass-through node m to source node is $(4rc/r^k)$. If node m fails, the number of packets to notify node j is $(l - 1)$. Finally, we have

$$E[N_{off}] = \sum_{r=1}^{\infty} \left(\frac{4c}{r^{k-1}} \right) \left(\sum_{l=1}^r (l - 1) \right) = 2c(f(k-3) - f(k-2)) \quad (35)$$

■

F. Scalability of the Manhattan grid

From previous results, the average total number of active sessions for a node (15) and the average number of packets to notify about the failure of a node (34) are bounded for an infinite Manhattan grid if the coefficient k is larger than 3, and 4. We have

Corollary 3.15: The Manhattan grid becomes *infinitely scalable* only if the coefficient k is larger than 4. This result, among others, is validated in the following section.

G. Numerical Results and ns-2 Simulations

To verify our analysis, we use the ns-2 [6] simulation tool to run a number of simulations described in this section. Since it is impossible to simulate an infinite grid, an issue arises in how to compare the analytical results with simulations. As mentioned in Section II, our analysis applies to a finite but symmetric grid (i.e. a torus) by replacing the upper limit in each summation in the results by the proper value that depends on the size of the finite grid used. A square grid of size N was used in generating the simulation results. Hence, for generating the numerical results from the analysis, the upper limit used is \sqrt{N} instead of ∞ . However, this doesn't take care of the edge effects that will arise in the simulations, which causes deviations from the analytical results, especially for small k .

1) *Simulation Set-up:* We consider five networks of size 49, 121, 225, 361 and 529⁴. Every node is placed at the intersection of a square Manhattan grid. Here, nodes have fixed positions without any movement for the entire simulation. Field sizes are different for different network sizes; 1400m \times 1400m field has 49 nodes; 2200m \times 2200m for 121 nodes; 3000m \times 3000m for 225 nodes; 3800m \times 3800m for 361 nodes; 4600m \times 4600m for 529 nodes.

As the goal of our simulation is to verify our analytical results, the transport protocol is irrelevant, so we use the most simple one to speed up the simulation run time. We choose our traffic sources to be Constant Bit Rate (CBR) sources. A 512 byte data packet is used for all CBR sources. The inter-arrival time of CBR sources is fixed at 4.5 seconds. The CBR agent will be attached to a UDP agent, which is in turn attached to the source node. The source-destination pairs are generated according to (4). Each data point represents an average of at least five runs with identical topology, but different traffic patterns (source-destination pairs).

2) *Numerical and Simulation Results:* Figure 6 shows the results of Theorem 3.4 and generated traffic pattern. The average number of the communicating sessions terminated/initiated at a node is counted. The analysis results are calculated as (11)⁵. It seems the session number per node in simulation increases not as rapidly as theoretical results. This is due to edge effects.

The second experiment verifies the results of Theorem 3.7 in Figure 7. The theory results are according to (17). We use AODV as the routing protocol in the simulation for Figure 7. We counted the total number of sessions initiated or passing through a node. Then the average percentage of forwarding sessions at a node is counted. Figure 8 shows the average total number of sessions from AODV simulations and analysis. The numerical results are calculated according to (15).

Figures 9, 10 and 11 verify Claim 3.1. The numerical results are calculated as $(1 - \alpha)$ with different TTL values in (19). In the simulated scenarios, we set up communicating sessions initiated from the center node. AODV is used as the routing protocol in the simulation about TTL. For every session, if the source node cannot find a route to destination due to TTL limit, the route request is counted as unsuccessful. We count the total number of unsuccessful route requests. From Figure 9, 10 and 11, we observe that as the network size N increases, simulation results and theoretical results provide an increasingly better match.

As mentioned earlier, when a node is turned off, route error packets need to be sent to the source nodes that have active sessions terminating at the failed node. Both AODV and DSR are used for simulations. Since default options of DSR in *ns-2* turn on route cache that is not considered in our theoretical analysis, it is interesting to know what is the effect of route caching on scalability of routing protocols. Simulation results show that route caching doesn't have significant effect on the scalability results as we can see from an example simulation result of DSR with and without route caching in Figure 12. But to be consistent with our theoretical

⁴Slow simulation speed and large memory requirement for the *ns-2* models prevented us from using larger network at this stage.

⁵As pointed out earlier, in all the following numerical analysis results, we summed r from 1 to the network diameter \sqrt{N} instead of to infinity in calculating $f(k)$ since our simulations are for finite networks.

analysis which doesn't consider route cache, "Replying to Route Requests using Cached Routes" and "Packet Salvaging" in DSR [2] are turned off in all other simulations.

Figure 13 verifies Theorem 3.11. In the simulated scenarios, we set up communicating sessions that all the destinations are the same, which is the center node of the network. After every session has run for a long time (enough for every source node to find a route to destination) to avoid transient effects, the destination node is turned off. Then we count every route error packet sent from neighboring nodes. The theoretical result is given by (31).

Theorem 3.14 derives the average number of packets to notify about the failure of a node to the nodes that have active sessions ending at or passing through the failed node. This time, only DSR is selected as routing protocol for Figure 14 as we mentioned in Section II-C.2. Theoretical result is obtained from (34), which states that the route failure notification overhead infinitely scale only if $k > 4$.

IV. ANALYSIS FOR RANDOM NETWORK

In this section, we present the analysis of the case of random network topology. The derivation steps are similar to the case of regular network topology. However, due to the random spatial distribution of nodes, the analysis here uses continuous rather than discrete mathematics.

A. Some Quantities of Interest

Lemma 4.1: Let o denote any node of the network. A is the area of the circle with node o as the center and with r as the radius. The average number of nodes located inside of the area A is $N(r) = \pi\lambda r^2$

Proof: The size of the area A is πr^2 . Hence,

$$\sum_{n=0}^{\infty} n p_N(n) = \sum_{n=0}^{\infty} n \frac{(\lambda A)^n}{n!} e^{-\lambda A} = \lambda A e^{-\lambda A} \sum_{n=0}^{\infty} \frac{(\lambda A)^n}{n!} = \lambda A = \pi\lambda r^2 \quad (36)$$

Taking the differentiation of the above equation, the average number of nodes located within a 'ring' from r to $r + dr$ from the given node is $D(r)dr = 2\pi\lambda r dr$. From Lemma 4.1,

$$g = \lambda\pi r_0^2 \quad (37)$$

Theorem 4.2: For $k > 2$,

$$p_R(r) = \begin{cases} \frac{2(k-2)}{k} \frac{r_1^{k-2}}{r^{k-1}} & \text{for } r \geq r_1 \\ \frac{2(k-2)}{k} \frac{r}{r_1^2} & \text{for } 0 \leq r < r_1 \end{cases} \quad (38)$$

For $k \leq 2$, $p_R(r) = 0$.

Proof: $p_R(r)dr$ is the probability that the distance between a source/destination pair is within the range of $[r, r + dr]$. From Baye's rule of probability theory, $p_R(r)dr \propto p(r)D(r)dr$. Normalizing this relation proves the theorem. ■

Taking the expectation of R ,

$$E[R] = \frac{2(k-2)}{3(k-3)}r_1; k > 3 \quad (39)$$

B. Average Number of Sessions for Random Network

Corollary 4.3: The average number of hops that an active session travels is

$$E[Z] \approx \left\lceil \beta \frac{2(k-2)r_1}{3(k-3)r_0} \right\rceil; k > 3 \quad (40)$$

Proof: From (6),

$$E[Z] \approx \left\lceil \beta \frac{E[R]}{r_0} \right\rceil = \left\lceil \beta \frac{2(k-2)r_1}{3(k-3)r_0} \right\rceil; k > 3 \quad (41)$$

■

Claim 4.1:

$$E[S] = \frac{k\pi\lambda c}{(k-2)r_1^{k-2}}; k > 2 \quad (42)$$

Proof:

$$E[S] = \int_0^\infty D(r)p(r)dr = 2\pi\lambda \left[\int_0^{r_1} \frac{c}{r_1^k} r dr + \int_{r_1}^\infty \frac{c}{r^k} r dr \right] = \frac{k\pi\lambda c}{(k-2)r_1^{k-2}} \quad (43)$$

As k approaches to 2, the average number of sessions will approach infinity. Also, $E[S]$ is proportional the value of λ . ■

Corollary 4.4: The average number of active sessions started at a node $E[S_i]$ is equal to the average number of active sessions terminated at a node $E[S]$, and

$$E[S_i] = E[S] = \frac{k\pi\lambda c}{(k-2)r_1^{k-2}}; k > 2 \quad (44)$$

Proof: From symmetry, the two averages have to be equal. ■

Theorem 4.5:

$$E[S_r] \approx \frac{2\pi\lambda ck}{3(k-3)r_0 r_1^{k-3}} \beta - \frac{\pi\lambda ck}{(k-2)r_1^{k-2}}; k > 3 \quad (45)$$

Proof: Assign a unique label m for each node in the network, and a unique label x for each active path (session) in the network. For each node m acting as a relay for session x , create an ordered pair (m, x) . Let Y be the set of all ordered pairs. The cardinality of Y is the total number of relay events in the network.

One can create this set in two ways; either by inspecting each node (i.e. for each node, create an ordered pair for each session relayed by the node) or by inspecting each session (i.e. for each session, create an ordered pair for each intermediate node visited). Thus, the average number of relay events in the network is the same as the average number of intermediate nodes per session.

Every session that starts from a given node m and ends at a destination that is a distance r away will incur $\beta \left\lceil \frac{r}{r_0} \right\rceil - 1$ intermediate nodes. Hence,

$$\begin{aligned}
E[S_r] &\approx \int_0^\infty (\beta \frac{r}{r_0} - 1)p(r)D(r)dr = \frac{2\pi\lambda c\beta}{r_0} \left[\int_0^{r_1} \frac{1}{r_1^k} r^2 dr + \int_{r_1}^\infty \frac{1}{r_1^k} r^2 dr \right] - E[S] \\
&= \frac{2\pi\lambda c k}{3(k-3)r_0 r_1^{k-3}} \beta - \frac{\pi\lambda c k}{(k-2)r_1^{k-2}}; k > 3
\end{aligned} \tag{46}$$

■

Since the average of the total number of sessions for a given node is the sum of the average number of sessions relayed, initiating from, and terminating at the node,

Corollary 4.6:

$$E[S_a] \approx \left[\frac{2\pi\lambda c k}{3(k-3)r_0 r_1^{k-3}} \beta + \frac{\pi\lambda c k}{(k-2)r_1^{k-2}} \right]; k > 3 \tag{47}$$

Corollary 4.7:

$$q \approx \frac{2(k-2)\beta r_1}{2(k-2)\beta r_1 + 3(k-3)r_0}; k > 3 \tag{48}$$

Proof: For $k > 3$,

$$q \approx \frac{E[S_r]}{E[S_r] + E[S_i] + E[S]} = \frac{2(k-2)\beta r_1}{2(k-2)\beta r_1 + 3(k-3)r_0} \tag{49}$$

For $k \leq 3$, $q = 1$. ■

Note that q is not related to the node density λ , but related to the parameter k of the traffic model.

C. Route Discovery Overhead for Random Network

Claim 4.2: For a given small positive value $0 < \alpha < 1$,

$$T(\alpha) = \left[\beta \left(\frac{2}{k\alpha} \right)^{\frac{1}{k-2}} \frac{r_1}{r_0} \right]; k > 2 \tag{50}$$

Proof: From Theorem 4.2, the probability that the destination node is located more than r away can be calculated as

$$\int_r^\infty p_R(r)dr = \int_r^\infty \frac{2(k-2)}{k} \frac{r_1^{k-2}}{r^{k-1}} dr = \frac{2(k-2)}{k} \int_r^\infty \frac{r_1^{k-2}}{r^{k-1}} dr = \frac{2}{k} \frac{r_1^{k-2}}{r^{k-2}} \tag{51}$$

which yields

$$r = \left(\frac{2}{k\alpha} \right)^{\frac{1}{k-2}} r_1 \tag{52}$$

From (6),

$$T(\alpha) = \left[\beta \frac{r}{r_0} \right] = \left[\beta \left(\frac{2}{k\alpha} \right)^{\frac{1}{k-2}} \frac{r_1}{r_0} \right] \tag{53}$$

which proves the claim. ■

Note that node density λ does not appear in above equation, but it does not mean that the *TTL* is not related to the node density. The higher λ is, the higher β (and so the smaller $\alpha(T)$) will be.

Corollary 4.8: The number of route request messages that must be sent out during the flooding phase to guarantee that the probability of not finding the destination node is less than $0 < \alpha < 1$ is

$$N_{find} = \lambda\pi \left(\frac{2}{k\alpha}\right)^{\frac{2}{k-2}} r_1^2; k > 2 \quad (54)$$

Proof: From Claim 4.2,

$$r = \left(\frac{2}{k\alpha}\right)^{\frac{1}{k-2}} r_1 \quad (55)$$

Here, r is the radius of the flooding region. From (4.1), the average number of nodes located within the region is

$$N_{find} = \lambda(\pi r^2) = \lambda\pi \left(\frac{2}{k\alpha}\right)^{\frac{2}{k-2}} r_1^2 \quad (56)$$

■

D. Route Notification Overhead

Unlike the case of regular network, we will analyze only the case of source-route notification here.

Claim 4.3:

$$E[N_{off}] \approx \frac{2\pi\lambda ck}{3(k-3)r_0 r_1^{k-3}} \beta - \frac{\pi\lambda ck}{(k-2)r_1^{k-2}}; k > 3 \quad (57)$$

Proof: For any session ending at the failed node, the RERR packet will pass through $\beta \left\lceil \frac{r}{r_0} \right\rceil - 1$ nodes. Using the result from Lemma 4.1,

$$\begin{aligned} E[N_{off}] &\approx \int_0^\infty (\beta \frac{r}{r_0} - 1) p(r) D(r) dr \\ &= \frac{2\pi\lambda c\beta}{r_0} \left[\int_0^{r_1} \frac{1}{r_1^k} r^2 dr + \int_{r_1}^\infty \frac{1}{r_1^k} r^2 dr \right] - E[S] \\ &= \frac{2\pi\lambda ck}{3(k-3)r_0 r_1^{k-3}} \beta - \frac{\pi\lambda ck}{(k-2)r_1^{k-2}} \end{aligned} \quad (58)$$

This result is the same as the result for average number of relay sessions. The reason has been addressed in our proof of Theorem 4.5, where we proved that to count the relay sessions and to count the intermediate nodes is the same.

Theorem 4.9: The average number of RERR packets per node failure is

$$E[N_{off}] \approx \frac{\pi\lambda ck\beta}{r_0^2 r_1^{k-4}} \left[\frac{\beta}{4(k-4)} - \frac{r_0}{3(k-3)r_1} \right]; k > 4 \quad (59)$$

Proof: Assign a unique label m for each node in the network, and a unique label x for each active path (session) in the network. For each node m acting as a relay or a destination for session x , create an ordered tuple (m, x, w) where w denotes the distance between the node m and the source of session x . Notice that $w - 1$ is the number of RERR packets that must be transmitted in order to notify the source of session x that node m has failed. Let Y be the set of all $(m, x, w,)$ tuples. The average of the sum of the weights of the tuples is the average number of RERR packets transmitted.

Similar to the proof of Theorem 4.5, one can create the set Y and compute w either by iterating through the nodes, or alternatively, by iterating through the sessions. Using the latter approach, the average number of

RERR packets transmitted per node failure is equal to the average of the sum of weights from a given source to each possible intermediate or destination node.

A session x initiated from some node m and terminated at a destination a distance r (or $h = \beta \left\lceil \frac{r}{r_0} \right\rceil$ hops) away has $(h - 1)$ intermediate nodes and one destination node. Hence, $w = \sum_{l=1}^{h-1} l = \frac{1}{2}h(h - 1)$. Averaging over all the possible sessions for node m (and neglecting the ceiling operation),

$$E[N_{off}] \approx \int_0^\infty \frac{1}{2}h(h - 1)p(r)D(r)dr = \frac{\pi\lambda ck\beta}{r_0^2 r_1^{k-4}} \left[\frac{\beta}{4(k - 4)} - \frac{r_0}{3(k - 3)r_1} \right] \quad (60)$$

Note that the routing overhead will scale only if $k > 4$. ■

E. Scalability of Random Network

From the previous results, $E[S_a]$ (47) and $E[N_{off}]$ (59) are bounded if the coefficient k is larger than 3 and 4 respectively. We have the following result.

Corollary 4.10: The random network becomes *infinitely scalable* only if the coefficient k is larger than 4.

F. Numerical Results and ns-2 Simulations

1) *Simulation Set-up:* We repeat the same simulations as for the case of regular network. Only the different aspects of the simulations are described here. Unlike the case of Manhattan grid simulations, nodes here are *randomly* placed in a square area of size A . A random network topology example is shown in Figure 2. Let L denote the side length of the square, then $L = \sqrt{A}$.

From [8], [9], g should be at least $\Theta(\log n)$ to keep the network asymptotically connected. Considering the largest network size in our simulation, we choose $g = 10$ or $g = 20$. The default value for r_0 in ns-2 is 250 meters. The side length L of every network size is calculated from the relations $L = \lfloor \sqrt{N\pi/gr_0} \rfloor$. For every network size, a fully connected topology is generated (we generate a number of topologies, check the connectivity of each using standard methods, and select only those that are connected).

The source-destination pairs are chosen according to (5). For each fixed k and N , we have setup two sets of simulation scenarios with $g = 10$ and $g = 20$ respectively.

2) *Numerical and Simulation Results:* Fig. 15 shows that average number of sessions becomes flat for $k = 3$ or 4 as the number of nodes grow. For $k = 2$, the curve continues increasing as the network size increases. The scalability features for different k are consistent with the theoretical result. From Corollary 4.4, the average source/destination session becomes infinitely scalable only if $k > 2$. Fig. 15 also shows that the average source/destination sessions is roughly doubled for $g = 20$ compared with the averages for $g = 10$. The result is also consistent with the theoretical analysis. From Corollary 4.4 and (37), the average number of source/destination sessions is proportional to λ as well as g .

The second experiment shows the results of q in Fig. 16. We use AODV as the routing protocol. We counted the total number of sessions initiated or passing through a node. Then the average percentage of forwarding

sessions at a node is counted. From the experiment, the probability becomes constant for $k = 4$ when the network size is larger than 200 nodes. But for $k = 2$ and 3, the probability continue increasing as the network size increases. Besides, by fixing all other parameters, the change of the probability is not very sensitive to the change of g . For $k = 4$, the probability values are almost the same for $g = 10$ and $g = 20$.

The result of Corollary 4.7 is validated by this experiment. For $k > 3$, the probability is given by (48), which tells us that q is independent from the node density parameter λ . This is consistent with the experiment results (that the changes of probability is not sensitive to the changes of values of g).

From theoretical analysis, q is the probability that a session is a relay session, and is 1 if $k \leq 3$. From our experiment, the probability values of q for $k = 2$ and $k = 3$ continue increasing as the network size increasing, but not increasing fast enough as we expect. One explanation for the slow increase of q could be the edge-effects.

Figures 17, 18 and 19 study the effect of TTL value $T(\alpha)$ on the probability of finding destination node $(1 - \alpha)$. We observe that the smaller the value of k , the faster $(1 - \alpha)$ approaches to one. This phenomena is consistent with our theoretical result (53). The simulation results also show that the value of α is not sensitive to the parameter λ (the curves with same value of k but with different values of g almost overlap together), which is consistent with our theoretical result. In (53), the value of the α is not a function of parameter λ .

Figures 20 and Figure 21 show the simulation results for route error notification in the case of AODV and DSR. The simulated results of DSR with different traffic parameters are drawn in Fig. 22 and 23. When a node fails, the failure notification message is not shared for the DSR protocol. The simulation results setup using DSR protocol can be used to compare our theoretical result given by Theorem 4.9. According to the theorem, the average number of root failure packets scales only if $k > 4$. In our simulation scenario, we do see that the average packet number continues increasing as the size of the network increases for $k = 4$. But for $k = 5$, the average number of packets increases for small network size, and becomes a constant for large network sizes. We see that the average number of packets is almost doubled when g changes from 10 to 20. This is also consistent with Theorem 4.9. From the theorem, the average number of packets is proportional to parameter λ .

V. CONCLUSION

In this paper, we developed a mathematical analysis of the overhead of a broad class of reactive routing protocols. Specifically, we focused on routing overhead associated with (i) route discovery and (ii) route failure notification. The analysis was developed in the context of an unreliable network, modelled by (i) an unreliable Manhattan (i.e. degree 4) grid or (ii) a random Poisson point distribution of nodes each having equal coverage radius. Validity of the results was analyzed by comparing them against *ns-2* simulations of regular and random topologies.

Several issues were considered, such as the key differences between distance-vector and source-route based routing protocols regarding the mechanism of route failure notification, and the use of route repair.

Our analytical results point to the key role that the traffic pattern plays in defining the scalability of these protocols. Expressions for various quantities of interest, as well as conditions for scalability were derived and validated via *ns-2* simulations. Although we do not model many aspects of the network, including MAC layer details and route caching details, there is a reasonable match between the analytical results and simulations that do take into account those detailed aspects. Specifically, the simulations validate the infinite scalability results.

Several avenues of future work remain, including the analysis of the effect of the traffic pattern on the overhead of other types of routing protocols such as hierarchical routing and the application of the results in the design of ad hoc networks.

REFERENCES

- [1] C. E. Perkins, E. M. Belding-Royer, and S. R. Das, "Ad hoc on-demand distance vector (AODV) routing," Mobile Ad Hoc Networking Working Group, Internet Draft draft-ietf-manet-aodv-12.txt, November 4 2002.
- [2] D. B. Johnson, D. A. Maltz, Y.-C. Hu, and J. G. Jetcheva, "The dynamic source routing protocol for mobile ad hoc networks (DSR)," IETF MANET Working Group, INTERNET-DRAFT draft-ietf-manet-dsr-07.txt, February 21 2002.
- [3] C. Santivanez, B. McDonald, I. Stavrakakis, and R. Ramanathan, "On the scalability of ad hoc routing protocols," in *Proceedings of IEEE INFOCOM 2002*, 2002.
- [4] J. Sucec and I. Marsic, "Clustering overhead for hierarchical routing in mobile ad hoc networks," in *IEEE INFOCOM 2002*, 2002.
- [5] J. Broch, D. A. Maltz, D. B. Johnson, Y. C. Hu, and J. Jetcheva, "A performance comparison of multi-hop wireless ad-hoc network routing protocols," in *Proceedings of ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM'98)*, Dallas, Texas, USA, October 1998, pp. 85–97.
- [6] "ns-Network Simulator," 1995, <http://www.isi.edu/nsnam/ns/>.
- [7] H. Taylor and S. Karlin, *An Introduction to Stochastic Modeling*, 3rd ed. Academic Press, 1998.
- [8] P. Gupta and P. R. Kumar, "Critical power for asymptotic connectivity in wireless networks," in *Stochastic Analysis, Control, Optimization and Applications: A Volume in Honor of W.H. Fleming*, W. M. McEneaney, G. Yin, Q. Zhang, and Birkhäuser, Eds., Boston, 1998.
- [9] F. Xue and P. R. Kumar, "The number of neighbors needed for connectivity of wireless networks," to appear in *Wireless Networks*.

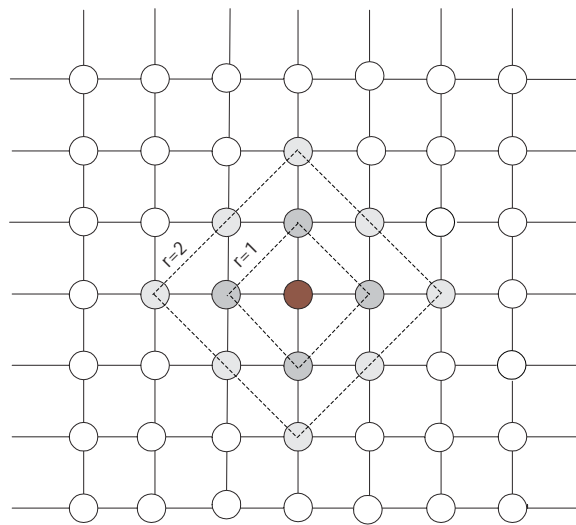


Fig. 1. A portion of a Manhattan grid around a given node (the darkest node in the center of the grid). Nodes linked together by the dashed line have the same distance to the node of interest.

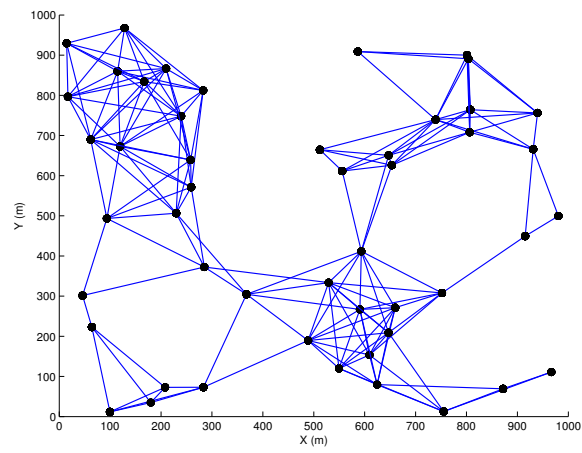


Fig. 2. An example of a random network topology with $N = 49$, $g = 10$ and $r_0 = 250m$. The dark spots represent the nodes.

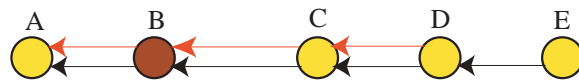


Fig. 3. Node failure example. Two sessions (paths) exist, one from D to A and another from E to A. Node C detects that node B has failed and hence notifies D and E by sending RERR packets.

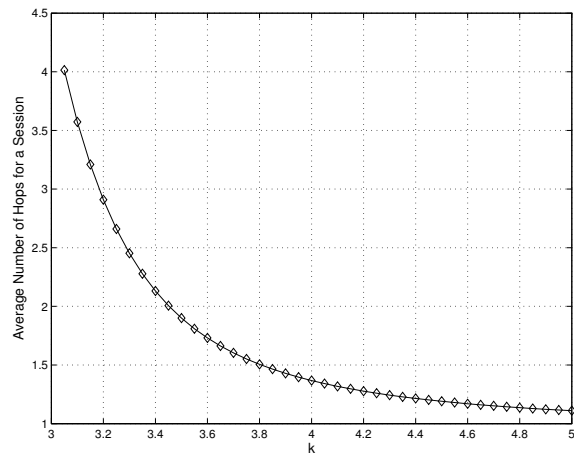


Fig. 4. Average number of hops that an active session travels.

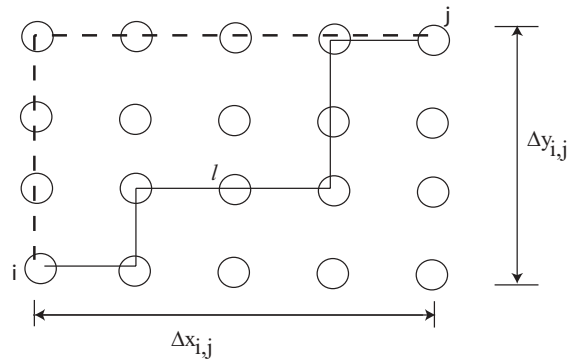


Fig. 5. A portion of the Manhattan grid. Two possible shortest paths between nodes i and j are shown, where the dashed path does not go through node l . For this example, $r_{i,j} = 7$, $\Delta x_{i,j} = 4$, $\Delta x_{i,l} = 2$, $\Delta x_{l,j} = 2$. By substituting in (23) and (24), $L_{i,j} = 35$ and $L_{i,j,l} = 18$, which can be verified by enumeration. Hence from (25), $a_{i,j,l} = 0.51$.

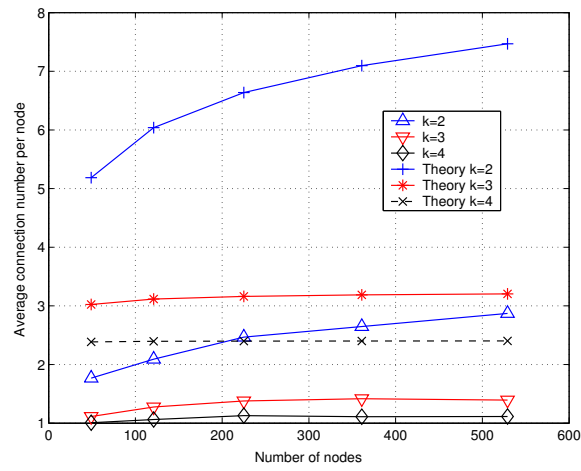


Fig. 6. Average number of source-destination sessions per node $E[S]$ (11), $c = 0.5$. $k > 2$ for convergence.

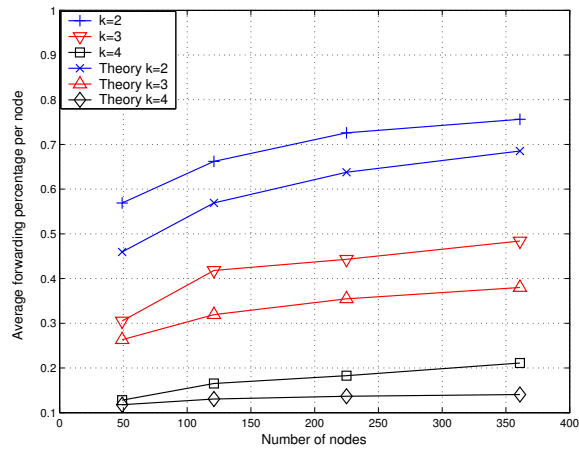


Fig. 7. Average forwarding percentage per node q (17), $c = 0.8$, AODV simulations.

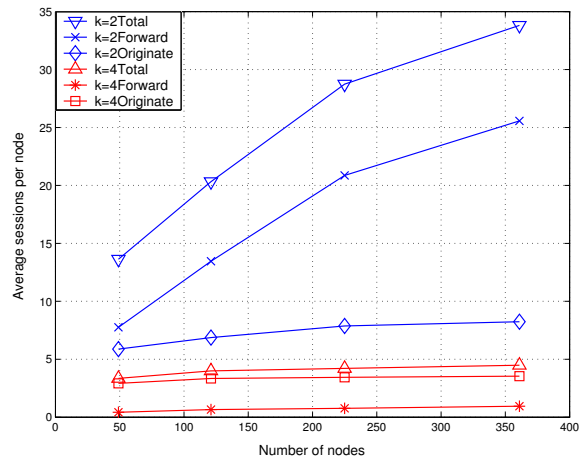


Fig. 8. Average number of total (source-destination and relaying) sessions per node $E[S_a]$ (15), $c = 0.8$, $k > 3$ for convergence, AODV simulations.

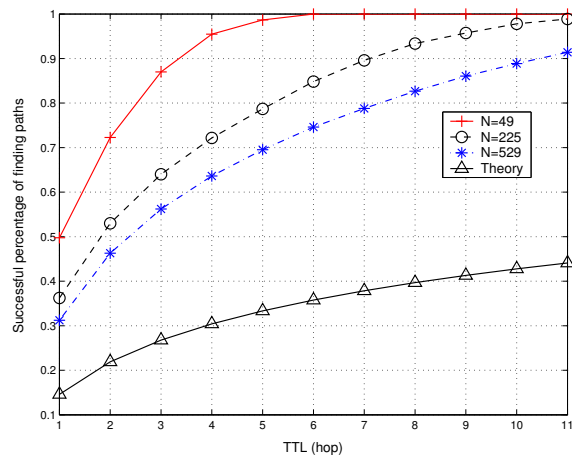


Fig. 9. Successful percentage of finding paths as a function of TTL (19), $k = 2$, $c = 0.8$, AODV simulations.

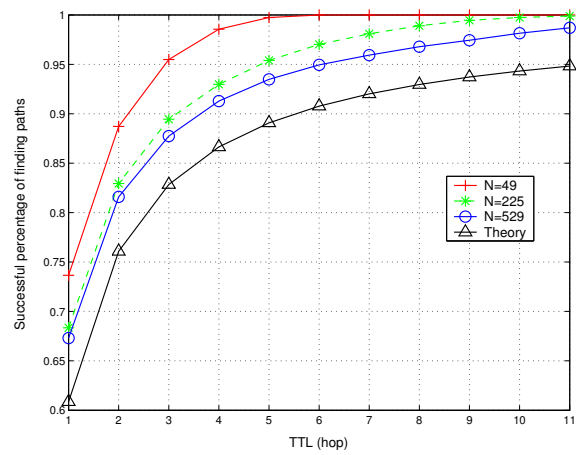


Fig. 10. Successful percentage of finding paths as a function of TTL (19), $k = 3$, $c = 0.8$, AODV simulations.

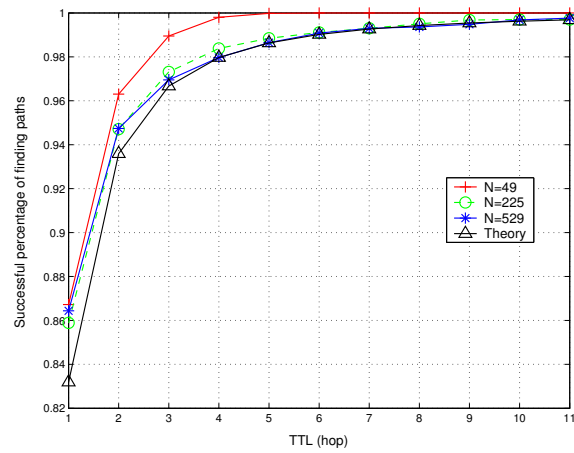


Fig. 11. Successful percentage of finding paths as a function of TTL (19), $k = 4$, $c = 0.8$, AODV simulations.

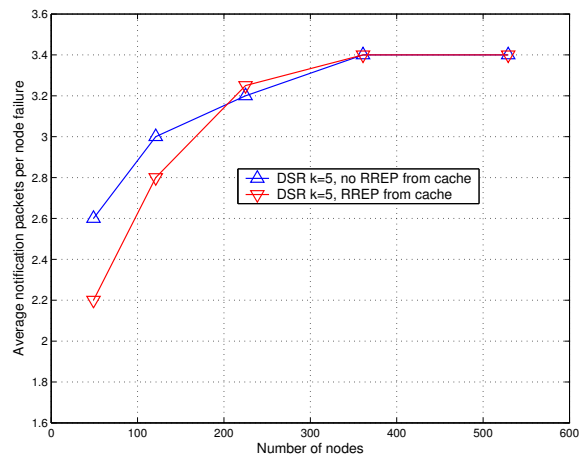


Fig. 12. Effect of route cache on RERR for a failed destination, $k = 5$, $c = 0.5$, DSR simulations.

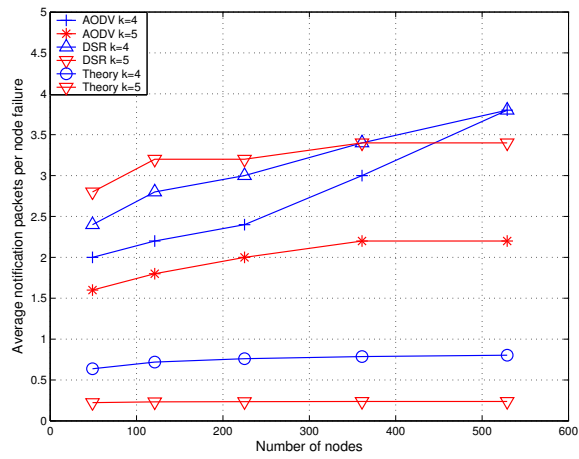


Fig. 13. Average route error packets for a failed node as destination (31), $c = 0.5$.

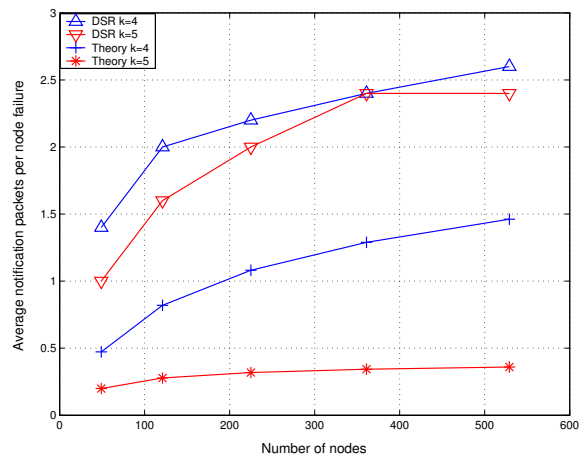


Fig. 14. Average route error packets for a failed node as destination or relay (34), $c = 0.5$.

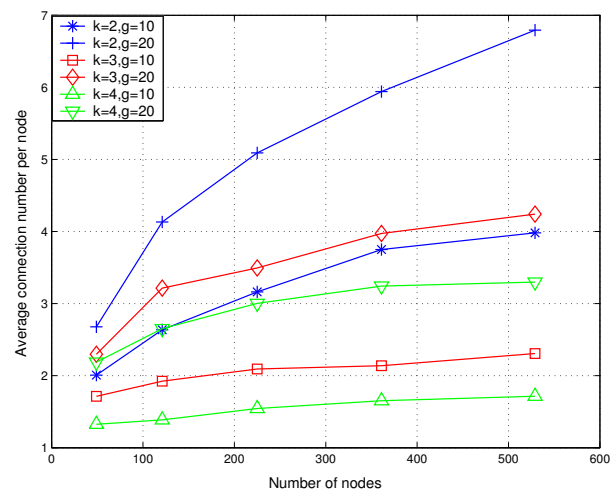


Fig. 15. Average source-destination sessions per node $E[S]$ (42), $c = 0.1r_1^k$, $k > 2$ for converge.

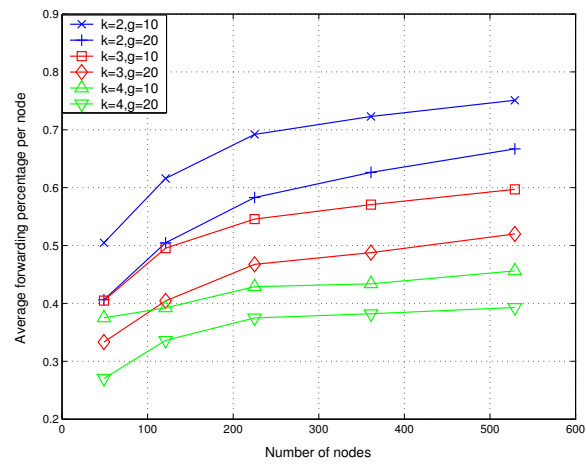


Fig. 16. Average forwarding percentage per node q (48), $c = 0.1r_1^k$, AODV simulations.

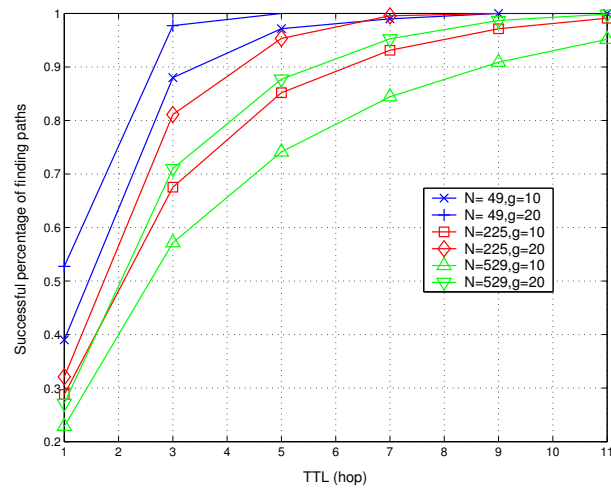


Fig. 17. Successful percentage of finding paths as a function of TTL (50), $k = 2$, $c = 0.1r_1^2$, AODV simulations.

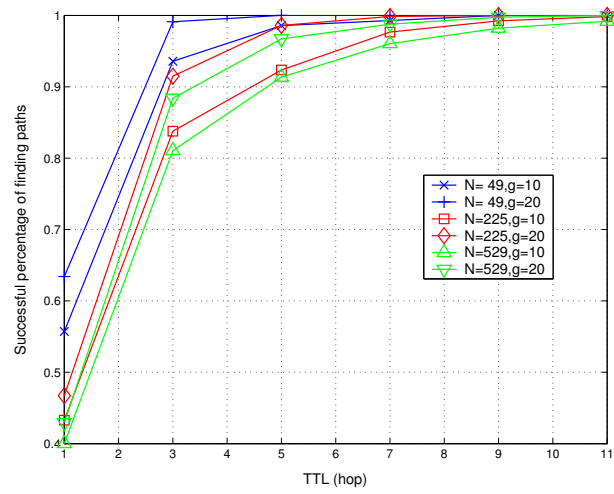


Fig. 18. Successful percentage of finding paths as a function of TTL (50), $k = 3$, $c = 0.1r_1^3$, AODV simulations.

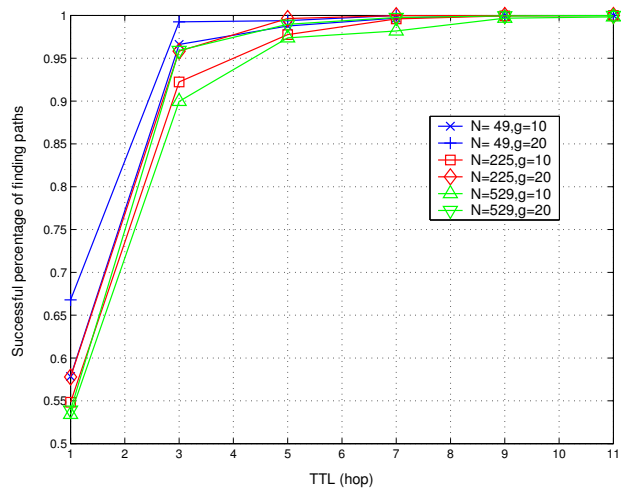


Fig. 19. Successful percentage of finding paths as a function of TTL (50), $k = 4$, $c = 0.1r_1^4$, AODV simulations.

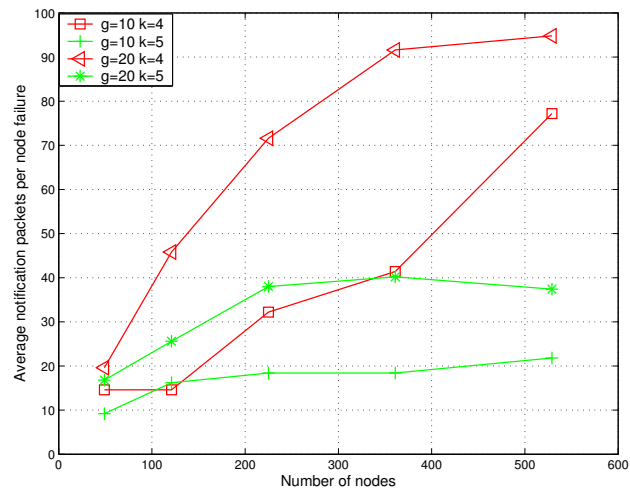


Fig. 20. Average number of route error packets for a fail node as destination, $c = 0.5r_1^k$, AODV simulations

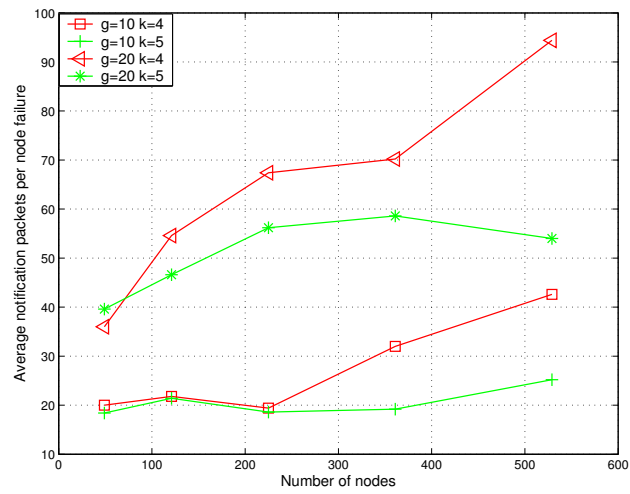


Fig. 21. Average number of route error packets for a fail node as destination (59), $c = 0.5r_1^k$, $k > 5$ for converge, DSR simulations

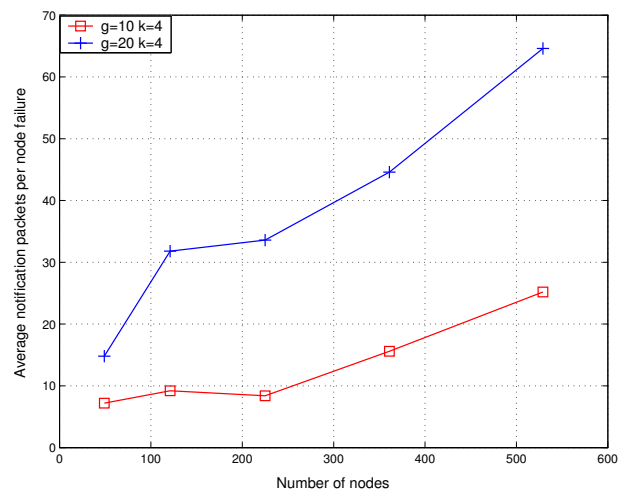


Fig. 22. Average number of route error packets for a fail node as destination or relay (59), $c = 0.5r_1^4, k = 4, k > 5$ for converge, DSR simulations.

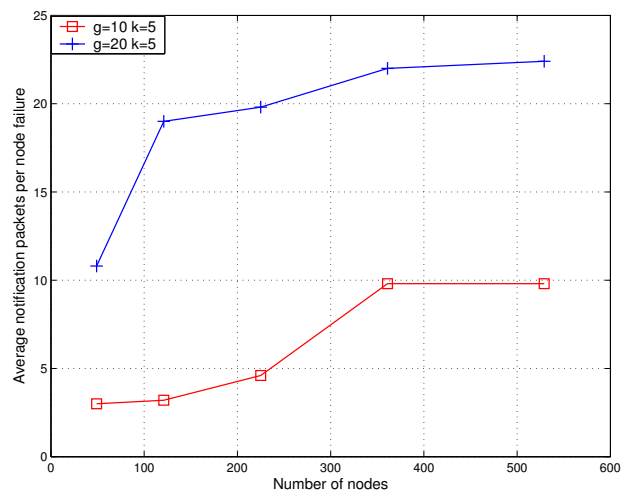


Fig. 23. Average number of route error packets for a fail node as destination or relay (59), $c = 0.5r_1^5$, $k = 5$, $k > 5$ for converge, DSR simulations.