Node Delay Analysis of Routing Protocols in Mobile Ad Hoc Networks

Taesoo Jun
Christine Julien

TR-UTEDGE-2008-009

Node Delay Analysis of Routing Protocols in Mobile Ad Hoc Networks

Taesoo Jun*, and Christine Julien†
The Center for Excellence in Distributed Global Environments
The Department of Electrical and Computer Engineering
The University of Texas at Austin
{hopelist*, c.julien†}@mail.utexas.edu

***Abstract*—Different deployments of mobile ad hoc networks (MANETs) can have widely varying characteristics that greatly impact the behavior of different routing protocols created for these networks. Before applications can be deployed in such environments, it is important for developers to understand the potential quantitative behavior of the protocols that support their applications. Analytical models exist to describe the behavior of MANETs, but they are restricted to simplistic statistical models that represent either node mobility or link connectivity individually without considering the interplay of the two and other important aspects of MANETs. Our previous work has begun developing an integrated analytical framework for routing protocols in MANETs. This framework considers the impact the physical environment, network characteristics, protocols' behaviors, and the applications' communication patterns have on performance. In this paper, we focus specifically on extending our existing analytical model to account for the subtleties of message queueing in MANETs. We describe our model and discuss how queueing aspects are incorporated into our complete framework. We validate our analytical results through network simulation.**

I. Introduction

Wireless mobile ad hoc networks (MANETs) enable network connections in situations where it is impossible or difficult to set up infrastructure. Network links in a MANET are created in a self-organizing manner by participating nodes, which relay data packets for other nodes. Such self-organizing networks introduce several research challenges, among which the problem of determining an optimal routing approach is prominent. A plethora of routing protocols have been proposed for mobile ad hoc networks, but evaluations of these protocols demonstrate that different approaches have benefits under different conditions.

In general, several factors influence the performance of routing protocols in MANETs. Our previous work [13] delineated two categories of parameters: topological parameters and traffic parameters. Topological parameters include factors that influence network topology changes, most often due to node movement (e.g., the speed of nodes in a network, node density, etc.). Traffic parameters define characteristics of data traffic (e.g., the rate of traffic generated, the number of simultaneous sessions, etc.). In [12], we proposed an analytical framework where we derived models of two important performance metrics (i.e., end-to-end delay and throughput) using parameters from connectivity and performance models. Although [12] provided a meaningful analytical framework using statistical modeling, the simulation result demonstrated the necessity to consider additional aspects to obtain more accurate models. In this paper, we seek to quantify the per-node delay that originates from message queueing in MANETs. The need for message queueing in such networks is ascribed mainly to contention associated with data transmission in a shared medium.

In MANET communication, a medium access control (MAC) protocol prevents packets sent by multiple nodes from colliding in a wireless medium. While a source node is able to send packets to a destination node within a radio range, an actual transmission starts only after the sending node acquires access to the medium. While the sender waits to acquire the medium, packets remain within the node; this process introduces delays in packet delivery. While the delay is negligible with light traffic, high traffic scenarios induce a significant delay because high demands to transmit from nodes make it very hard for each node to obtain access to the medium. Although multiple MAC protocols are available for MANETs, most studies adopt IEEE 802.11 DCF (Distributed Coordination Function) [10] as the MAC protocol because the protocol is an international standard and is widely used in real implementations [5], [6], [11].

Previous studies have investigated the maximum achievable throughput [9], [8] and tradeoffs between capacity and delay [7] in wireless ad hoc networks. While these studies provide insight into the performance of wireless ad hoc networks with static or mobile nodes, the results are restricted to scale-level knowledge (e.g., the per-node capacity scales as $\Theta(W/\sqrt{n\log n})$ in a wireless ad hoc network with static nodes). Contrasted to information-theoretic approaches, other work provided the closed form of performance metrics, applying a queueing modeling [4]. Specifically, using an open G/G/1 queueing network model, the authors derive the average end-to-end delay and the maximum achievable throughput in a wireless ad hoc network with stationary nodes.

Since the approaches serve as a valid starting point for our modeling, we utilize them as a foundation. However, our model specifically accounts for *node mobility* in wireless ad hoc networks. In addition, to build the queueing model, we apply the *Kleinrock Independence Approximation*, which states that the queues in a large network behave independently regardless of the dependency of packet flows between connected

queues [14], [1]. In particular, our work captures the delay caused by packet queueing in each mobile node. With the queueing model, we can obtain a value for the average end-to-end delay. As a result, the work herein extends our previous analytical framework by making our framework more accurate and meaningful in consequence. Ultimately, the analytical analysis in this work will provide a good basis for selecting the communication protocol suitable to a particular network situation by representing queueing effects on a protocol's performance.

The remainder of this paper is organized as follows. Section II provides assumptions and system models used to develop our analysis. We present a queueing model for MANETs in Section III and the result of queueing analysis with the model in Section IV. Section V provides simulation results that verify our analytical approaches, and we provide our conclusions and future work in Section VI.

II. System Model

In this section, we provide the fundamental assumptions that underlie the system model we develop throughout the remainder of the paper. These assumptions and system descriptions are applied to our queueing models and guide our derivation.

A. Assumptions

1) Links between neighboring nodes are bi-directional. Thus, all nodes can both send to and receive from any neighbor.
2) All nodes have an equal transmission power, and transmissions can reach up to distance *R*. With an equal radio range, communication links are created with all neighbor nodes within the range and broken when the distance is greater than the range. The connectivity property is reasonably supported by an omni-directional antenna.
3) The Distributed Coordination Function (DCF) in our MAC model removes all hidden and exposed nodes.
4) The MAC approach we model achieves collision freedom. Applying a collision avoidance scheme in the MAC model, we can neglect the effect of packet collisions.
5) The MAC layer sends entire packets from the link layer without packet fragmentation. MAC protocols provide packet fragmentation in practice, but this assumption is valid by restricting the length of a packet from the link layer.

B. System Properties

1) n nodes are initially distributed uniformly over a given space.
2) Nodes move according to some form of random mobility (e.g., random walk model, random way point model, random direction model). In all cases, when nodes reach the boundary of the network, they "wrap-around," re-entering the area from the opposite side.
3) Each node generates traffic at a rate of λ packets per second on average, with the characteristics of i.i.d. Poisson process. The use of a Poisson process has been shown to provide a good model for packet generation from the application layer [1]. Each packet is of a constant size of L bits.
4) Every node can be a source, a destination, and/or a relay node. In addition, sources and destinations are paired at random. Consequently, when a packet is received by a node, the node determines that it is the destination of the packet at reception time with the probability of p_{dst}[1].
5) A node forwards a packet to every neighbor with an equal probability. This implies a relaying node spreads a forwarding packet to every node within a transmission range. This property with the previous properties 3 and 4 describes a random pattern in traffic.
6) Each node uses a random access MAC model which is described in Section IV. The MAC model results in a simple and tractable analytical derivation[2].

In a real world scenario, nodes might move with specific movement patterns, and packets are forwarded according to an applied routing algorithm. However, our system model is statistically reasonable to capture an intuition of a node delay in MANETs by considering random situations. Based on the system model, the following section describes our analytical model of a node delay.

TABLE I
NOTATIONS USED IN THE DERIVATION

Notation	Meaning
n	Number of nodes in the system
a	Width of the territory
b	Height of the territory
R	Radio range
W	Transmission rate in MAC layer
L	Data packet size
λ	Average packet generation rate in a node

III. Queueing Model for MANETs

To identify the effect of packet queueing on performance in MANETs, we need to abstract the packet queueing behavior. In this section, we describe a queueing network model for MANETs which leads to a queueing analysis in Section IV. Specifically, we first state mobility effects on packet arrivals to and departures from a node and then provide a queueing model for MANETs using Kleinrock Independence Approximation. Notations used in this paper are summarized in TABLE I.

A. Mobility Effect on Queueing Behavior

In wireless ad hoc networks, each node transmits packets that the node itself generates or that are forwarded for some neighbor node. Given the system model described in the previous section, packets are guaranteed to be successfully delivered to a neighbor node in static wireless ad hoc networks. However, in the MANETs we model, delivery can fail due to

[1] p_{dst} corresponds to absorption probability in [4].

[2] [3] derived an analytical model for the 802.11 MAC protocol, considering packet collisions. However, the author utilized a numerical method to derive a packet collision probability which was not expressed as a closed form.

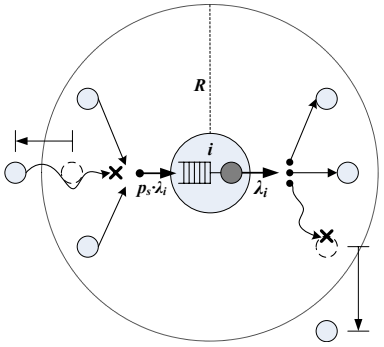

Fig. 1. Packet arrival and departure in a mobile node

node mobility. Fig. 1 shows a situation where node *i* receives packets on the left from its neighbor nodes and forwards them on the right to other neighbor nodes in a MANET. In a mobile scenario, sending nodes and/or receiving nodes can move to a place where wireless links to node *i* become disconnected. In Fig. 1, a node forwarding packets to node *i* moves out of the radio range, resulting in the failure of a packet delivery from node *i*. The node can neither receive packets from node *i* nor send to node *i* after the node moves beyond the radio range of node *i*.

The node *i* has a radio range of R and is modeled as a FIFO queue. In a steady state, every node in the system is symmetric in traffic; each node generates the same amount of traffic and forwards an equal number of packets from the same number of neighbor nodes. Hence, the node *i* should have a balanced influx and efflux of traffic, denoted as λ_i, in a steady state. However, due to mobility, some packet transmissions fail in MANETs. To express the successful delivery in a mobile environment, we apply a probability of successful delivery over one hop, p_s. Given this probability, only $p_s \cdot \lambda_i$ out of the original influx, λ_i, is finally delivered; $(1 - p_s) \cdot \lambda_i$ fails.

B. Approximate Queueing Model

Kleinrock suggested that it is a good approximation to apply an *M/M/1* queueing model for each link in a large network [14]. In fact, the approximation is appropriate for systems which have packet sources with Poisson arrivals in the links, several communication connections, and relatively heavy traffic uniformly distributed over the network. While the approximation is suitable for virtual circuit networks, it also is valid for MANETs with some conditions. In Section II, we stated that each node creates traffic according to a Poisson distribution, and the packet stream spreads evenly due to a probabilistic routing scheme. Therefore, we can apply the Kleinrock Independence Approximation to our system model by adjusting parameters relevant to a node's density (i.e., a, b, n in TABLE I), the node's degree of network connections (i.e., n, R), and the node's traffic generation rate (i.e., λ). The approximation provides us with a simple but reasonable model reflecting the MANETs we described in Section II. In Section IV, we perform a queueing analysis for our model which reasonably approximates an M/M/1 queueing model.

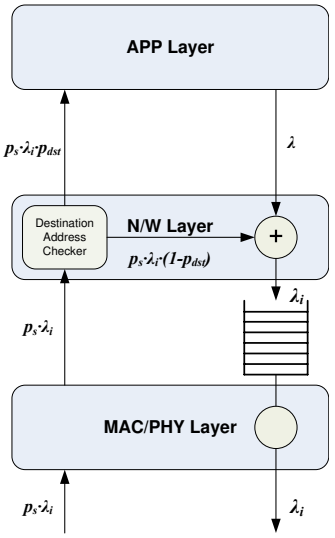

Fig. 2. Queueing model from the viewpoint of protocol stacks

IV. Queueing Analysis

Based on the approximate queueing model in Section III, we rely on analysis results for an M/M/1 queueing system from queueing theory [1]. However, to complete the analysis, it is necessary to define two important queueing system parameters for our network model: an average arrival rate and an average service rate. In this section, we derive the arrival rate and the service rate and then provide results of a queuing analysis for our MANET system model.

A. Packet Arrival Rate

We depict the message queueing within node *i* in Fig. 2. An application layer generates messages at the rate of λ, and the messages are passed to the network layer. Before the messages are transmitted from the node to the next hop, they are stored in a message buffer located between the network layer and the MAC/PHY layer. Node *i* also receives messages from neighbor nodes at the rate of $p_s \cdot \lambda_i$, and the messages are passed to the network layer for routing decisions[3]. Packets destined to the node are dispatched to the application layer, and the remaining packets are enqueued to be forwarded to their next hop. From the system model, an application layer accepts packets at the rate of $p_s \cdot \lambda_i \cdot p_{dst}$ on average, and packets which should be forwarded to other nodes are appended to the message buffer at the rate of $p_s \cdot \lambda_i \cdot (1 - p_{dst})$ on average. Accordingly, the message buffer has two sources of packet arrivals (i.e., the application layer and packet forwarding). Therefore, as we model a node as an M/M/1 queueing system, the buffer maps to a queue, and the MAC/PHY layer to a server of the queue.

[3]From the properties of the Poisson process, two independent Poisson processes with average rates of λ_1, λ_2 are merged into a Poisson process with the rate of $\lambda_1 + \lambda_2$, and Poisson processes split with probability (p), $(1 - p)$ from a Poisson process with an average rate of λ are also Poisson processes with rates of $p \cdot \lambda$, $(1 - p) \cdot \lambda$, respectively.

In a steady state, referring to Fig. 2, an average packet arrival rate at node i, λ_i is expressed as:

$$\lambda_i = p_s \cdot \lambda_i \cdot (1 - p_{dst}) + \lambda.$$

By rearranging with respect to λ_i, we obtain λ_i as:

$$\lambda_i = \frac{\lambda}{1 - p_s + p_s \cdot p_{dst}}. \quad (1)$$

B. Packet Service Rate

Packets arriving at the queue are served in the MAC/PHY layer and flow out to the next hop. To identify an average packet service rate, we derive an average packet service time, which is an inverse of the service rate. Before deriving an average packet service time in a node, we first describe a random access MAC model that captures the back-off scheme and DCF mechanism implemented by the IEEE 802.11 MAC protocol. To avoid collisions in the wireless medium, the MAC model coordinates nodes' transmissions by assigning a random count-down timer to trigger a transmission and by exchanging information between a sender and a receiver prior to an actual data transmission.

For analytical analysis of IEEE 802.11, previous models have relied on tailored MAC models that simplify mathematical derivation [3], [4]. In this work, we use a MAC model that is simple but has performed reasonably well in modeling the IEEE 802.11 MAC protocol in other analytical approaches [4]. In this random access MAC model, nodes within $2R$ interfere with each other, and the back-off timer is generated by sampling randomly from a contention window size that increases exponentially. The increase of a contention window follows the same algorithm from IEEE 802.11 and is described in detail in the end of this section. When the back-off timer of a node expires, the node starts the transmission. In addition, the back-off timers of interfering nodes are frozen while one node sends a packet, and all the frozen timers resume immediately after the transmission finishes[4].

In DCF, each node performs RTS/CTS handshaking before sending a data packet. Specifically, a sending node transmits an RTS packet, and a receiving node responds with a CTS packet when the correspondents are within radio range of each other. During RTS/CTS handshaking, a sender and a receiver perform the virtual carrier-sense mechanism, which reserves the medium against interfering nodes. Specifically, an RTS and a CTS packet contain information about the time duration needed to finish delivering the data packet corresponding to the RTS/CTS packet. When a node receives an RTS or a CTS packet, the node updates the Network Allocation Vector (NAV) which indicates the expected duration of future traffic on the medium. Upon receiving a correct CTS packet, the sending node starts to send the data packet. As a final step of a data delivery procedure, the receiving node replies to the sending node with an ACK packet when reception of the data packet completes.

On the other hand, as a receiver moves out of a sender's radio range, a failure in the data delivery procedure can occur either during RTS/CTS handshaking or during data/ACK exchange. When the receiver moves beyond the sender's radio range while the sender is transmitting an RTS, the sending node will fail to receive the corresponding CTS packet. After transmitting an RTS packet, a node waits for a timeout interval to receive a CTS, T_{CTSto}, before the node initiates a back-off timer. When the node fails to receive a corresponding CTS packet, it increases a short retry count. Once RTS/CTS handshaking succeeds, a node starts to send data. However, when a receiver goes out of the sender's radio range during the data packet transmission, the sender does not receive the corresponding ACK packet. As a similar case to RTS/CTS handshaking, a sending node permits a timeout interval for ACK packets, T_{ACKto}, before the node triggers a back-off timer. However, when a sender fails to receive a corresponding ACK packet, the node increases a long retry count if the size of a data packet is longer than *dot11RTSThreshold*[5]. After the re-initiated back-off timer expires, the sending node starts the procedure for a data delivery. A sending node stops the procedure and discards the data packet when a short retry count reaches a short retry limit, N_{sr} or a long retry count reaches a long retry limit, N_{lr}. We herein define the packet discards as failures in a packet delivery.

When a packet delivery fails, other interfering nodes adjust their NAV; if a node does not detect any valid packet on the medium during a time interval, T_{NAVRS}, the node is allowed to reset its NAV and enters the back-off window, starting its own back-off timer. While T_{NAVRS} is specified as $2T_{SIFS} + T_{CTS} + 2T_{slot}$, neither T_{CTSto} nor T_{ACKto} is provided in the IEEE 802.11 specification [10][6]. In this work, by setting the timeout interval to be the time for the corresponding packet to return as well as the time required for a node to determine that the medium is idle, we assign T_{CTSto} as $T_{CTS} + T_{SIFS} + T_{DIFS}$ and T_{ACKto} as $T_{ACK} + T_{SIFS} + T_{DIFS}$. In fact, since a CTS packet has the same packet length as an ACK packet, T_{CTSto} is equal to T_{ACKto}. Referring to the IEEE 802.11 specification, T_{NAVRS} has the same duration as our T_{CTSto} and T_{ACKto} since T_{DIFS} is defined as $T_{SIFS} + 2T_{slot}$. It is interesting that using our assignment of T_{CTSto} and T_{ACKto}, all associated nodes (i.e., a sender, a receiver, and interfering nodes) enter the back-off window at the same time and have the same opportunity to hold the medium. Fig. 3 illustrates NAV reset scenarios when each of a CTS packet or an ACK packet fails.

While packet delivery can fail in either RTS/CTS or data/ACK exchange, we assume that most failures occur during RTS/CTS handshaking. It is probable that link failure is

[4]It is unrealistic to assume that all interfering nodes detect the medium status and, accordingly, freeze or resume their back-off timer coincidentally, resulting in a collision-free condition. However, the MAC model provides mathematical tractability and a good approximation to the case with packet collisions [4].

[5]In 802.11 specification, *dot11RTSThreshold* is not specified as a fixed value but is set per node, depending on the operating environment

[6]T_{SIFS}, T_{DIFS} and T_{slot} are a time of short interframe space, a time of DCF interframe space and a slot time respectively, which are defined in the IEEE 802.11 specification.

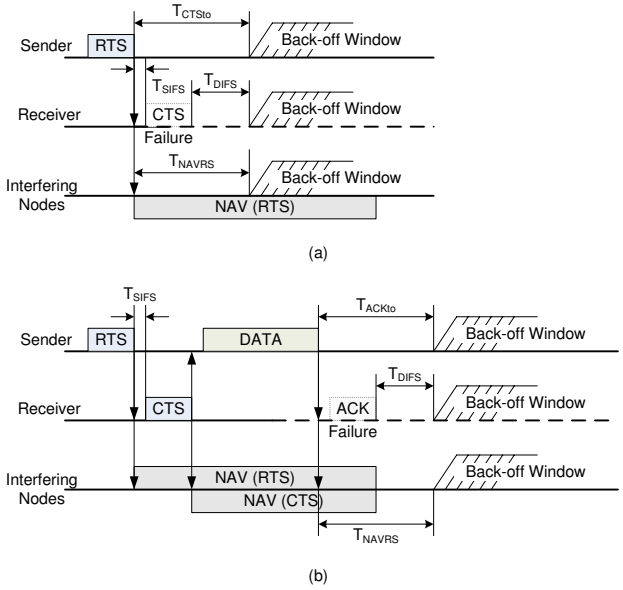

Fig. 3. NAV reset due to failure in (a) RTS/CTS handshaking and (b) data/ACK handshaking

detected more often during this RTS/CTS exchange since the speed of packet transmission is fast compared to the speed of nodes. To verify the assumption, we provide simulation results in Section V.

Based on the description of the IEEE 802.11 DCF, we can derive an average service time for a packet as follows. We first define p_l as a probability of successful delivery along a single hop at each transmission attempt. p_l is different from p_s presented in Section III in that p_s includes all the cases before a packet is discarded. Consequently, p_l is related to p_s as:

$$p_f = 1 - p_s = (1 - p_l)^{N_{sr}}, \tag{2}$$

where p_f is a probability of delivery failure over a single hop. Considering packet delivery failures due to nodes' mobility, we can express an average service time for a packet in a node, $\overline{T_{svr}}$ as:

$$\begin{aligned} \overline{T_{svr}} &= p_l \cdot \overline{T_s} + \sum_{k=1}^{N_{sr}-1} (1-p_l)^k p_l (k\overline{T_f} + \overline{T_s}) + \\ &\quad (1-p_l)^{N_{sr}} N_{sr} \overline{T_f}, \end{aligned} \tag{3}$$

where $\overline{T_s}$ is an expected time spent in a delivery success, and $\overline{T_f}$ is an expected time spent in a delivery failure described in the above. In Eq. 3, the first and second terms correspond to the cases that a data packet delivery succeeds in the first try and in k-th try ($k \leq N_{sr}$), and the last term corresponds to cases in which the packet delivery finally fails and the packet is dropped.

While the mean service time required to serve a data packet for static wireless ad hoc networks has been derived [4], we present a different derivation of $\overline{T_s}$, using a similar approach but taking node mobility into account. Let a random variable P_i denote the service time for a packet transmission in node i. In addition, we denote random variables T_i and F_i as a back-off time in node i and the number of times the back-off timer is frozen before it expires in node i, respectively. Then, we can write P_i as:

$$P_i = T_i + F_i \overline{T_{frz}} + T_{eff}^{data}, \tag{4}$$

where T_{eff}^{data} is an effective time spent sending a data packet, and $\overline{T_{frz}}$ is an expected time for the node to be frozen due to an interfering node's packet transmission. From the IEEE 802.11 specification, T_{eff}^{data} is represented as:

$$\begin{aligned} T_{eff}^{data} &= T_{RTS} + T_{SIFS} + T_{CTS} + T_{SIFS} + \\ &\quad \frac{L}{W} + T_{SIFS} + T_{ACK}. \end{aligned} \tag{5}$$

Since an interfering node can succeed to send a data with probability of p_l and can fail with probability of $(1 - p_l)$ with the timeout mechanism, $\overline{T_{frz}}$ is expressed as:

$$\overline{T_{frz}} = p_l T_{eff}^{data} + (1 - p_l)\overline{T_f^{to}}, \tag{6}$$

where $\overline{T_f^{to}}$ is an expected time to determine failure of a packet delivery by timeout mechanisms. Since we assume that a packet delivery mainly fails in the phase of RTS/CTS handshaking, $\overline{T_f^{to}}$ is given as:

$$\overline{T_f^{to}} \approx T_{RTS} + T_{CTSto}. \tag{7}$$

Since a back-off timer in a node is frozen whenever its interfering neighbors start to transmit, $\overline{F_i}$ represents the number of nodes which are ready to transmit on average among the interfering neighbors of node i. Using a utilization factor of a server in our queueing model, ρ (i.e., the proportion of time that the MAC works in transmitting a packet), $\overline{F_i}$ is expressed as:

$$\overline{F_i} = \rho \overline{N_{int}},$$

where $\overline{N_{int}}$ is the expected number of interfering nodes. Since all nodes in our system are symmetric, the utilization factor, ρ is given by:

$$\rho = \rho_i = \frac{\lambda_i}{\mu_i} = \lambda_i \overline{T_{svr}}, \tag{8}$$

where μ_i is an average packet service rate in node i[7]. As nodes are distributed uniformly at all times in the system model, $\overline{N_{int}}$ is given by:

$$\overline{N_{int}} = \pi(2R)^2 \cdot \frac{n}{a \cdot b}. \tag{9}$$

Therefore, considering that all nodes are identical and using the above derivations,

$$\overline{T_s} = \overline{P_i} = \frac{1}{\xi} + \rho \overline{N_{int} T_{frz}} + T_{eff}^{data}. \tag{10}$$

In Eq. 10, the first term corresponds to the duration of the back-off timer, and the second term corresponds to the amount of time for which the back-off timer is frozen due to transmissions from interfering nodes. The last part is the time

[7]Since nodes are symmetric from the system model, each node has the same value of μ_i

required to send the data packet. Based on the description of a timeout occurrence and a similar approach, $\overline{T_f}$ is expressed as:

$$\overline{T_f} = \frac{1}{\xi} + \rho\overline{N_{int}T_{frz}} + \overline{T_f^{to}}. \quad (11)$$

Using Eq. 3, 10, and 11, we can obtain $\overline{T_{svr}}$, rearranging with respect to $\overline{T_{svr}}$. For our convenience, let $T_{to}^* = 1/\xi + \overline{T_f^{to}}$, $T_{data}^* = 1/\xi + T_{eff}^{data}$, and $T_{frz}^* = \overline{N_{int}T_{frz}}$, where T_{to}^* and T_{data}^* imply times including an average back-off time, and T_*^{frz} indicates an average frozen time due to an average interfering nodes' transmission. Consequently, we obtain $\overline{T_{svr}}$ as:

$$\overline{T_{svr}} = \frac{\sum_{k=0}^{N_{sr}-1}(1-p_l)^k p_l(T_{to}^* k + T_{data}^*) + N_{sr}(1-p_l)^{N_{sr}}T_{to}^*}{1-\sum_{k=0}^{N_{sr}-1}(k+1)(1-p_l)^k p_l\lambda_i T_{frz}^* - N_{sr}\lambda_i T_{frz}^*(1-p_l)^{N_{sr}}}. \quad (12)$$

Finally, an average packet service rate, μ_i is given as the reciprocal of the average service time, $\overline{T_{svr}}$.

C. Parameter Adjustment

In the previous section, we derived an average packet arrival rate and an average packet service rate in each node. While the derivations are expressed with parameters from a system description, a parameter is used based on our simple traffic modeling; p_{dst} is introduced to represent what portion of received packets are dispatched to the application layer in the node. In fact, the parameter enables us to abstract a degree of packet forwarding; for p_{dst} close to 1, a packet would travel just one hop, and for p_{dst} close to 0, most incoming packets would be forwarded. In other words, by changing this parameter, we can adjust the number of hops a packet travels on average. Thus, we need to determine an appropriate value for p_{dst} that reflects the traffic situation in a particular system.

The lengths of communication paths (in hops) impact performance in MANETs; for example, the parameter is deeply relevant to an average end-to-end delay or an average route discovery time. In [2], the authors provide an average hop count with uniformly distributed nodes on a rectangular area for a wireless ad hoc network. The average hop count represents the number of hops between an arbitrary source and destination pair in the network.

On the one hand, using p_{dst}, we can express an average hop count, $\overline{N_{hop}}$ as:

$$\overline{N_{hop}} = \sum_{k=1}^{\infty} k\cdot(1-p_{dst})^{k-1}p_{dst} = \frac{1}{p_{dst}}. \quad (13)$$

On the other hand, referring to [2], $\overline{N_{hop}}$ is approximated in a dense network[8] as:

$$\overline{N_{hop}} \approx \frac{E[D_{sd}]}{R}, \quad (14)$$

[8]When node's density is high in the network, a path from a source and a destination is almost a straight line.

where $E[D_{sd}]$ is an average distance between an arbitrary source and destination pair. In addition, in a rectangle area with a width of a and a height of b, $E[D_{sd}]$ is given as:

$$E[D_{sd}] = \frac{1}{15}\left[\frac{a^3}{b^2} + \frac{b^3}{a^2} + dg(a,b)\cdot\left(3 - \frac{a^2}{b^2} - \frac{b^2}{a^2}\right)\right] + \frac{1}{6}\left[\frac{b^2}{a}\cosh^{-1}\frac{dg(a,b)}{b} + \frac{a^2}{b}\cosh^{-1}\frac{dg(a,b)}{a}\right], \quad (15)$$

where $dg(a,b)$ is $\sqrt{a^2+b^2}$ (i.e., diagonal of the rectangular area). From Eq. 13 and 14, we can determine an appropriate p_{dst} as:

$$p_{dst} \approx \frac{R}{E[D_{sd}]}. \quad (16)$$

In addition, we apply an average back-off time, $1/\xi$ for our random access MAC model. Referring to the IEEE 802.11 specification, the contention window size, CW_i increases exponentially, depending on the back-off stage, i, and the window size is given as:

$$CW_i = \begin{cases} 2^i CW_{min} & \text{if } 0 \le i \le S_{max} \\ CW_{max} & \text{if } S_{max} < i \end{cases} \quad (17)$$

where CW_{min} and CW_{max} are the minimum and maximum contention window size respectively, and S_{max} is the maximum back-off stage. Specifically, CW_{min} and CW_{max} depend on the PHY layer; in the case of Direct-Sequence Spread Spectrum (DSSS), CW_{min} and CW_{max} are defined as 32 and 1024, and consequently S_{max} is calculated as 5.

Taking packet delivery successes into account, we can express an average contention window size, $\overline{CW}$ as:

$$\overline{CW} = \sum_{k=0}^{N_{sr}-1}(1-p_l)^k p_l CW_k + (1-p_l)^{N_{sr}}CW_{N_{sr}}. \quad (18)$$

In Eq. 18, the first term corresponds to the cases when a packet is successfully delivered. The second term corresponds to the case when a packet delivery fails and the packet is dropped. Since a contention window value is chosen randomly from the range [0, CW_i] in back-off stage i, an average back-off time, $1/\xi$ is obtained as:

$$\frac{1}{\xi} = \frac{\overline{CW}T_{slot}}{2}. \quad (19)$$

D. Node Delay Analysis

Referring to the analysis result of an M/M/1 queueing system, we obtain an average delay in each node, $E[T]$ as:

$$E[T] = \frac{1}{\mu_i - \lambda_i} = \frac{\overline{T_{svr}}}{1 - \overline{T_{svr}}\lambda_i}. \quad (20)$$

In addition, an average queueing delay in only a queue of each node, $E[T_Q]$ is given as:

$$E[T_Q] = E[T] - \overline{T_{svr}} = \frac{\lambda_i\overline{T_{svr}}}{1 - \overline{T_{svr}}\lambda_i}. \quad (21)$$

Fig. 4 depicts the average delay in our queueing model as the probability of successful one hop delivery varies. The

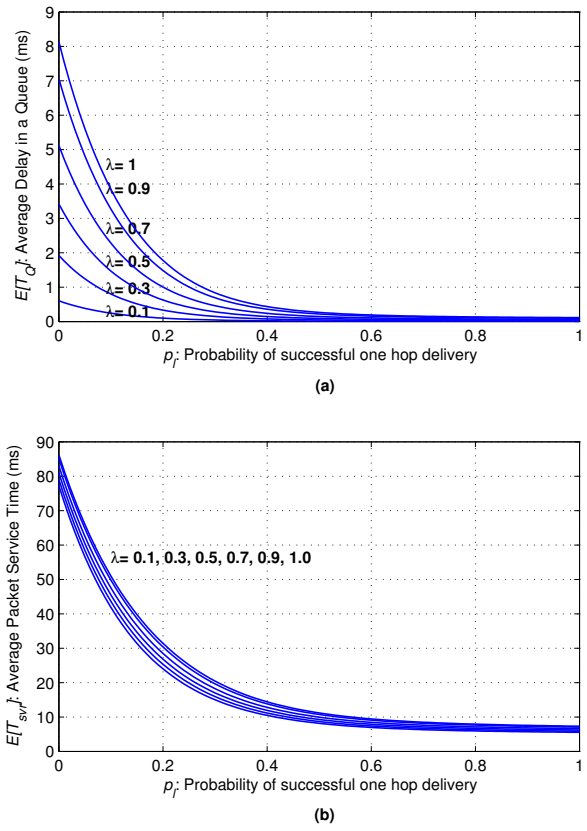

Fig. 4. Numerical analysis on an average delay in (a) a message queue and (b) a server of the queue

graphs show values for different average packet arrival rates from the application layer (λ). Numerical evaluations shown in Fig. 4 apply the following parameters: L = 512 (byte), W = 1 (Mbps), n = 50, a = 1500 (m), b = 300 (m), and R = 250 (m). For the PHY layer, we apply DSSS. Other parameters are obtained from the IEEE 802.11 specification.

Basically, if the MAC layer cannot serve a packet before the next packet arrives, the next packet is queued for delivery and experiences a delay in the message buffer. In Fig. 4(a), when the arrival rate is relatively small (e.g., λ = 0.1), the average delay a packet experiences in the message buffer is nearly constant. When the arrival rate is small, the MAC layer has enough time to serve packets such that additional delay in the buffer is unlikely. On the other hand, when the arrival rate is relatively large (e.g., $\lambda = 1.0$), the average delay in the message buffer increases as p_l decreases (i.e., the network becomes dynamic). Accordingly, Fig. 4(b) shows an increase in average packet service times as p_l decreases. The service time for a packet in the MAC layer depends little on the packet generation rate. In addition, our model demonstrates that the average packet delay on a particular node is dominated by the MAC layer's service time rather than a delay in the message buffer, even for high arrival rates.

TABLE II
SIMULATION PARAMETERS

Parameter	Value
Network size($a \times b$)	1500 m × 300 m
Simulation time	900 sec.
Number of mobile hosts(n)	50
Radio range(R)	250 m
Pause time	1, 100, 200, $\cdots$, 900 sec.
Packet size(L)	512 byte
Transmission rate in MAC(W)	1 Mbps
Avg. packet generation rate(λ)	0.1, 0.3, 0.4, 0.5 packet/sec.

V. SIMULATION RESULTS

In this section, we compare expected queueing delays calculated from our analytical framework with measured delays from simulations. We performed simulations with ns-2.33, applying IEEE 802.11 as the MAC protocol and DSDV [15] as the routing protocol. In practical operation, a routing protocol generates control packets for purposes such as route maintenance, and control packets influence packet queueing and delay on each node. However, it is hard to directly model the effect of control packets because they depend not on the routing protocol as much as on the operating environment. Previous work has shown that proactive routing protocols such as DSDV are more independent of changing environments than reactive ones [5], [11]. Therefore, to minimize our evaluation's dependency on the impact of control messages, we selected DSDV as the routing protocol. We generated 100 independent mobility scenarios and attached an exponential traffic source to all nodes for symmetry. Mobility is based on random waypoint mobility presented in [16] where a node's speed is normally distributed using a minimum of $5m/s$ and a maximum of $20m/s$, and a pause time is uniformly chosen over a time; an average pause time is a half of the time. In addition, since we are interested in the steady state, we took the simulation data into account after $200sec.$ from the simulation starting time. While we assume freedom of packet collision, it is highly probable that packets collide in a real situation or even in a simulation environment. It is likely that packets collide with high probability under heavy traffic and under high mobility degree. To follow the assumption, we generate a relatively light traffic(e.g., $\lambda \leq 0.5$). Detailed simulation parameters are presented in TABLE II.

We first validate our assumption that the RTS retransmission dominates data retransmission. For each simulation run, we measured the proportion of RTS retransmissions out of the total retransmissions. Specifically, we measured the ratio R_{RTS}^{retx}, expressed as

$$R_{RTS}^{retx} = \frac{N_{RTSretx}^{total}}{N_{RTSretx}^{total} + N_{DATAretx}^{total}},$$

where $N_{RTSretx}^{total}$ is the total number of RTS retransmissions, and $N_{DATAretx}^{total}$ is the total number of data retransmissions.

Fig. 5 depicts the results with 95% confidence intervals. As shown, more than 95% of all retransmissions are RTS retransmissions, though the proportion decreases as the network

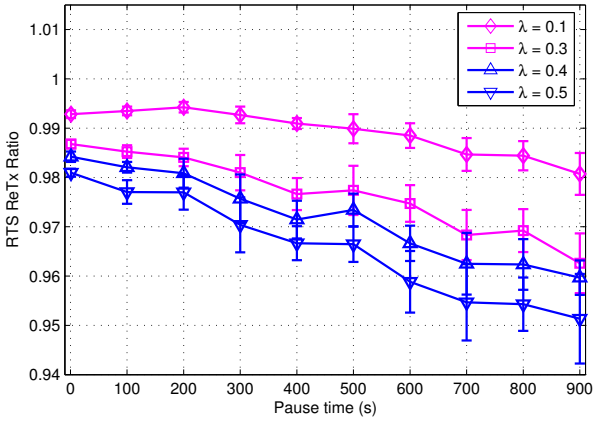

Fig. 5. Proportion of RTS retransmission among all retransmissions

becomes more static (i.e., as pause time increases). The reason is because, in a static network, it is more probable that a data packet which is longer than RTS packet fails to be delivered after RTS/CTS handshaking succeeds.

We next use our simulations to verify our analytical derivation presented in Section IV. To measure the average queueing delay in simulation, we marked the time when a data packet entered the packet queue located between the MAC layer and the network layer [9] and again when the packet is served in the MAC layer. We also calculated an average for all other residual time on the node for each data packet. However, to compare simulation results with our analytical model, it is necessary to find the the probability of successful transmission over one hop, p_l corresponding to the mobility degree (i.e., pause time) in each simulation setting. For this purpose, we additionally measured the total number of data packets, N_{data}^{total}, and the total number of ACK packets, N_{ACK}^{total}. When a packet delivery fails, the sender does not receive the corresponding ACK packet. Intuitively, the probability of a final packet delivery failure is approximately the ratio of the number of data transmissions of a data packet that are not finally replied to with an ACK packet to the number of data packet transmissions. Using the notation from Eq. 2, this probability is expressed as:

$$p_f \approx \frac{N_{DATAtx}^{total} - N_{ACKsuc}^{total}}{N_{DATAtx}^{total}}, \tag{22}$$

where N_{DATAtx}^{total} is the total number of a data packet transmission (excluding retransmissions), and N_{ACKsuc}^{total} is the total number of ACK packet received successfully. By referring to Eq. 2, 22, we obtains p_l as:

$$p_l = 1 - \sqrt[N_{sr}]{p_f} \approx 1 - \sqrt[N_{sr}]{\frac{N_{DATAtx}^{total} - N_{ACKsuc}^{total}}{N_{DATAtx}^{total}}}. \tag{23}$$

It is notable that we can measure the parameters in the right hand side of Eq. 23 and calculate p_l at runtime.

[9]In ns-2, the queue is implemented as IFQ module.

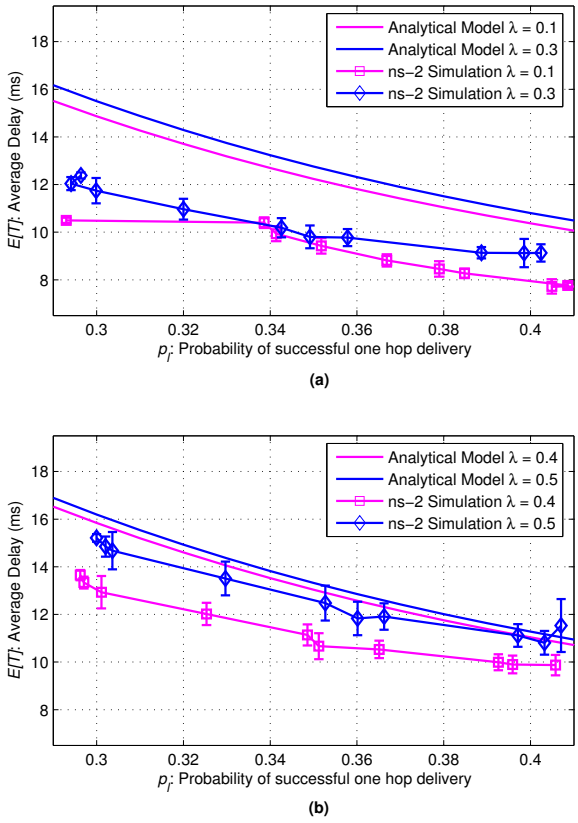

Fig. 6. Comparison of analytical results with simulation results

Fig. 6 shows the comparison of average delays in a node (i.e., $E[T]$ in Eq. 20) from our analytical model with those from simulations with ns-2. The simulation results are depicted with 95% confidence intervals. In our simulation runs, p_l is varying from 0.29 to 0.42. Basically, we can not control p_l explicitly but can change the p_l implicitly by controlling mobility degrees in the simulation runs; the higher a mobility degree is, the lower the p_l would be, and vice versa. We tried to obtain a higher value of p_l but failed; in our simulations, p_l was not over 0.5 even in stationary scenarios. The reason is because even a very few packet delivery failures result in a severe decrease of p_l according to Eq. 23.

Over the range of p_l, we observe that simulation measurements show the same tendency as our analytical model with a certain amount of deviation. For all λs, simulation measurements have smaller delays than the analytical model. The reason is because the analytical model does not take border effects into account. Specifically, in Eq. 9, we derived an expected number of interfering nodes under a torus-shape assumption which is presented in Section III. However, in the simulation environment, the number of interfering nodes may be less than that from Eq. 9. In addition, while our analytical model seems not to be affected by λ, we observe that the average delay in the node, $E[T]$, is increasing as the traffic

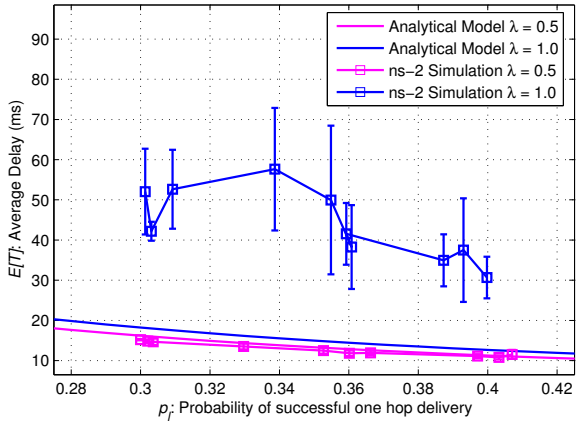

Fig. 7. Effect of a relatively heavy traffic

is heavier (i.e., a larger value of λ). In a relatively heavier traffic environment (e,g,. λ = 0.5), a probability of a packet collision will increase, which we do not consider in modeling. Finally, while our analytical model assumes probabilistic routing scheme, packets flow along a deterministic path in a real situation. When each node generates many packets, nodes located in a good position to relay packets will handle more packets than other nodes. In this situation, the nodes suffer from distinguishing delay due to an exponential backoff scheme of a contention window.

Fig. 7 depicts average delays in a node under heavy traffic. Given that the denominator of Eq. 12 should be positive, we can obtain a theoretical upper bound of λ as about 1.7 for p_l valued between 0.29 and 0.42. The difference between the theoretical upper limit and the simulation results is due to the fact that more packets collide in heavy traffic situations. As the probability of a packet collision increases, a packet spends more time blocked by the contention window whose size expands exponentially. Although our model turns out to be valid with traffic lighter than $\lambda = 0.5$, given the impact of congestion, the traffic corresponding to $\lambda = 0.5$ is not light at all.

VI. Conclusion

In our previous work, we have introduced a software tool with which we separate the implementation of an application's interactive portions from that of the communication infrastructure for the interactions. In addition, we presented an analytical framework which includes detailed models for two important protocol performance metrics: end-to-end delay and throughput. The models are expressed with respect to environment-, protocol-, and application-dependent parameters.

In this paper, we investigated message queueing in MANETs, which extends our existing analytical framework. Our analytical model captures a per-node delay in MANETs, and the model was validated by extensive simulations. We derived the model mainly as a function of p_l and provided a mechanism to measure p_l at runtime. In addition, the queueing model is described with feasible parameters which can be obtained from systems (e.g., W), designers (e.g., λ) or operating environments (e.g., p_l). By incorporating the analytical model for a queueing effect into our analytical framework, we can estimate the performance of MANETs associated with potential concrete implementations more accurately. When applications operate in MANETs, our framework can provide a software designer with the potential impact on overall application performance under a specific operational environment.

Acknowledgements

The authors would like to thank the Center for Excellence in Distributed Global Environments for providing research facilities and the collaborative environment. This research was funded, in part, by the National Science Foundation (NSF), Grants # CNS-0615061 and CNS-0620245. The conclusions herein are those of the authors and do not necessarily reflect the views of the sponsoring agencies.

References

[1] D. Bertsekas and R. Gallager. *Data Networks*. Prentice Hall, Inc., 2nd edition, 1992.

[2] C. Bettstetter and J. Eberspächer. Hop distances in homogeneous ad hoc networks. In *Proceedings of IEEE VTC*, pages 2286–2290, April 2003.

[3] G. Bianchi. Performance analysis of the ieee 802.11 distributed coordinationfunction. *IEEE Journal on Selected Areas in Communications*, 18(3):535–547, 2000.

[4] N. Bisnik and A. Abouzeid. Queuing network models for delay analysis of multihop wireless ad hoc networks. In *Proceedings of ACM IWCMC*, pages 773–778, New York, NY, USA, 2006. ACM.

[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 the 4th Annual ACM/IEEE International Conference on Mobile Computing and Networking*, pages 85–97, 1998.

[6] S. R. Das, C. E. Perkins, and E. M. Royer. Performance comparison of two on-demand routing protocols for ad hoc networks. In *Proceedings of the IEEE INFOCOM*, volume 1, pages 3–12, March 2000.

[7] A. E. Gamal, J. Mammen, B. Prabhakar, and D. Shah. Throughput-delay trade-off in wireless networks. In *Proceedings of IEEE INFOCOM*, pages 464–475, March 2004.

[8] M. Grossglauser and D. N. C. Tse. Mobility increases the capacity of ad hoc wireless networks. *IEEE/ACM Transactions on Networking*, 10(4):477–486, 2002.

[9] P. Gupta and P. R. Kumar. The capacity of wireless networks. *IEEE Transactions on Information Theory*, 46(2):388–404, March 2000.

[10] IEEE Std. 802.11-1999. *Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications*. Reference number ISO/IEC 8802-11:1999(E), IEEE Std. 802.11, 1999 edition, 1999.

[11] P. Johansson, T. Larsson, N. Hedman, B. Mielczarek, and M. Degermark. Scenario-based performance analysis of routing protocols for mobile ad-hoc networks. In *Proceedings of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking*, pages 195–206, August 1999.

[12] T. Jun, A. Dalton, S. Bodas, C. Julien, and S. Vishwanath. Expressive analytical model for routing protocols in mobile ad hoc networks. In *Proceedings of the IEEE International Conference on Communications*, May 2008.

[13] T. Jun and C. Julien. Automated routing protocol selection in mobile ad hoc networks. In *Proceedings of the 2007 ACM Symposium on Applied Computing*, pages 906–913, March 2007.

[14] L. Keinrock. *Communication Nets: Stochastic Message Flow and Delay*. McCraw-Hill, 1964.

[15] C. Perkins and P. Bhagwat. Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers. *ACM SIGCOMM Computer Commuication Review*, 24(4):234–244, October 1994.

[16] J. Yoon and B. Noble. Random waypoint considered harmful. In *Proceedings of IEEE INFOCOM*, pages 1312–1321, April 2003.