Expressive Analytical Model for Routing Protocols in Mobile Ad Hoc Networks

Taesoo Jun
Angela Dalton
Shreeshankar Bodas
Christine Julien
Sriram Vishwanath

TR-UTEDGE-2007-017

Expressive Analytical Model for Routing Protocols in Mobile Ad Hoc Networks

Taesoo Jun, Angela Dalton, Shreeshankar Bodas,
Christine Julien, and Sriram Vishwanath
The Center for Excellence in Distributed Global Environments
The Department of Electrical and Computer Engineering
The University of Texas at Austin
{hopelist, adalton, c.julien}@mail.utexas.edu
{bodas, sriram}@ece.utexas.edu

Abstract—**Many routing protocols exist for mobile ad hoc networks. To select the most appropriate protocol, evaluation of candidate protocols must be performed with respect to a specific operating environment, which is not an easy task. However, selecting the best protocol can be a key factor in system behavior, determining whether the system successfully satisfies application requirements. Most of the relevant research in this area relies on simulation studies or empirical analysis to select a routing protocol, requiring an infeasible amount of time and resources for the approaches to be used in real-time decision making. In this paper we describe work toward analytically expressing protocol performance metrics in terms of environment-, protocol-, and application-dependent parameters. This work provides a foundation for adaptive protocol suites that will eventually enable an integrated context-aware communication paradigm.**

I. INTRODUCTION

Mobile ad hoc networks, often comprised of unreliable or mobile nodes, are promising for enabling applications when it is difficult or impossible to build or use infrastructure. In an ad hoc network, each node creates a network link in a self-organizing manner, forwarding data packets for other nodes in the network. Developing efficient routing protocols for such self-organizing networks continues to be a primary research challenge and has resulted in a plethora of available protocols with performance characteristics that vary widely as the operating environment changes.

Existing routing protocols are generally divided into two categories: proactive routing protocols, which maintain routes from every potential source to every potential destination [17], and reactive routing protocols, where routes between a source and a destination are set up on-demand [11], [18]. Several routing protocols have been proposed in each category, each with different properties [3], [6], [10], [19]. Our previous work [12] demonstrated that an optimal routing protocol can be selected for a particular application given specifications of physical characteristics of the network and application properties, the combination of which we refer to as a *network deployment*. Selecting the most appropriate routing protocol for a particular network deployment facilitates meeting application requirements for performance metrics such as average end-to-end delay and average throughput.

Making an informed suggestion to a deployer with respect to the appropriate underlying communication protocol requires expressive cost metrics that allow fast analytical evaluation of the expense of incorporating competing communication protocols. Previous work has almost exclusively used simulations to evaluate protocol performance, though some approaches have attempted real-world measurements [15]. Our previous work created a pre-design time model using extensive simulation results [12]. While these approaches can all be reasonably accurate, they require prohibitively long times and excessive resources to estimate protocol performance and are therefore difficult to incorporate into the type of instant-feedback system application developers really demand.

The work described herein continues our efforts toward analytically expressing protocol performance metrics in terms of environment-, protocol-, and application-dependent parameters. Specifically, we derive detailed models for two common and important protocol performance metrics: end-to-end delay and throughput. Both metrics are heavily influenced by both changing network connectivity and by characteristics of the communication protocol in use. Topological changes such as how long paths remain valid fall in the first category, while specific routing mechanisms such as how efficiently a protocol handles a failure fall in the second. By combining these two classes of characteristics, we are able to express how application-level performance metrics depend upon various aspects of the operating environment. This work is a first step in enabling a recommendation framework that takes as input specifications of applications' requirements and potential operating environments. The framework will ultimately select the communication protocol or protocols most suitable to a particular network deployment and potentially adjust the selected protocol at run-time as conditions change.

The remainder of the paper is organized as follows. Section II provides related work on analytical models in mobile ad hoc networks. In Section III, we present a detailed analytical model of the aforementioned routing protocol performance metrics and provide simulation results to validate our modeling approach in Section IV. Section V discusses issues and implications of our model and outlines future research directions. Finally, we summarize our conclusions in Section VI.

II. RELATED WORK

Most of the previous related work has presented simulation results to show routing protocols' behavior with varying parameters for either mobility or traffic [3], [6], [10], [13], [19]. Interestingly, the results showed a consistent pattern of protocols' performance metrics over the parameters. Drawing on the noted consistency of protocol behaviors, we propose a framework to build analytical models of protocol behavior using probability theory. Although simulation based approaches may be able to be more closely matched to a real world situation, they cannot be used to quickly provide information about protocols' potential behavior in arbitrary environments. This is due to the complexity of running the simulations or empirical evaluations on-demand and the large number of parameters that can be tweaked between any two evaluations. Therefore, a tool that provides instantaneous feedback to developers with regard to the impact of their implementation decisions on overall deployment performance requires expressive analytical models.

Comprehensive analytical modeling of mobile ad hoc networks that entail mobile nodes has been limited due to intrinsic difficulties arising from node mobility. Accounting for such dynamics requires modeling in continuously variable topologies. However, some research has been performed towards a theoretical analysis of mobility [5], [20], [23]. Each model starts by identifying specific assumptions for the mobility pattern, leading to a link availability analysis using the mobility model. Although it is not likely that a mobility pattern exactly matches a real-world movement scenario, it is meaningful to acquire statistical results from nodes' mobility in predicting an overall effect. In addition, some research tackled defining link availability with an assumption on a probability distribution [1], [9], [16]. The link availability analysis resulted in meaningful statistical parameters: link duration time and path duration time. Some of this work even showed applications using the parameters, such as a route cache expiration timer. We do not try to construct new mobility and link connectivity models and instead rely on this body of existing models as a foundation to construct models of application-level protocol performance. Moreover, in our framework, more realistic mobility and link availability models can be substituted, or the existing model can be updated to estimate a protocol's behavior more accurately.

While most previous analytical studies have focused on the quality of communication links of a node in a mobile ad hoc network, some work has gone a step further in deriving routing overhead analytically [7], [22], introducing a stochastic model for message delay of a routing protocol [8] or an analytical framework for routing delay, given an optimized route cache [14]. Our work builds on these approaches to create a framework for estimating a routing protocol's behavior, specifically the average end-to-end delay and average throughput, by making use of an analytical mobility model and parameters that affect the routing protocols' behaviors.

III. ANALYTICAL MODEL

We present our analytical model in this section. Network connectivity and changes in network connectivity have a significant impact on the performance of communication protocols in mobile ad hoc networks since communication relies on the formation and maintenance of paths through the network. In this section, we first create a generic system model that applies to wireless networks deployed in an ad hoc fashion. Because changes in the network topology can have such an important impact on protocol performance, we then show how a mobility model can be stated in terms of parameters that will impact the performance model we will derive. This latter model of behavior with respect to traditional performance metrics is the subject of the final subsection.

A. System Model

Before presenting our modeling approach, we state some assumptions and system descriptions that guide our derivation.

1) All links are bi-directional. All nodes can send and receive data, and, if a node can send data to another node, it also receives any messages that node sends.
2) A node create links with all neighbor nodes that are within the radio range R, and a link breaks when a node is separated from another node by a distance greater than the radio range. This is the unit disk model of connectivity and can be reasonably well provided by an omni-directional antenna. We assume that all nodes have an equal transmission power and that when a transmission reaches distance R from the transmitter, the signal abruptly attenuates.
3) A node can move in any direction uniformly distributed over $[0, 2\pi)$.
4) A node's speed, its direction of movement, and its location are mutually independent. Together with the previous assumption, this implies that node mobility is completely random. As a general case, it is reasonable to use a random model of mobility because all nodes are mutually independent. Although some nodes can move together for some purpose in a specific scenario, a random pattern can give a statistical intuition of mobility effects in an ad hoc network.
5) Nodes' initial positions are uniformly distributed over an area. This states that, with the effect of a wrap-around border behavior, nodes are distributed uniformly in the simulation space at any given time [2].[1]

The above assumptions are frequently made to theoretically analyze and simulate an mobile ad hoc network [20], [23].

B. Modeling Changing Connectivity

Our analytical model also assumes provisions within the communication layers that create and tear down links between a node and its one-hop neighbors, i.e., those nodes within the

---

[1] Since we do not suppose a specific mobility model, we also do not specify the area in which nodes can be located, which depends on the particular mobility model

communication radius R. To reach a node that is out of direct communication range, a node creates a route to the remote node using a sequence of hop-level links. A communication link is also associated with a node's mobility since movement changes the link-level connectivity of pairs of nodes, resulting in both broken routes and newly available routes. We model connectivity characteristics using the aforementioned radio range (R) and a mobility model.

Several mobility models have been proposed for evaluating the performance of MANET routing protocols [4]. Simulation shows that the mobility model profoundly effects a routing protocol's performance. It is, however, a challenging task to compose an analytical model of mobility patterns. In several instances, researchers make additional simplifying assumptions. Although some mobility patterns are assumed in our system model, the assumptions are not sufficient to completely describe the nodes' mobility. In this paper, we provide an example use of our analytical framework assuming a particular mobility model.

The contribution of this paper is not the elucidation of new mobility models or even the formalization of existing models. For this reason, to develop and demonstrate our behavior model, we use an existing mobility model and relations previously derived for that model. We are able to do so because our behavior models are not dependent on a particular mobility model, and the model we have currently selected could be exchanged, depending on the network deployment. As we will demonstrate, instead of relying on specifics of the style of mobility, our behavior models rely on parameters that can be derived for any number of different mobility models.

We use the straight line mobility model [23] as an example. In addition to the assumptions stated in the previous section, the mobility model assumes that a node's velocity is uniformly distributed over [0, v_{max}], and the area where the nodes move is torus-shaped. From the description of the mobility pattern, a probability density function (PDF) of relative velocity, $f_{v_r}(v; v_{max})$, can be acquired, where v_{max} is the maximum possible velocity of each node. This PDF of relative velocity provides statistical information about how long it takes a node to escape the communication range R of another node.

Using this PDF of relative velocity and radio range, a PDF of *link duration time*, a measure of the longevity of a single link in the straight line mobility model, can be derived. We refer to this distribution as $f_{T_{ld}}(t; R, v_{max})$, though detailed coverage of the distribution's derivation is beyond the scope of this paper; the interested reader is referred to the original derivation [23]. Given this link duration time and considering the fact that a path duration time should be the minimum of the duration times of the links in the path, one can derive the following relation:

$$P(T_{pd} > t) = \prod_{i=1}^{k} P(T_{ld} > t). \tag{1}$$

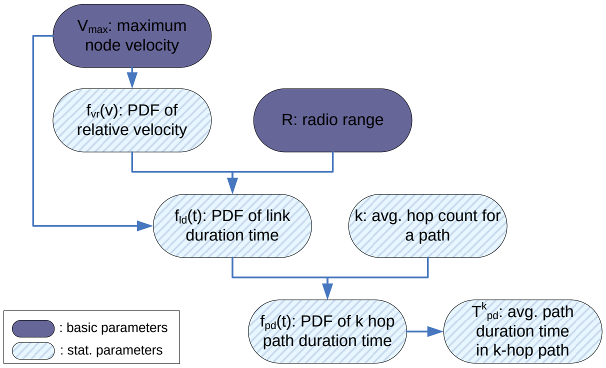

Fig. 1. Connectivity model parameters in the straight-line mobility model

From the definition of a cumulative probability function (CDF), we can express the CDF of a k-hop path duration time $F_{pd}(t; k)$ as:

$$1 - F_{T_{pd}}(t; k) = \prod_{i=1}^{k} \{1 - F^i_{T_{ld}}(t)\} = \{1 - F_{T_{ld}}(t)\}^k \,, \tag{2}$$

where $F^i_{T_{ld}}(t)$ is the CDF of the link duration time of the i^{th} link in the path. However, given the above assumptions, the link duration time does not depend on the ordering of links in the path. It follows from the definition of the CDF that the PDF of path duration for a path consisting of k links, $f_{T_{pd}}(t; k)$ is:

$$f_{T_{pd}}(t; k) = k \cdot \left[1 - \int_0^t f_{T_{ld}}(x)dx\right]^{k-1} \cdot f_{T_{ld}}(t). \tag{3}$$

Finally, the average path duration time for a path of length k hops can be expressed as:

$$\overline{T_{T_{pd}}} = \int_0^\infty t \cdot f_{T_{pd}}(t; k)dt. \tag{4}$$

This path duration time expression is the component necessary to describe the impact of a mobility model on the protocol performance metrics characterized in the next section. This section has briefly summarized a path duration time derivation done previously for the straight-line mobility model [23]; to use the approach in the next section to characterize protocol behavior under a different mobility model requires a similar derivation of path duration time for that particular mobility model. Figure 1 shows the extracted parameters for a connectivity model; using the basic parameters an application deployer provides (e.g., a node's maximum velocity, v_{max}), useful statistical parameters (e.g., an expression of the link duration time, $f_{T_{ld}}(t)$) can be derived.

C. *Modeling Performance*

In the remainder of this section, we derive analytical expressions for protocol performance in terms of two application-level metrics: delay and throughput. These metrics are expressed in terms of parameters influenced by both the mobility model as derived in the previous section and by characteristics of the communication protocol as discussed in this section.

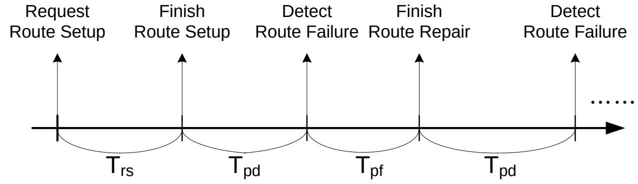

Fig. 2. Communication time line consisting of a route setup, then alternating path durations, path repairs

1) Communication Sessions: We rely on the notion of a "communication session," or the sequence of exchanges between a source and destination node in the network. This models the high-level fact that, to support real applications, nodes do not often exchange single messages but instead engage in conversations of a longer duration. Specifically, a communication between a source and destination node follows a sequence consisting of a protocol to initiate the network path used for communication and a protocol to maintain that path in the face of topology changes. This pattern can be characterized by the following regular expression:

$$T_{rs}(T_{pd}T_{pf})^{+}T_{f} \tag{5}$$

where T_{rs} represents the time to set up a route, T_{pd} is the path duration time derived in the previous section, and T_{pf} is the path failure time due to a broken route. When a route break happens after T_{pd}, the routing protocol recovers a route during T_{pf}. Both T_{rs} and T_{pf} are dependent on the nature of the routing protocol. T_f captures the fact that a communication session does not likely end with a route repair; most likely the source finishes the communication task and cuts off the communication before a new route repair is required. That is, $T_f \leq T_{pd}$; T_f is the remaining communication time required once the communication is cut into chunks that take time T_{pd}. In the remainder of this section, we ignore this communication tail, as we assume that communication sessions are relatively lengthy and entail a significant number of $(T_{pd}T_{pf})$ pairs to complete the transmission.

This concept of communication session relies on the notion that the only time useful communication can occur is after the path has been created, while it remains valid (i.e., T_{pd}). If the path breaks, useful work can only be resumed once the path has been repaired (which may or may not require a much time as the initial setup of the route). As such, the time line for a communication session consists of a single route setup followed by repeated, alternating path durations and path repairs. Figure 2 depicts such a communication session.

2) End-to-end Delay: Having established the notion of a communication session, we build models of two application-level performance metrics: the end-to-end delay experienced by a data packet in the communication session, and the overall throughput of the session. We start with the derivation for an expression of the end-to-end delay.

Depending on the mobility model and the rate of change within that model, a data packet sent by a source node may encounter link failures along its delivery path. These failures may be due to mobility or to the intrinsic unreliability of mobile nodes on the route. The end-to-end delay between a source and destination, T_{DLY} can be abstractly characterized as:

$$T_{DLY} = (Time \text{ spent performing route repairs}) + (Time \text{ spent delivering a data packet to a destination}).$$

The average time required for route repairs can be derived using the average number of route failures and the mean time to repair a single route. Since a packet is likely to encounter more route failures as the route length grows, the mean count of route failures is proportional to the length of the route. The time to repair a single route is a protocol-dependent parameter that depends on the path length and the amount of the path that has been traversed; a route failure happening near the source may be fixed more quickly in some protocols because the source can be more quickly notified, allowing the source to more quickly find the alternative path to the destination. The route repair mechanism for each routing protocol is unique. In some routing protocols, a route failure near a destination node may also be recovered in a short amount of time using a caching strategy. The implications of these protocol differences are revisited in more detail in Section V.

Based on the above, we can express the expected end-to-end delay for a route with k hops as:

$$\overline{T_{DLY}^{k}} = \overline{N_{rf}^{k}} \cdot \overline{T_{rr}^{k}} + k \cdot \overline{T_{1h}^{data}}, \tag{6}$$

where N_{rf}^{k} is the number of route failures a packet experiences in a k hop route, T_{rr}^{k} is the route repair time for a k hop route, and T_{1h}^{data} is the one hop delivery time of a data packet. Of these three values, T_{1h}^{data} is the most straightforward; it can be calculated given the size of a data packet and the data rate of the radio. The other two values, N_{rf}^{k} and T_{rr}^{k}, depend on the mobility model and routing protocol, respectively. As a packet traverses a multihop path to a destination, it may encounter a broken link. Once the path is repaired, the packet resumes its journey, but it may encounter additional broken links in the process. Given the average length of a path (k), one can therefore express the average number of such route failures a packet will encounter. This is N_{rf}^{k}.

We can express N_{rf}^{k} by first defining the probability, $p_{s,k}$, that a packet arrives at its final destination without encountering a link failure. N_{rf}^{k} is then:

$$\overline{N_{rf}^{k}} = \sum_{n=1}^{\infty} n(1-p_{s,k})^{n} p_{s,k} = \frac{1-p_{s,k}}{p_{s,k}}. \tag{7}$$

Again, a packet can be successfully delivered over a k hop path when the packet is relayed successfully over each link in the path. Therefore, $p_{s,k}$ can be calculated based on the probability of successful delivery over a one hop link, p_l [2]:

$$p_{s,k} = (p_l)^{k}. \tag{8}$$

---

[2] Since we assume an independent node's mobility in a system model, each link is considered independent

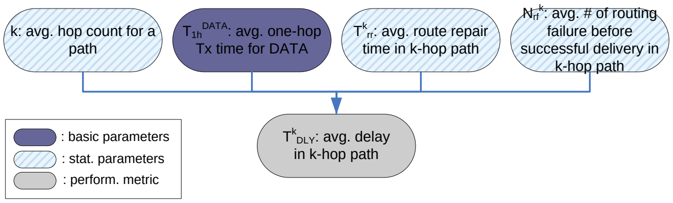

Fig. 3. Performance model parameters: end-to-end delay

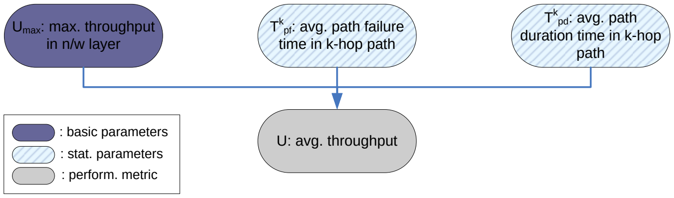

Fig. 4. Performance model parameters: throughput

In fact, p_l depends on the mobility model in question; it expresses the possibility that a data packet of a given size can be transferred during the expected link duration time. Hence, p_l can be acquired from a conditional probability, $P(S \mid T_{ld} = t)$, that a packet can be delivered successfully over a link when the link duration time is t. This expression also requires a statement of the PDF of link duration time:

$$\begin{aligned} p_l &= \int_0^\infty P(TX\_Success|T_{ld} = t) \cdot f_{T_{ld}}(t)dt \\ &= \int_{T_{1h}^{data}}^\infty \left(\frac{t - T_{1h}^{data}}{t} \right) \cdot f_{T_{ld}}(t)dt. \end{aligned} \tag{9}$$

In our expression of the total delay a packet experiences, the average number of route failures is multiplied by the average amount of time it takes for a protocol to recover from a route failure. This route repair time, T^k_{rr}, depends largely on the nature of the particular routing protocol in use. Different routing protocols handle route failures in different ways; for example, some protocols give up and require a new route discovery to be initiated, while others cache network topology information and attempt to reconstruct a route on the spot. Figure 3 depicts the delay model with parameters we derived. In the model we present here, we leave these expressions in terms of such protocol dependent parameters; Section V discusses our next steps to provide instantiations of the model for specific routing protocols.

3) Throughput: From an application's perspective, throughput is a measure of how much useful (application-level) data is sent over a connection in a unit of time. We measure throughput as a ratio of a given total amount of data to the amount of time it takes to transfer that data; this expresses a particular network deployment's maximum achievable throughput.

Let the total data to be transferred be D_{total} and the time duration needed to deliver that data be T_{total}. We assume that the application layer pushes data to the network layer and that there is no loss in the buffer of the network layer (e.g., data loss due to buffer overflow). In addition, it is assumed that each link can provide the same bandwidth and that there is no queuing delay at the intermediate nodes. In this situation, once the application generates D_{total}, data is always ready to be sent by the source node until all of the data has been sent. Once the path from a source to a destination is available, the data starts to be transferred during the path duration time at the best throughput the wireless links can support, expressed as U_{max}. On average, the amount of data that can be transmitted in a time interval $(T_{pd}\ T_{pf})$ is $U_{max} \times \overline{T_{pd}}$. Therefore, to send all of the data (D_{total}), we require $D_{total}/(U_{max} \times \overline{T_{pd}})$ such time intervals. Hence, the time needed to completely transmit D_{total} can be expressed as:

$$T_{total} = \overline{T_{rs}} + \frac{D_{total}}{U_{max} \cdot \overline{T_{pd}}} \cdot (\overline{T_{pd}} + \overline{T_{pf}}), \tag{10}$$

where, $\overline{T_{rs}}$ is the average route setup time. Therefore, throughput, U, can be derived as:

$$U = \frac{D_{total}}{T_{total}} = \frac{U_{max}}{1 + \frac{\overline{T_{pf}}}{\overline{T_{pd}}}} \cdot \left(1 - \frac{\overline{T_{rs}}}{T_{total}} \right) \tag{11}$$

During a long enough communication with a large enough D_{total}, the throughput ceases to be affected by the initial route setup. Referring back to the communication time line, this means the total time spent for delivering the data is much greater than the average route setup time, dampening the effect of the latter (i.e., $\overline{T_{rs}} \ll T_{total}$). In this case, since we assume long-lived conversations, we can assume $\frac{\overline{T_{rs}}}{T_{total}}$ is close to 0, and the throughput expression can then be approximated as:

$$U \approx \frac{U_{max}}{1 + \frac{\overline{T_{pf}}}{\overline{T_{pd}}}}. \tag{12}$$

Figure 4 shows the extracted parameters for a throughput estimation using our model. Equation 12 essentially states that we can achieve a high throughput when a broken path is repaired in a short time or when there is a very long path duration time. Interestingly, our result for throughput is similar to the result in [1].

IV. SIMULATION RESULTS

In this section, we compare results of our analytical model to simulation results to determine the quality of our model. The simulations were performed in ns-2 [21] using AODV [18] as the routing protocol. While utilizing a straight line mobility model to abstract a changing connectivity in Section III, we applied the random waypoint mobility model to the simulations since the latter is a well-known and widely available mobility model for MANET simulations. The simulation was performed over 10 randomly generated scenarios and we took an overall average of each measured value. Detailed simulation parameters are described in Table I.

Varying the possible maximum speed of mobile hosts, we observed several of our analytical model's parameters directly to enable evaluation of Equations 6 and 12. Specifically, for our delay model, we measured four parameters: the average number of route failures per a packet ($\overline{N^k_{rf}}$), the average

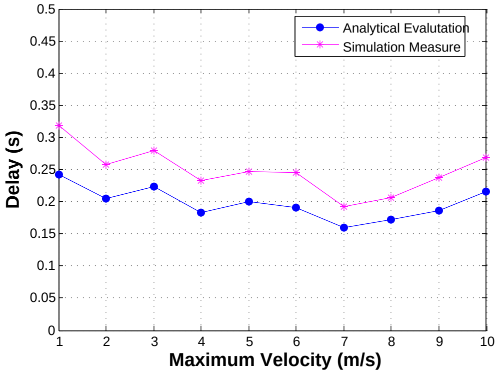

Fig. 5. Comparison of analytical model and simulation: Delay

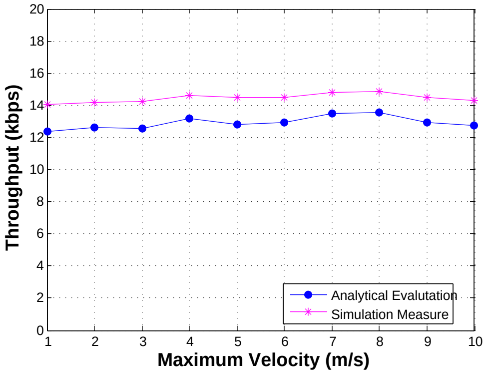

Fig. 6. Comparison of analytical model and simulation: Throughput

TABLE I
SIMULATION PARAMETERS

Parameters	Value
Network Size	1500 m x 300 m
Simulation Time	900 $sec.$
Number of Mobile Hosts	40
Traffic Type	Constant Bit Rate
Packet Size	512 $bytes$
Packet Transmission Rate	4 $packets/sec.$
Number of Traffic Sources	15
Movement Model	Random Waypoint Model
Mobile Hosts Max. Speed	1, 2, $\cdots$, 9, 10 $m/sec.$
Pause Time	0 $sec.$

route repair time (T^k_{rr}), the average hop count (k), and the average data packet transmission time for one hop ($\overline{T^{data}_{1h}}$). The average end-to-end delay was calculated using Equation 6 and was compared with an average end-to-end delay measured directly from simulation. Future work will focus on how to generate the model's parameters given specifications of an applications' intended operating environment. Figure 5 depicts a comparison between the results for our analytical evaluations and simulation for the average end-to-end delay. Although the analytical model gives an approximately 20% smaller value than the simulation study, the trends between the two match exactly. The reason for lower values from the analytical model is because our model does not consider nodal delays such as queueing delays.

In our model of throughput, Equation 12 requires both an average path duration time and an average path fail time as inputs. However, it is difficult to identify when a path is broken and is recovered for a particular source and destination pair. Instead, we measured the time during which packets with the same itinerary arrive at the destination and the interval between their arrivals. A packet should arrive at the destination with a different itinerary only after a path recovery has succeeded. We therefore consider the duration described above to be an approximation of the path duration and the measured interval between packets with different itineraries to be an approximation of the route recovery time. In Fig. 6, we demonstrate the validity of our analytical model of throughput. The throughput is less in the analytical model because the model is conservative in the fact that it considers an entire path to be either connected or disconnected. This does not allow the model to capture the optimistic buffering that occurs at intermediate nodes in the actual protocol implementation.

V. DISCUSSION AND FUTURE WORK

The model derived in the previous section gives expressions for application-level performance metrics in terms of characteristics of the operating environment and of the communication protocols themselves. For a protocol recommendation framework to effectively use the expressions we have derived, the operating environment characteristics must be provided by the application deployer. In our model, these are currently expressed at a relatively low level (e.g., T_{pd} represents path duration time). While a developer will not directly provide these low-level parameters, they can be derived from specifications of the mobility model or the topological properties of the network in which the application will be deployed. Other work investigates the relationship between these topological and mobility models and these low-level parameters [5], [20], [23]; we build on these successful efforts. Therefore, to input information about the environment, the developer must provide as input a model of the network topology and the potential mobility within that topology.

The model from the previous section also relies on specifications of protocols' behaviors. Different protocols incur different amounts of overhead or take different amounts of time to accomplish similar tasks. In our model, we currently express the unique protocol characteristics in terms of variables such as T_{rs}, the time to initialize a route, or T_{rr}, the time to completely repair a broken route. In fact, T_{rr} could be affected by route caching schemes (e.g., aggressive caching), route recovery strategies (e.g., expanded ring search), the degree of mobility, etc. In addition, the nodes' initial distribution and the distance between a given source and destination impacts T_{rs}. Using these high-level expressions of protocol behavior simplifies the presentation of the preceding model. Future work will look at specific routing protocols such as DSR [11], AODV [18], and others and demonstrate how to generically derive these parameters for the different protocols.

The next major step in this effort is to incorporate the performance models described in this paper into a recommendation framework for application deployers. Given good expressions for the two types of parameters (network configuration parameters and protocol parameters), our performance models will be able to communicate to an application deployer the impact that changing both the intended network topology and the underlying communication protocol will have on the application's overall performance. Future work will include building out this framework. Further efforts will also include developing similar models for other application performance metrics in addition to delay and throughput.

VI. CONCLUSION

We have introduced the underpinnings of a software design, development, and deployment tool enabling specifications of operating environments to influence the way in which interactive behaviors are implemented. We separate the implementation of an application's interactive portions from that of the communication infrastructure required to support interactions, automatically generating a set of potential concrete implementations and their associated costs.

The primary contributions we presented in this paper are derivations of detailed models for two important protocol performance metrics: end-to-end delay and throughput. These models address the challenge of generating expressive cost metrics allowing fast analytical evaluation of the expense incorporating competing communication protocols. Prior efforts at evaluating protocol performance for network deployments have been unacceptably burdensome, requiring excessive computing time and resources to produce useful results.

Our performance metric models are expressed in terms of environment-, protocol-, and application-dependent parameters. Our framework can then use these models to inform a software designer of potential impact on overall application performance when implementing various possible routing protocols under specified environmental conditions. The eventual goal for the models presented herein is that of enabling real-time protocol adaptation to continually provide the best possible performance, even under changing environmental conditions. We are currently engaged in work to generically derive parameters for specific routing protocols, and the realization of a deployment framework as a useful, effective design tool.

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] F. Bai, N. Sadagopan, B. Krishnamachari, and A. Helmy. Modeling path duration distributions in manets and their impact on reactive routing protocols. *IEEE Journal on Selected Areas in Communications*, 22(7):1357–1373, September 2004.

[2] C. Bettstetter. Mobility modeling in wireless networks: categorization, smooth movement, and border effects. *SIGMOBILE Mobile Computing and Communications Review*, 5(3):55–66, 2001.

[3] 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 *Proc. of the 4^{th} Annual ACM/IEEE Int'l. Conf. on Mobile Computing and Networking*, pages 85–97, 1998.

[4] T. Camp, J. Boleng, and V. Davies. A survey of mobility models for ad hoc network research. *Wireless Communications & Mobile Computing: Special issue on Mobile Ad Hoc Networking: Research, Trends and Applications*, 2(5):483–502, 2002.

[5] S. Cho and J. Hayes. Impact of mobility on connection stability in ad hoc networks. In *Proc. of the 2005 IEEE Wireless Communications and Networking Conf.*, volume 3, pages 1650–1656, March 2005.

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

[7] R. Duggirala, R. Gupta, Q.-A. Zeng, and D. P. Agrawal. Performance enhancements of ad hoc networks with localized route repair. *IEEE Trans. on Computers*, 52(7):854–861, 2003.

[8] R. Groenevelt, P. Nain, and G. Koole. The message delay in mobile ad hoc networks. *Performance Evaluation*, 62(1-4):210–228, October 2005.

[9] S. Jiang, D. He, and J. Rao. A prediction-based link availability estimation for routing metrics in manets. *IEEE/ACM Trans. on Networking*, 13(6):1302–1312, December 2005.

[10] P. Johansson, T. Larsson, N. Hedman, B. Mielczarek, and M. Degermark. Scenario-based performance analysis of routing protocols for mobile ad-hoc networks. In *Proc. of the 5^{th} Annual ACM/IEEE Int'l. Conf. on Mobile Computing and Networking*, pages 195–206, August 1999.

[11] D. Johnson, D. Maltz, and J. Broch. DSR: The dynamic source routing protocol for multihop wireless ad hoc networks. In *Ad Hoc Networking*, pages 139–172, 2001.

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

[13] S. Lee., J. Hsu, R. Hayashida, M. Gerla, and R. Bagrodia. Selecting a routing strategy for your ad hoc network. *Computer Communications*, 26(7):723–733, May 2003.

[14] B. Liang and Z. Haas. Optimizing route cache lifetime in ad hoc networks. In *Proc. of the 22^{nd} Annual Joint Conf. of the IEEE Computer and Communications Societies*, volume 1, pages 281–291, March-April 2003.

[15] H. Lundgren, D. Lundberg, J. Nielsen, E. Nordstrom, and C. Nordstrom. A large-scale testbed for reproducible ad hoc protocol evaluations. In *Proc. of the 2002 IEEE Wireless Communications and Networking Conf.*, pages 412–418, March 2002.

[16] A. McDonald and T. Znati. A path availability model for wireless ad-hoc networks. In *Proc. of the 1999 IEEE Wireless Communications and Networking Conf.*, volume 1, pages 35–40, 1999.

[17] 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.

[18] C. Perkins and E. Royer. Ad-hoc on-demand distance vector routing. In *Proc. of the 2^{nd} IEEE Wkshp. on Mobile Computer Systems and Applications*, pages 90–110, 1999.

[19] E. Royer and C. K. Toh. A review of current routing protocols for ad hoc mobile wireless networks. *IEEE Personal Communications*, 6(2):46–55, April 1999.

[20] P. Samar and S. B. Wicker. On the behavior of communication links of a node in a multi-hop mobile environment. In *Proc. of the 5^{th} ACM Int'l. Symp. on Mobile Ad Hoc Networking and Computing*, pages 145–156, 2004.

[21] VINT Project. *The network simulator - ns-2, http://www.isi.edu/nsnam /ns/*, 2002.

[22] D. Yu and H. Li. A model for performance analysis of mobile ad-hoc networks. *Proc. Int'l. Conf. on Telecommunications*, June 2002.

[23] D. Yu, H. Li, and I. Gruber. Path availability in ad hoc network. In *Proc. of the 10^{th} Int'l. Conf. on Telecommunications*, volume 1, pages 383–387, February-March 2003.