Context-Aware Protocol Selection for Routing in Mobile Ad Hoc Networks

Taesoo Jun
Christine Julien

TR-UTEDGE-2006-008

Context-Aware Protocol Selection for Routing 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

Many protocols exist for supporting routing in mobile ad hoc networks (MANETs). Selecting a particular protocol for an application or deployment environment involves evaluating many complex inter-dependent tradeoffs and can be an overwhelming task for an application designer. However, this decision can have a significant impact on the success of a system in terms of performance, cost, and responsiveness. This paper introduces a design tool that automates this evaluation process by controlling for environmental and usage properties of an intended deployment. This provides the foundation for a highly adaptive protocol suite that leverages the relative benefits of competing communication paradigms.

Categories and Subject Descriptors

A.26 [**Mobile Computing and Applications**]: Miscellaneous; D.2.8 [**Wireless Mobile Ad Hoc Network**]: Routings—*adaptive routing, performance measures*

General Terms

MANET

Keywords

Routing Protocol Selection, Protocol Model, Target Deployment

1. INTRODUCTION

Mobile ad hoc networks (MANETs) are self-organizing networks in which each node establishes communication links without the help of an infrastructure. In MANETs, each node acts as a router by forwarding data packets for other nodes. One of the main issues in the design of MANETs is the development of routing protocols that can efficiently find routes from a source node to a specified destination node.

SAC'07 March 11-15, 2007, Seoul, Korea

Existing routing protocols can be divided into two categories: proactive routing protocols which maintain routes between every pair of nodes and reactive routing protocols where routes between nodes are created on-demand. Several protocols exist in each category, each with different characteristics. The selection of the most appropriate protocol depends on the environment (e.g., how fast the network topology changes) and application requirements (e.g., how latency-bound the applications are). In general, previous work has shown that several factors, including the degree of mobility and the types of traffic applications generate influence the performance of routing protocols in MANETs [4, 5, 8, 13]. Moreover, optimizing for metrics such as the average end-to-end delay and the average throughput is crucial to meeting application requirements from the perspective of the network layer. Marrying these application requirements to the environmental and network factors can help select routing protocols that best achieve application goals.

Our goal in this paper is to generate a tool that uses characteristics of the environment and application requirements to select the protocol most appropriate for a target network deployment. We define *network deployment* to be the combination of the network's physical characteristics (e.g., mobility degree, density, error rate, etc.) and the characteristics of the applications deployed in the network (e.g., rate of traffic generated, number of communication endpoints, application goals and requirements, etc.). Using our tool, a software designer provides parameters defining the operational environment and applications' requirements. The tool then analyzes these parameters and compares them to internalized models of protocols' behavior under different conditions. These models can be created from simulation studies or real world experiments, and the tool allows models of new protocols to be inserted or existing models to be updated. These models allow the tool to evaluate the designer's inputs and select the best protocol matching his or her stated parameters.

While previous papers have presented the characteristics of different routing protocols and have made comparisons among them [5, 8, 13, 24], this paper describes a first effort at automating the analytical selection of the routing protocol best suitable to a particular deployment. Our tool incorporates specifications of the network's applications' requirements and preferences to influence the protocol selection process. Ultimately, this work hopes to lead to a model in which the routing protocol can be adjusted at run-time to adapt to changing applications and network conditions.

The paper is organized as follows. Section 2 describes our

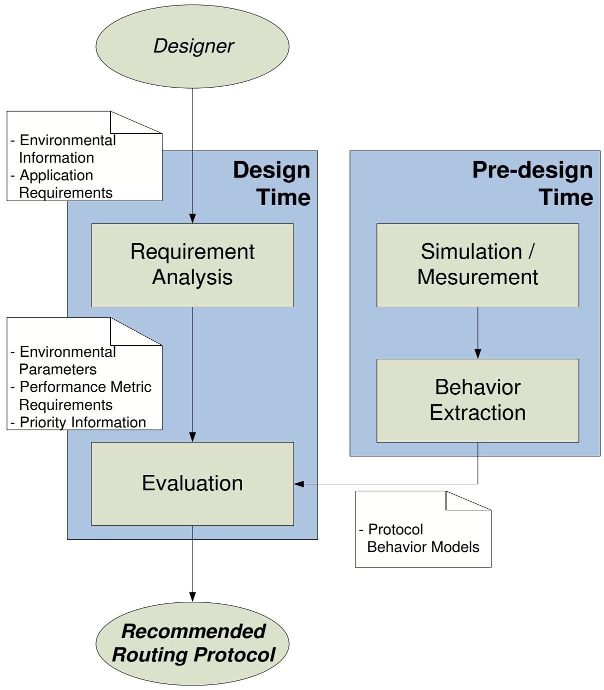


Figure 1: Routing Protocol Selection Process

novel decision process and how we model protocol behavior. Section 3 provides a brief background of the three routing protocols we used to evaluate our tool. Section 4 presents a motivating scenario and applies the process described in Section 2 to the scenario. We summarize related work in Section 5 and conclude in Section 6.

2. PROTOCOL SELECTION PROCESS

To fulfill applications' requirements in MANETs, it is important to match an appropriate routing service to the network deployment [16]. To that end, we have defined a process that uses models of protocols' characteristics, the target network's characteristics, and requirements of applications expected in that network to analytically determine the optimal routing approach before deployment. In this section, we describe this protocol selection process, which is depicted in Figure 1. The process takes two types of inputs (environmental information, and application requirements) and returns the protocol most appropriate for the stated conditions. In pre-design time, as shown on the right of Figure 1, the process builds *protocol behavior models* by incorporating results from simulation and/or real world measurement, followed by a behavior extraction process. Because the acquisition of the model information takes a significant amount of time, the models serve as *input* to the design-time process, which is shown to the left of Figure 1. In design-time, the process uses the protocol behavior models to recommend a routing protocol through evaluation of expected applications' requirements and network deployment. To support general MANETs, multiple applications are assumed to run in a single network, and the differing (sometimes even competing) application requirements must also be considered.

2.1 Simulation and Measurement

The first step in generating the protocol behavior models, simulation and measurement, gathers information about performance characteristics of candidate routing protocols. This is a complicated undertaking since a protocol's performance depends on various parameters of a scenario. In Section 4, to provide an example of the process at work, we have run simulations for a variety of scenarios. These simulations are not exhaustive and are simply used to demonstrate the process of building protocol behavior models. When using our process, protocol developers must provide adequate simulations or other measurements for their protocols.

Each scenario that helps generate a protocol behavior model is affected by several parameters that fall in two categories: topological parameters and traffic parameters. The former indicate influences that nodes' movements in a topology have on a protocol's performance, and the latter represent effects by changing data traffic. The topological parameters are highly correlated with the particular underlying movement model because each movement model has radically different characteristics. For example, in the case of the random waypoint mobility model [14], a node's *pause time*, a node's maximum possible speed, and the node density are used as topological parameters. The pause time is the time for which a node pauses before moving toward a new destination, and the average node density is derived from the number of total nodes and the rectangular size of the area. As an another instance, for the reference point group mobility model [11], topological parameters are the number of groups, the number of nodes in each group, the speed and angle deviation for a random motion vector, and the checkpoint trace of the group leaders. The second category of factors, the traffic parameters, consists of a data traffic type, an average data payload size, a data sending rate, and a source node density, which is defined as the number of source nodes among total nodes. When building the protocol behavior models, the broader the set of data available to define the models, the more representative they will be.

To move towards defining a protocol's behavior model, after isolating parameters of the various topological and traffic models, we need to define the performance metrics used to measure different aspects of the protocol. The average throughput, packet delivery ratio, and average end-to-end delay are important metrics for best-effort networks. Other performance metrics such as an average hop count or average routing byte overhead can also be important. For any particular application, the most important metric or metrics for evaluating an underlying communication protocol are defined by factors specific to that application. To generate complete protocol behavior models, these scenario parameters must be combined with several simulations or real world experiments. This step produces the data that represents performance characteristics of protocols for a variety of conditions and performance metrics and is used in the next step, to extract behavior models.

2.2 Behavior Extraction

Based on the results of the simulation and measurement step above, we can create internalized models of protocols' behavior. From several simulation results, the tendencies of the identified performance metrics are estimated with respect to particular scenario parameters. To generalize tendencies, we use the least square data fitting method [9] to extrapolate a model from the results of the simulation and measurement step for a particular protocol. However, since a performance metric can be dependent on more than one independent scenario parameter, we use a multiple regression [19]. In the multiple regression, results from the simulation and measurement step are taken as inputs, and the least square fit model of the data is acquired. Namely, the multiple regression generates complete protocol behavior models with respect to intended scenario parameters. As a result, the models are given as functions with inputs of scenario parameters and outputs of performance metrics. For example, one behavior model of a single protocol could present an average throughput with respect to a mobility degree (measured, for example, as a pause time in the random waypoint mobility model) and be constructed from the results of several simulations where average throughputs are measured for different mobility degrees. Then, for a given mobility degree, an average throughput for a particular situation can be retrieved from the model.

2.3 Requirement Analysis

Given protocol behavior models generated in pre-design above, it is necessary to collect environment and application information at design time because the information will be different for each combination of network deployment and expected applications. The environmental information defines in which situation the applications will operate, and the application requirements dictate what conditions should be satisfied in order to achieve the goals of the applications. Each application goal maps to one or more performance metrics. For example, a goal of real-time voice communication might map to a minimum bandwidth required and a maximum delay tolerated.

The requirement analysis step shown in Figure 1 extracts information with which protocols are evaluated in the evaluation step with the protocol behavior model from the environment information and application requirements. In order to acquire user requirements, our tool allows a designer to specify the particular requirements of one or more applications intended for deployment in the network. These map to specific network performance metrics. The tool also gathers information about the target environment, including the relative speeds of nodes, network density, traffic patterns, and the mobility model that is the closest match for movement in that environment (e.g., a highway mobility model for automobile networks or other domain-specific or generic mobility models). The tool forces the user to choose from among the mobility models for which results are available. Finally, the designer also uses the tool to specify the relative (weighted) importance of the applications, providing priorities with which application requirements should be considered. We omit a detailed description of the tool's interface and instead focus in this paper on the tool's inner-workings. The environmental parameters the designer provides may map to scenarios that have been simulated or measured of the protocol behavior models or may require some interpretation or extrapolation. For example, density specifications can be accommodated by combining the number of nodes with the physical area of a given simulation.

The metrics specified by the designer for each application indicate the performances required to achieve the applications' goals. Such performance metrics are indicative of the quality of service a particular application expects from the underlying communication protocols. Take, for instance, an application that sends urgent messages that should be delivered within 5 seconds and with very high reliability over 90%. In this example, the performance metric requirements for the application are a maximum end-to-end routing delay of 5 seconds and a minimum packet delivery ratio of 90%. In the end, the designer specifies a set of such performance requirements for each application expected to be deployed in the network.

Table 1: Priority Information Table

Weight Factor	Applications			
	app.1	app.2	$\cdots$	app.j
application	w_1^{app}	w_2^{app}	$\cdots$	w_j^{app}
performance metric 1	w_{11}^{pm}	w_{12}^{pm}	$\cdots$	w_{1j}^{pm}
performance metric 2	w_{21}^{pm}	w_{22}^{pm}	$\cdots$	w_{2j}^{pm}
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
performance metric i	w_{i1}^{pm}	w_{i2}^{pm}	$\cdots$	w_{ij}^{pm}

Finally, in many cases, a user wants to consider a particular application to be more important than others. In such cases, the more important application's performance metric should be weighted more significantly with respect to other less important applications. To represent this, we associate priority information with both individual applications and performance metrics selected for each application. Each priority weight factor reflects the importance of each element (i.e., an application or a performance metric). The weight factor is assumed to have a value from 1 (least important) to 10 (most important). For applications, the application weights are given by the designer according to the relative significance of applications, and for each application, the performance metric weights are assigned considering which metric is critical to achieve the particular application's goals. Table 1 shows the priority information table in general form; as shown the priority information reduces to a simple weight applied to each of the metrics considered for each application. The priority information represented in Table 1 will be used in Section 2.4. An application may not have requirements for some performance metrics; in such cases, the application's weights for those metrics are 0.

2.4 Evaluation

The final step in Figure 1, evaluation, determines the most appropriate routing protocol given the protocol behavior models as input and the results from the requirement analysis step. For each protocol, we compare its compatibility to the performance metric requirements. The protocol behavior models evaluated on performance metrics and the scenario parameters produce the expected values of the pertinent metrics, and these values are compared with the performance metric requirements. To generate a quantitative decision, we introduce a preference value; the value states how favorable a particular protocol is with respect to the performance metric. Since smaller preference values are more favorable, the preference value is defined in two ways, depending on whether the performance metric is maximizing

Table 2: Preference Value Table

Performance Metric	Applications			
	app.1	app.2	$\cdots$	app.j
metric 1	p^k_{11}	p^k_{12}	$\cdots$	p^k_{1j}
metric 2	p^k_{21}	p^k_{22}	$\cdots$	p^k_{2j}
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
metric i	p^k_{i1}	p^k_{i2}	$\cdots$	p^k_{ij}
Sum	S^k_1	S^k_2	$\cdots$	S^k_j

or minimizing:

$$p^k_{ij} = \begin{cases} \frac{|s^k_{ij} - r_{ij}|}{r_{ij}} & s^k_{ij} < r_{ij}(if\ r_{ij}\ is\ min.req.),\ or \\ & s^k_{ij} > r_{ij}(if\ r_{ij}\ is\ max.req.) \\ 0 & otherwise,\ or\ r_{ij}\ is\ not\ defined \end{cases} \tag{1}$$

where, with respect to the *k-th* routing protocol, p^k_{ij} is the preference value for the *j-th* application and the *i-th* performance metric, and similarly s^k_{ij}, r_{ij} are the simulated value for the *k-th* routing protocol and the application requirement respectively. A preference value of 0 indicates that the *k-th* routing protocol is expected to exactly fulfil the application requirement for the *j-th* application and the *i-th* performance metric; if r_{ij} is not defined, the preference value is defined to be 0.

When the preference values for each application are acquired, our tool constructs a preference value table such as Table 2 for each target protocol. The priority information is then applied to preference values to consider the relative importance among performance metrics and applications. For each application, the performance metric weight factor from Table 1 is applied to Table 2, and the weighted sums that result have the following form:

$$S^k_j = P^k_j \times W^{pm}_j \tag{2}$$

where, P^k_j is $[\ p^k_{1j}\ \ p^k_{2j}\ \ \cdots\ \ p^k_{ij}\ ]$, and W^{pm}_j is defined as the performance metric weight factors for the *j-th* application, which is $[\ w^{pm}_{1j}\ \ w^{pm}_{2j}\ \ \cdots\ \ w^{pm}_{ij}\ ]^T$ from Table 1. Then, to differentiate the importance among applications, the application weight factor is applied to a vector of Equation 2. Equation 3 shows the total weighted sums of the preference values with respect to the *k-th* routing protocol.

$$S^k_{total} = S^k \times W^{app} \tag{3}$$

where, S^k is $[\ S^k_1\ \ S^k_2\ \ \cdots\ \ S^k_j\ ]$, and W^{app} is defined as the application weight factors, represented as $[\ w^{app}_1\ \ w^{app}_2\ \ \cdots\ \ w^{app}_j\ ]^T$ (the first row of Table 1).

For each routing protocol, the total weighted sum is acquired, and a routing protocol that has the smallest total weighted sum is determined as more favorable with respect to satisfying the applications' requirements.

3. ROUTING PROTOCOLS REVIEW

To provide a variety of models for the initial evaluation of our tool, we modeled the behavior of three protocols. For completeness, this section provides a brief overview of the three protocols we chose to use as samples. Our tool is completely independent of the protocols, and models of other protocols' behaviors (or models updated from those we used) can be swapped into the tool.

3.1 AODV

Ad-hoc On-demand Distance Vector routing (AODV) [23] is an example of a reactive routing protocol, or a protocol that creates routes on-demand. In AODV, when a node S needs a route to a node D, S broadcasts a ROUTE REQUEST which is flooded in a controlled manner until the packet reaches D or a node that has a fresh route to D. While forwarding the ROUTE REQUEST, each node creates temporary routing table entries for the reverse route. After finding a route to D, a ROUTE REPLY is unicast to S following the reverse route. The routing table entry is deleted after a specified time-out period. In AODV, each node sends a periodic HELLO packet to notify its neighbor nodes of its presence, and this information is used to maintain valid routes. If a node does not receive an expected HELLO packet from a neighbor (i.e., a HELLO packet from a neighbor through which it is maintaining a route), the node creates a ROUTE ERROR packet that is sent to the nodes that use the route.

Since AODV data packets contain no routing information, they are small, and the overhead of routing once routes are established is minimal. However, while the periodic HELLO packets can help ensure fresh routes in the dynamic environment, they generate unnecessary packet overheads in more static environments.

3.2 DSR

Dynamic Source Routing (DSR) [15] is also a reactive routing protocol but uses the source routing paradigm. In DSR, when a node S wants to send a data packet to a node D and S does not have a valid route in its route cache, S initiates a route discovery. S broadcasts a ROUTE REQUEST packet that is flooded within the network in a controlled manner. When a node receives the ROUTE REQUEST, if the node is the destination or has a valid route in its route cache, it sends a ROUTE REPLY. Otherwise, the node forwards the ROUTE REQUEST, appending its address to the list of nodes the ROUTE REQUEST has traversed. Since the ROUTE REQUEST records the nodes on the path, the ROUTE REPLY packet can deliver to S the complete path to D. DSR makes aggressive use of the route cache for the reduction of ROUTE REQUEST packets by learning or overhearing the route from the ROUTE REQUEST/REPLY packets.

Since each DSR data packet carries the complete route from the source to the destination in the header, the intermediate nodes do not need to maintain the routing information. In addition, nodes do not need to advertise themselves periodically. However, every data packet carries an entire route as part of its payload, increasing the overhead associated with forwarding data packets.

3.3 DSDV

The third protocol we look at, Destination Sequenced Distance Vector (DSDV) [22], is a proactive routing protocol. Each node in the MANET maintains the next-hop and distance for all reachable destinations in the network. In addition, for loop-free routing, each route entry in the routing table has a sequence number, and a greater sequence number indicates more recent route information. All of the nodes periodically exchange distance vector updates to enable them to maintain valid routes to destinations. When a node A detects a link failure to its neighboring node B, A

advertises routes via B with an infinite hop count, eventually propagating the new distance vector information to all other nodes.

Using periodic route updates, each node maintains up-to-date route information. Thus, although the routing overhead can be greater than reactive routing protocols, the delay from the time a source wants to send a message until the node starts sending the message is quite short, in contrast to the reactive protocols that each must initiate a route discovery first.

4. MOTIVATING SCENARIO

In this section, we present a motivating scenario, a disaster recovery deployment, and demonstrate how our process is applied in this deployment. We first describe the operational environment and then demonstrate the steps of the process through to protocol selection. We include details of building (limited) protocol behavior models for the three protocols described in the previous section.

4.1 Scenario Overview

In a disaster recovery situation, various groups of first responders are deployed into an area in which the communication infrastructure may be inaccessible or even destroyed. In these situations, people with varying tasks, e.g., emergency medical technicians (EMTs), firemen, policemen, search and rescue officers, etc., must perform concurrent tasks. They each collect information about the site (e.g., hot spots, smoke density, location of survivors, etc.) and benefit from accessing data collected by others. In such an area void of an infrastructure, a MANET can be a good solution. For the deployed MANET to fulfill the various users' and applications' requirements, it is necessary to determine the most favorable communication mechanism to use in the network.

4.2 Process in Pre-design Time

The first step in determining an appropriate communication protocol with our tool is ensuring that the tool contains protocol behavior models for the available protocols. For the usual case, these protocol behavior models will already be resident in the tool; we briefly detail how we generated simplified models to evaluate the effectiveness of our initial proof-of-concept tool. Ideally, the tool will use complete protocol behavior models, but for a simple case study we restrict the model to three performance metrics: packet delivery ratio, average end-to-end delay, and average throughput, and we vary measurements within the model based only on two scenario parameters: the degree of mobility and the number of source nodes. Although this model is restricted, it is enough to show that our process is useful to determine the most favorable routing protocol using characteristics of the environment and application requirements.

To build our limited protocol behavior models, we conducted several simulations using the ns-2 network simulator [26]. The simulation parameters are summarized in Table 3, where the only two variable parameters are the number of traffic sources and the pause time, a measure of mobility degree in the random waypoint mobility model. For each pause time from 1 to 900 at possible intervals of 50 (i.e., a total of 19 pause times) and each number of sources (i.e., a total of 10 different numbers of source nodes), we simulated 10 times for each protocol.

Table 3: Simulation Parameters

Parameters	Value
Network Size	1500 m x 300 m
Simulation Time	900 *sec.*
Number of Mobile Hosts	50
Traffic Type	Constant Bit Rate
Packet Size	512 *bytes*
Packet Transmission Rate	4 *packets/sec.*
Number of Traffic Sources	3,6,9,12,15,18,20,24,27,30
Movement Model	Random Waypoint Model
Mobile Hosts Max. Speed	1 $m/sec.$
Pause Time	1, 50, 100, $\cdots$, 850, 900

With the simulation results, we derived a protocol behavior model for each of the protocols in Section 3 on each of the above three performance metrics. Since there were two independent variables (i.e., pause time and the number of source nodes), we used the least square method for multiple variables to fit a smooth model to the simulation data. Table 4 shows the derived protocol behavior models for three target protocols on three performance metrics; the protocol behavior models are represented as equations with two independent variables, p and s, which stand for a pause time and the number of source nodes[1]. From the pertinent equations in Table 4, we can retrieve the performance metrics with a pause time and the number of source nodes in the specific situation. Figure 2 presents simulation results for each pause time and each number of source nodes and multiple regression results for simulations; simulation results are depicted as dots and multiple regression results as a surface.

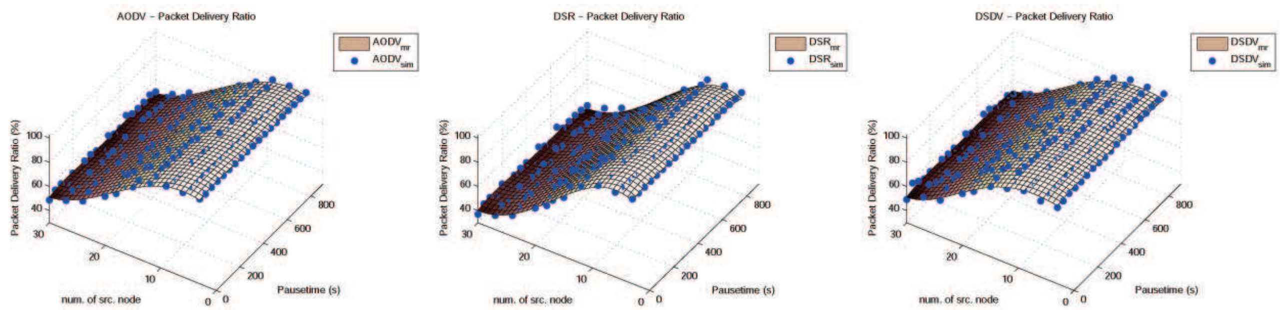
4.3 Process in Design Time

In a disaster recovery scenario, a designer provides our process with environment information and application requirements in design time. Determining the most favorable routing protocol for a particular situation leverages specifics about the movement in that situation (for example, movement in disaster areas has generated a realistic movement model [1]). In this example, we assume 50 first responders are distributed in a disaster area, and each responder follows the random waypoint model at the maximum possible speed of 1m/s (this simplifies the models we had to create above, and is sufficient to demonstrate how our tool works). Since first responders move actively, they do not take a long rest in one position but try to move continuously for prompt action. This "active" environment specification is translated into a low pause time; in this case our tool chose 100 seconds of the pause time for the random waypoint mobility model. For a different movement model, this level of activity may translate to a different parameter specific to that model; this translation is handled by the tool depending on the model. These parameters are entered into our tool by the deployer of the disaster recovery network. Each responder communicates with other responders in a peer-to-peer manner, and the network supports four applications: 1) voice communication, 2) location information exchange 3) command dispatch, and 4) snapshot transfer. All application data is sent in 512 byte UDP packets. On average, 20 communication

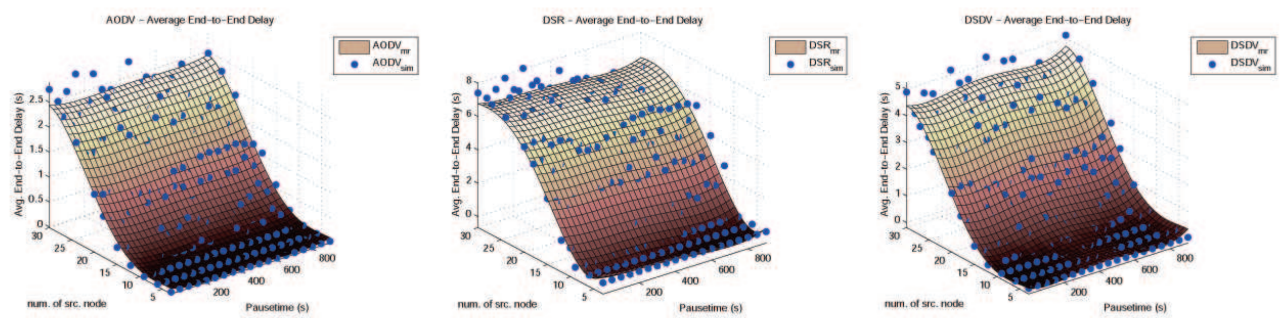
[1]In the table, **PDR**, **DLY**, and **THR** stand for the packet delivery ratio, the average end-to-end delay, and the average throughput, respectively.

Table 4: Protocol Behavior Model

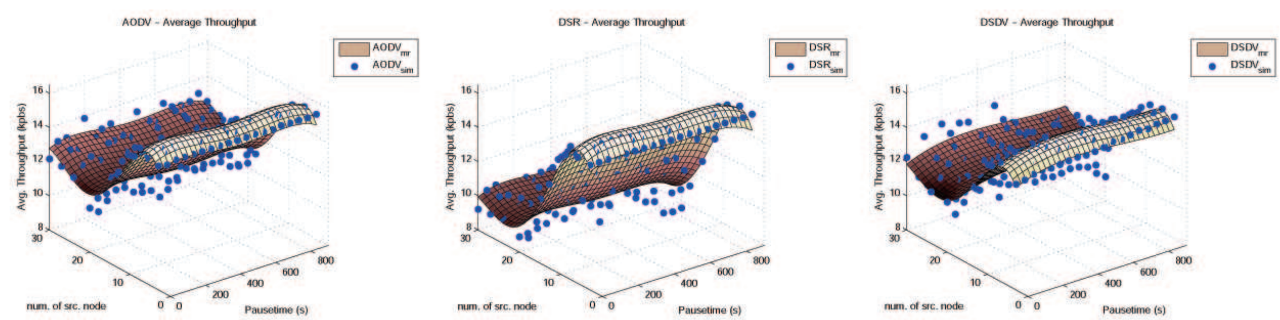
	$f(p,s) = a_0 + a_1 \cdot p^1 + a_2 \cdot p^2 + a_3 \cdot p^3 + a_4 \cdot p^4 + a_5 \cdot s^1 + a_6 \cdot s^2 + a_7 \cdot s^3 + a_8 \cdot s^4$								
	a_0	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8
$AODV_{PDR}$	$9.27 \cdot 10^{1}$	$2.78 \cdot 10^{-2}$	$-1.46 \cdot 10^{-6}$	$2.49 \cdot 10^{-7}$	$-1.37 \cdot 10^{-10}$	$2.84 \cdot 10^{0}$	$-2.74 \cdot 10^{-1}$	$3.12 \cdot 10^{-3}$	$4.35 \cdot 10^{-5}$
DSR_{PDR}	$8.65 \cdot 10^{1}$	$3.83 \cdot 10^{-2}$	$-1.98 \cdot 10^{-4}$	$3.42 \cdot 10^{-7}$	$-1.93 \cdot 10^{-10}$	$5.67 \cdot 10^{0}$	$-6.07 \cdot 10^{-1}$	$1.46 \cdot 10^{-2}$	$-7.83 \cdot 10^{-5}$
$DSDV_{PDR}$	$9.13 \cdot 10^{1}$	$3.11 \cdot 10^{-2}$	$-1.41 \cdot 10^{-4}$	$2.32 \cdot 10^{-7}$	$-1.26 \cdot 10^{-10}$	$1.08 \cdot 10^{0}$	$-8.20 \cdot 10^{-4}$	$-9.29 \cdot 10^{-3}$	$2.21 \cdot 10^{-4}$
$AODV_{DLY}$	$6.48 \cdot 10^{-2}$	$-1.14 \cdot 10^{-3}$	$4.83 \cdot 10^{-6}$	$-7.71 \cdot 10^{-9}$	$4.16 \cdot 10^{-12}$	$3.43 \cdot 10^{-2}$	$-1.00 \cdot 10^{-2}$	$8.84 \cdot 10^{-4}$	$-1.67 \cdot 10^{-5}$
DSR_{DLY}	$2.90 \cdot 10^{0}$	$-2.81 \cdot 10^{-3}$	$1.24 \cdot 10^{-5}$	$-2.43 \cdot 10^{-8}$	$1.51 \cdot 10^{-11}$	$-1.16 \cdot 10^{0}$	$1.39 \cdot 10^{-1}$	$-4.73 \cdot 10^{-3}$	$5.15 \cdot 10^{-5}$
$DSDV_{DLY}$	$-7.13 \cdot 10^{-2}$	$-2.74 \cdot 10^{-3}$	$1.78 \cdot 10^{-5}$	$-3.38 \cdot 10^{-8}$	$1.98 \cdot 10^{-11}$	$8.90 \cdot 10^{-2}$	$-2.47 \cdot 10^{-2}$	$2.02 \cdot 10^{-3}$	$-3.76 \cdot 10^{-5}$
$AODV_{THR}$	$1.17 \cdot 10^{1}$	$5.42 \cdot 10^{-3}$	$-2.88 \cdot 10^{-5}$	$4.97 \cdot 10^{-8}$	$-2.75 \cdot 10^{-11}$	$1.63 \cdot 10^{0}$	$-2.07 \cdot 10^{-1}$	$8.49 \cdot 10^{-3}$	$-1.11 \cdot 10^{-4}$
DSR_{THR}	$1.12 \cdot 10^{1}$	$5.93 \cdot 10^{-3}$	$-2.82 \cdot 10^{-5}$	$4.79 \cdot 10^{-8}$	$-2.71 \cdot 10^{-11}$	$1.86 \cdot 10^{0}$	$-2.31 \cdot 10^{-1}$	$8.92 \cdot 10^{-3}$	$-1.11 \cdot 10^{-4}$
$DSDV_{THR}$	$-3.24 \cdot 10^{1}$	$9.49 \cdot 10^{-3}$	$-1.03 \cdot 10^{-6}$	$5.05 \cdot 10^{-9}$	$-1.07 \cdot 10^{-13}$	$6.52 \cdot 10^{0}$	$2.05 \cdot 10^{-1}$	$-4.64 \cdot 10^{-3}$	$-4.64 \cdot 10^{-5}$



(a)



(b)



(c)

Figure 2: Simulation Results: (a) packet delivery ratio, (b) average end-to-end delay, and (c) average throughput

channels are expected to be set up between pairs of responders at any given time. In case of voice communication, the G723.1 audio codec is used. The command dispatch application is assumed to be the most important among the four applications, and it is necessary to guarantee delivery of the command. Having consistent up-to-date location in-

Table 5: Performance Metric Requirements - Disaster Recovery Scenario

Performance Metric	Applications			
	voice	location	command	snapshot
throughput (kbps)	6.4	0.3	1	1.6
delivery ratio (%)	90	50	95	90
delay (s)	1	5	3	undefined

Table 6: Priority Information - Disaster Recovery Scenario

Weight Factor	Applications			
	voice	location	command	snapshot
application	5	1	10	3
throughput	7	1	9	3
delivery ratio	5	3	10	5
delay	10	10	7	0

formation is not as important, partly because the updates are sent periodically and if a few are missed, the application can survive. Therefore the location application has the lowest priority; the other two applications fall in between.

From the requirement analysis step, the above environment parameters, performance metric requirements, and priority information are derived from the information provided by the designer. Performance metric requirements are inferred from application requirements. In the voice communication application, it is known that the G723.1 audio codec requires 6.4 kbps as a bandwidth [12, 20]. In addition, the audio packets are required to arrive within 1 second and to deliver under 15 percent packet loss. Similarly, the other applications are described with respect to a minimum throughput, a minimum delivery ratio, and a maximum delay. Table 5 shows the specific performance metric requirements for our simple case study.

Priority information is also constructed from the requirement analysis step; it is shown in Table 6. In the command dispatch application, since the delivery certainty is the most significant, 10 is assigned to the pertinent performance metric weight factor (i.e., deliver ratio weight factor). Similarly, the end-to-end delay and throughput weight factor have 8 and 1 as weight factors in the command dispatch application.

With derived parameters in design time and the protocol behavior model in pre-design time, we can construct the preference value table (i.e., Table 2). In the final stage, we calculate the total weighted sum of the preference values with respect to three routing protocols; Table 7 is the result for our simple case study. Therefore, for the situation, *AODV* is determined to be the most appropriate routing protocol among three routing protocols.

In other situations, application weights can differ from Table 6; each responder must notify his or her position as exactly as possible, and voice communication is unimportant since responders are unable to speak due to poisonous fumes. In this situation, when application weights are assigned to 0, 10, 9, and 6 for voice, location, command, and snapshot application respectively, our tool shows that *DSDV* is more suitable than AODV or DSR[2] demonstrating that changing application requirements or even just the weights associated with them can have a significant impact on the "ideal" protocol for deployment.

Table 7: Total Sum of Weighted Preference Values

	Routing Protocols		
	AODV	DSR	DSDV
S^k_{total}	31.88	330.72	69.17

5. RELATED WORK

While most of the related previous work has provided simulation results to compare the performance of routing protocols, some work has discussed using the results to select a routing strategy [16]. We build on this to describe a tool that automates the analytical selection of the most appropriate routing protocol for a particular deployment. To support our tool, it is necessary to build protocol behavior models described by parameters enumerated above.

In general, node mobility is one of the most important factors influencing to characteristics of routing protocols in MANETs. For that reason, several studies on a modeling of the mobility have been made [3, 6, 7, 11, 13, 25]. This work has focused on how to describe a node's movement in a MANET more realistically. They also demonstrated that routing protocols showed different performance with different movement models. Other work has created a mobility models and simulation results that describe a specific real world situation [1]. Our work builds on these approaches to allow realistic models and results that come from using them to guide network deployers in the proper selection of routing protocols.

Recently, work on an autonomic routing in MANETs has been published [17, 18]. Autonomic routing protocols originate from autonomic computing [10, 21] where a system adapts to its environment dynamically. In an autonomic routing system, an adaptive routing service is provided to mobile nodes, guaranteeing consistency, smoothness, and efficiency [18]. Recent work [2] summarizes the main design issues for autonomic routing, which focus on real-time adaptation at an increased cost in terms of overhead and delay. Our approach is different from autonomic routing; while autonomic routing focuses on interoperability among routing protocols, we find the most favorable routing protocol in a particular situation. However, the approaches can be complementary in that their combination has the potential to provide a highly adaptive protocol suite.

6. CONCLUSIONS AND FUTURE WORK

In this paper, we described a novel design tool to support a software designer in selecting the most appropriate routing protocol for a target network deployment. Using environment characteristics and application requirements input by the designer, the tool analyzes the information and refers to protocol characteristic models (generated from simulations or real world experiments).

To extend this work, further research can investigate how to more effectively create general protocol behavior models. Furthermore, our work can be developed to a tool for the *real-time* routing protocol decision. However, to extend to the real-time adaptive protocol scheme, we must solve the

[2]The total sum of weighted preference values for AODV, DSR, DSDV are 24.18, 101.95,and 23.27, respectively

complicated problems such as the real-time sensing of environmental information and the consistency while switching a routing protocol in MANETs. This approach supports a highly adaptive protocol suite that leverages the relative benefits of competing communication paradigm.

7. REFERENCES

[1] N. Aschenbruck, M. Frank, P. Martini, and J. Tölle. Human mobility in manet disaster area simulation - a realistic approach. *Proceedings of the 29th Annual IEEE International Conference on Local Computer Networks (LCN04)*, November 2004.

[2] O. Bazan and M. Jaseemuddin. Routing in autonomic communications. *Consumer Communications and Networking Conference, 2006. CCNC 2006. 2006 3rd IEEE*, 1:91–95, Jan. 2006.

[3] C. Bettstetter. Smooth is better than sharp: A random mobility model for simulation of wireless networks. *In Proc. 4th ACM Int. Workshop on Modeling, Analysis, and Simulation of Wireless and Mobile Systems (MSWiM01)*, July 2001.

[4] A. Boukerche. Performance evaluation of routing protocols for ad hoc wireless networks. *Mob. Netw. Appl.*, 9(4):333–342, 2004.

[5] J. Broch, D. A. Maltz, and et al. A performance comparison of multi-hop wireless ad hoc network routing protocols. *In proceedings of the ACM/IEEE MOBICOM*, pages 25–30, October 1998.

[6] T. Camp, J. Boleng, and V. Davies. A survey of mobility for ad hoc network research. *Wireless Commun. Mobile Comput. (WCMC)*, pages 483–502, 2002.

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

[8] S. R. Das, C. E. Perkins, and E. M. Royer. Performance comparison for two on-demand routing protocols for ad hoc networks. *In proceedings of the IEEE InfoCom*, 2000.

[9] T. Greville. *In Mathematical Methods for Digital Computers*, volume 2. Wiley, New York, 1967.

[10] S. Hariri. Autonomic computing: research challenges and opportunities. *CAIP Seminar*, February 2004.

[11] X. Hong, M. Gerla, G. Pei, and C.-C. Chiang. A group mobility model for ad hoc wireless networks. In *MSWiM '99: Proceedings of the 2nd ACM international workshop on Modeling, analysis and simulation of wireless and mobile systems*, pages 53–60, New York, NY, USA, 1999. ACM Press.

[12] ITU Rec. G.723.1. *Dual Rate Speech Coder for Multimedia Communications Transmitting at 5.3 and 6.3 kbit/s*, March 1996.

[13] P. Johansson, T. Larsson, N. Hedman, B. Mielczarek, and et al. Scenario-based performance analysis of routing protocols for mobile ad-hoc networks. *In proceedings of the ACM/IEEE MOBICOM*, pages 195–206, August 1999.

[14] D. B. Johnson and D. A. Maltz. *Dynamic source routing in ad hoc wireless networks. In Mobile Computing*, chapter 5, pages 153–181. Kluwer Academic Publishers, 1996.

[15] D. B. Johnson and D. A. Maltz. *Mobile Computing*. Kluwer Academic Publishers, 1996.

[16] S.-J. 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, 2003.

[17] F. Legendre, M. D. de Amorim, and S. Fdida. Context-aware inter-networking for wireless networks. *in Proceedings of IEEE/IFIP International Conference on Mobile and Wireless Communication Networks*, 2004.

[18] F. Legendre, M. D. de Amorim, and S. Fdida. Some requirements for autonomic routing in self-organizing networks. In *Autonomic Communication: First International IFIP Workshop, WAC 2004*, pages 13–24, London, UK, 2005. Springer-Verlag.

[19] S. G. W. Leona S. Aiken. *Multiple Regression: Testing and Interpreting Interactions*. SAGE Publications, 1991.

[20] Microsoft. *Network Bandwidth Considerations, http://www.microsoft.com/windows/NetMeeting/Corp/reskit/Chapter7/default.asp.*

[21] M. Parashar and S. Hariri. Autonomic computing: An overview. *Lecture Notes in Computer Science*, 3566 / 2005, 2005.

[22] C. E. Perkins and P. Bhagwat. Highly dynamic destination-sequenced distance vector routing (dsdv) for mobile computers. *In proceedings of the ACM SIGCOMM '94 Conference on Communications Architecture, Protocols and Applications*, pages 234–244, August 1994.

[23] C. E. Perkins and E. M. Royer. Ad hoc on demand distance vector routing. *In proceedings of the IEEE Workshop on Mobile Computing System and Applications*, February 1999.

[24] E. Royer and C. K. Toh. A review of current routing protocols for ad hoc mobile wireless networks. *in IEEE Personal Communications*, 1999.

[25] D. Shukla. A method of program refinement. Master's thesis, KReSIT ITT Bombay, Bombay, India, November 2001.

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