

# Component Based Performance Modelling of the Wireless Routing Protocols

Vahid Tabatabaee, John S. Baras, Punyaslok Purkayastha,  
Kiran Somasundaram

The  
Institute for  
**Systems**  
Research



**A. JAMES CLARK**  
SCHOOL OF ENGINEERING

ISR develops, applies and teaches advanced methodologies of design and analysis to solve complex, hierarchical, heterogeneous and dynamic problems of engineering technology and systems for industry and government.

ISR is a permanent institute of the University of Maryland, within the A. James Clark School of Engineering. It is a graduated National Science Foundation Engineering Research Center.

[www.isr.umd.edu](http://www.isr.umd.edu)

# Component Based Performance Modelling of the Wireless Routing Protocols

Vahid Tabatabaee  
Institute for Systems Research  
University of Maryland  
vahidt@umd.edu

John S. Baras  
Institute for Systems Research  
University of Maryland  
baras@umd.edu

Punyaslok Purkayastha  
Institute for Systems Research  
University of Maryland  
punya@isr.umd.edu

Kiran Somasundaram  
Institute for Systems Research  
University of Maryland  
kirans@umd.edu

## ABSTRACT

In this paper, we propose a component based methodology for modelling and design of wireless routing protocols. Componentization is a standard methodology for analysis and synthesis of complex systems. Throughout the paper, we use Optimized Link State Routing (OLSR) protocol as a case study to demonstrate effectiveness of our methodology. We focus on modelling of three main components: neighborhood discovery, selector of topology information to disseminate, and the path selection components. For each component, we identify the inputs, outputs, and a generic methodology for modelling. Using the neighborhood discovery component, we will present our design methodology and design a modified enhanced version of the OLSR NDC, and compare its performance to the neighborhood discovery component of the OLSR protocol.

## Keywords

Component based routing, component based modelling, OLSR

## 1. INTRODUCTION

We present component based methodology for performance modelling of the wireless networks. This method provides a systematic and low complexity approach that can be used in the study, analysis, design and optimization of wireless networks.

Despite tremendous research and interest in wireless networks, we still do not have systematic methodologies and tools that would allow us for efficient analysis and cross-layer synthesis of such networks with the provision of accurate performance bounds. The main reason for this is the different nature of wired and wireless networks rendering the use of wired network techniques inappropriate for the case of wireless networks. Key quantities, such as the link

capacity, that remain constant in a wired network vary in wireless communication environments with the transmission power, the interference, the node mobility and the channel condition. Due to the performance variability and interdependence, design, analysis, optimization, management and maintenance of such systems are daunting tasks. Modelling and model-based approaches that consider complex relations, interactions, and interdependencies between connections are needed to help us in achieving deeper understanding and greater insight in the fundamental properties of wireless networks, and in development of a systematic and scalable methodology for design of wireless networks and protocols.

The input variables and functions that determine the performance of a wireless communication network can be classified into four groups associated with the four main sections of the wireless network architecture, which are: (i) Routing, (ii) Node scheduling, (iii) MAC layer, and (iv) PHY layer. Hence, besides inter-dependence and coupling between the node to node communication channel characteristics, we should also model the cross-layer (cross-section) interaction and inter-dependence between the four main sections of the communication network architecture. It may be possible to develop packet level simulation tools based on appropriate physical (PHY), medium access control (MAC), routing and scheduling models. However, the result would not be scalable. Further, the developed models are not appropriate for studying dynamic relation and correlation between the network parameters and performance metrics. Therefore we need alternative approaches for synthesis, optimization, sensitivity analysis and studying fundamental properties of wireless networks.

Our overarching objective is to develop low complexity combined analytical and computational (numerical) models, which can efficiently approximate wireless networks performance. Since the models are deterministic in addition to zero order metrics such as throughput and delay, we can use analytical and numerical methods to approximate the first and second order metrics (gradient and Hessian of the zero order metrics). The higher order metrics enable us to develop efficient systematic methodologies for sensitivity analysis, synthesis and optimization of wireless networks.

In this paper, we focus on the routing and present our method for component based performance modelling of the routing protocol. In this method, we first identify the main components that implicitly or explicitly should be present in wireless routing protocols. We specify inputs and outputs of these components and specify how they are interconnected to each other. We then propose appropriate methodologies for modelling each one of these components. We use Neighborhood Discovery Component (NDC) as an example and introduce metrics for the study of its performance. Based on the introduced metrics, we will also present a methodology for design and enhancement of the NDC.

Some of the main advantages of the component based Modeling are: (1) It provides a natural and top-down strategy for *structural design* of the routing protocols. (2) In the analysis stage, sources of problems and performance degradation can be detected and fixed faster by using a systematic component based approach. (3) We can quantify and study effects of the components and their design parameters on the system performance, and perform sensitivity analysis to enhance the performance and/or design a more robust system. (4) Since the Modeling is not based on a packet based discrete even simulation the simulation is much faster and can be efficiently incorporated in a design by analysis loop.

The paper is organized as follows: In section 2, we introduce the three main components and the high-level model architecture. In section 3, we review OLSR and introduce its functionality based on the component based Modeling framework. In sections 4, 5, 6, we present our methodology for Modeling of the three main routing components by using OLSR as an example. In section 7, we explain how we can use the component models for the design of routing protocols and will use the developed models to evaluate performance (throughput) of the designed routing protocols. In section 8, we summarize the results and discuss future research directions.

## 2. THE PERFORMANCE MODEL ARCHITECTURE

In this section, we introduce the main components of our performance model, their inputs, outputs, and inter-connections. Consider a network that consists of  $N$  nodes,  $i = 1, \dots, N$ . The probability of successful packet transmission from node  $i$  to node  $j$  is  $s_{ij}$ , and  $f_{ij} = 1 - s_{ij}$  is probability of transmission failure. For simplicity, we are assuming that packets have fixed length; hence, we can specify the probability of successful packet transmission.

There are three main components in our performance model for routing protocols, which are: (i) Neighborhood Discovery Component (NDC), (ii) Selector of Topology Information to Disseminate Component (STIDC), (iii) Route or path Selection Component (RSC). Figure 1 illustrates how these components are connected to each other and to the Packet forwarding/scheduling and PHY and MAC layer models. Paper [1] presents a methodology for modelling the 802.11 MAC layer in a multi-hop/multi-path wireless network. In this paper, we present methodologies for modelling the first 3 mappings that corresponds to the three components of the routing protocol.

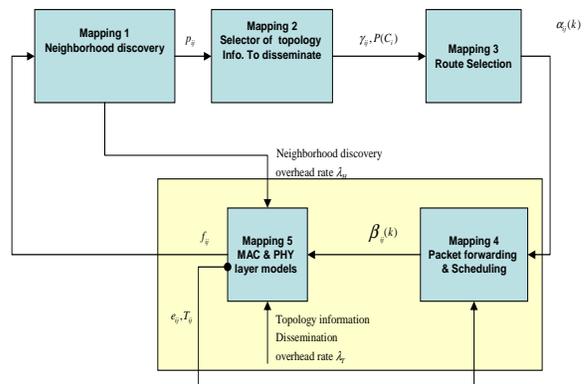


Figure 1: Architecture of the Component Based Model.

The main purpose of the first mapping, NDC is to detect neighbors and bidirectional links between the nodes. The input to NDC is the probability of failure in transmission of a packet,  $f_{ij}$  on the link between node  $i$  and  $j$  for  $i, j = 1, \dots, N$ . More precisely, input is the probability of failure in transmission of HELLO messages between the nodes. Note that nodes broadcast HELLO messages to inform other nodes about their presence and to identify neighboring relations among themselves. The output of this model is,  $p_{ij}$  for  $i, j = 1, \dots, N$ , which is the detection probability of bidirectional links among nodes  $i$  and  $j$ .

The probability of bidirectional link detection is the input to the second mapping, STIDC. STIDC specifies the topology graph that is presented to the network nodes that they use to select the paths and/or next-hop toward the destination nodes. In the link-state routing protocols, which we consider in this work, nodes broadcast a subset of their connected links in the network. For instance, in OSPF state of all links are flooded into the network, hence every node has access to the complete network topology and use it to select the optimal next-hop node for each destination. In the wireless networks, due to the capacity scarcity, it is desirable to select a subset of the nodes to perform the topology information dissemination and to select a subset of the links to broadcast their information in the network. Thus, instead of broadcasting every link information, STIDC component selects a subset of the connected links of a node to disseminate their information. We say state of node  $i$ ,  $C_i$  is its connected links that are selected by STIDC. Clearly,  $C_i$  is a random process. Thus, the natural output for the STIDC is  $P(C_i)$ , the probability mass function (pmf) of the node state. Note that there are finite possible combinations for the state vector, and the pmf will have finite non-zero values.

The inputs to the RSC are the outputs of the NDC and STIDC components. The main task of RSC is to select paths or the next-hops in the paths to the destination nodes. The decision is based on the local neighboring information provided by the NDC and global network topology information provided by the STIDC component. In this paper, we consider hop-by-hop routing protocols; hence, the output of RSC is the probability of selecting node  $j$  as the next-hop of node  $i$  to destination  $k$ , which is denoted by  $\alpha_{ij}(k)$ .

### 3. OLSR COMPONENTS

OLSR [2] is a proactive, link-state routing protocol that uses periodic message flooding in the network to disseminate and update topology information in the network. Each node uses the topology information to find the shortest path and the corresponding next-hop node for each destination in the network. In this section, we describe the OLSR functionality based on the component based architecture that was presented in section 2. For details on the operation of OLSR, refer to [2].

The NDC component of OLSR is responsible for detection of changes in a node neighborhood. A node is said to be a neighbor of another node if there exists a bidirectional communication link between them. The link is bidirectional if communication in both directions are possible. Node  $k \neq i$  is said to be the second order neighbor of  $i$ , if it is not a neighbor (or first order neighbor) of  $i$ , but it is a neighbor of one of the node  $i$ 's neighbors.

NDC relies on periodic transmission of HELLO messages for detection of first and second order neighbors. A HELLO message from node  $i$  contains the transmitting node ID, and the list of detected nodes by node  $i$ , and the status of the links (directional or bidirectional) between node  $i$  and its detected nodes. Thus, from the received HELLO messages a node can detect its first and second order neighbors. When a node  $j$  receives a HELLO message from node  $i$  that contains its own ID, it will add node  $i$  to its neighbor list. Node  $j$  removes  $i$  from its neighbor list if it does not receive any HELLO message from  $i$  for the Neighbor Hold Time (NHT) period.

The STIDC component of a node in OLSR is responsible for selection of Multi-Point Relay (MPR) nodes among the neighbors. Every node selects its MPRs such that the selected neighbors cover all of its second order neighbors. The MPR selector set of a node  $i$  is the set of node  $i$  neighbors that have selected node  $i$  as their MPR. All nodes with non-empty MPR selector set periodically generates a topology control message that is sent to all nodes in the network and contains the ID of the generating node and all MPR selectors of that node. In this way, a node advertises that it can forward packets to its MPR selectors. Upon reception of the topology control messages every node has access to a partial topology of the network that contains all nodes and a subset of the links. Furthermore, if the original network is connected, every node is reachable through the partial topology too.

In our modelling, we consider a simple heuristic that is commonly used in the OLSR implementations for selection of MPR nodes [4]: At every node  $i$ , the algorithm starts with an empty set  $MPR(i)$ . It first selects the neighbors as MPR, which are the only neighbor for some second order neighbors. Then, while there are some second order neighbors that are not covered by the selected MPRs, it adds the neighbor node, which covers most number of not covered second order neighbors to its MPR list.

The RSC uses the partial topology that is produced by STIDC to estimate the distance (hop-count) of the detected neighbors to every destination. For each destination the

neighbor with minimum distance is selected as the next-hop. It is crucial to note that the next-hop is selected among all detected neighbors and not only the MPR selectors. Hence, while the partial topology is used for distance approximation, all detected bidirectional links are considered in the next-hop selection process.

### 4. NDC MODELING

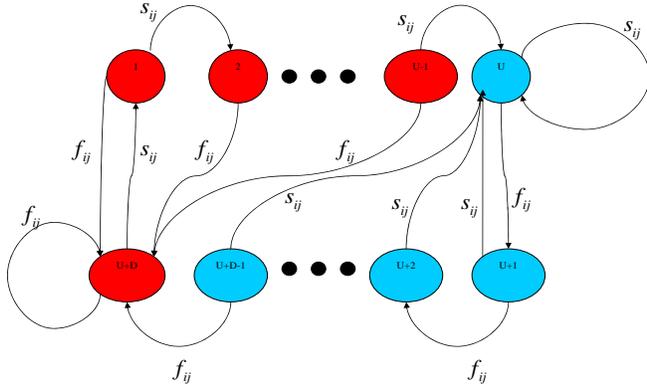
In wireless networks where the underlying topology is constantly changing, we need a reliable mechanism to detect the neighbor nodes. This is the role of neighborhood discovery component (NDC) of routing protocols. Delay and/or error in the NDC results in missing links in the topology which in turn affects the routing performance. It is shown that [3] neighborhood discovery has profound impact on the performance of the routing algorithm. Thus, it is crucial in our performance modelling to study and consider the effects of NDC carefully.

In general, a node periodically broadcasts HELLO messages containing information about its detected neighbors. By using this local information nodes can establish bidirectional links and also identify their second order neighbors. Each HELLO message contains the list of detected neighbors of a node. When a node receives a HELLO message from another node that contains its own ID, it realizes that there is a functional bidirectional link between them and add that node to its neighbor list.

We model the NDC as a Finite State Machine (FSM), so that we can use Markov Chain machinery to derive the desired steady state probabilities for NDC. A simple Markov Chain Model for the link between nodes  $i$  and  $j$  is depicted in Figure 2. The parameter  $s_{ij}$  is the probability of success in sending hello messages and  $f_{ij}$  is the probability of failure. In states  $U$  to  $U + D - 1$  (blue states), NDC considers that the link is (detected) and node  $i$  will add  $j$  in its HELLO messages; in states 1 to  $U - 1$  and state  $U + D$  (red states) we consider that the link is DOWN (not detected). Suppose that we are initially at state  $U + D$ ; only after  $U$  subsequent successful reception of HELLO messages we move to state  $U$ , and a directional link is declared detected. Similarly, when we are in state  $U$ , there should be  $D$  subsequent failures in reception of HELLO messages before a directional link is removed from the list.

Our proposed model is more comprehensive than OLSR neighborhood discovery. In OLSR the value of  $U$  is fixed to 1, since after the reception of only one HELLO message, the recipient assumes that a directional link exists, and after receiving a HELLO message which contains the recipient ID, it declares that there is a bidirectional link between the nodes. Therefore, in OLSR we can only control the  $D$  parameter by changing NHT.

Let  $\pi_k$  be the steady state probability that NDC is in state  $k$  of the Markov Chain. We can use the generalized global balance equations to derive the steady state probabilities. The probability of detecting a directional link to node  $j$  at node  $i$  is:



**Figure 2: Markov chain model for the NDC detection mechanism.**

$$q_{ij} = \sum_{k=U}^{U+D-1} \pi_k, \quad (1)$$

and if we assume that the probability of successful transmission from  $i$  to  $j$  and from  $j$  to  $i$  are independent from each other, then the probability of a bidirectional link detection is:

$$p_{ij} = q_{ij}^2. \quad (2)$$

From the global balance equation, we can derive the probability of every state as a function of state  $U + D - 1$ :

$$\begin{aligned} \pi_{U+k} &= \pi_{U+D-1} / f_{ij}^{D-1-k} \quad k = 0, \dots, D-2 \\ \pi_{U-k} &= f_{ij} \pi_{U+D-1} / s_{ij}^k \quad k = 1, \dots, U-1 \end{aligned} \quad (3)$$

Now we use the fact that the summation of states probabilities should be one to get,

$$\pi_{U+D-1} = \left( s^{1-U} + f s^{-U} + (1 - f^{D-2}) f^{2-D} s^{-1} + f^{1-D} \right)^{-1}. \quad (4)$$

For simplicity, we have dropped the  $ij$  index for  $f$  and  $s$  in the above equation.

The design (or control) parameters for NDC are  $U$  and  $D$  parameters that can be set to achieve the desired performance. We can consider a number of performance metrics for NDC. Delay in the detection of a neighbor and delay in removing a node from the neighbor list are good examples. We can use the Markov chain analysis techniques to approximate and/or compute lower and upper bounds for these parameters. Here, we consider predictability of the link status as the performance metric.

Many of the problems that arise in wireless networks routing protocols are related to the unpredictability of the link status. In a link state routing protocol, such as OLSR, changes in the link status result in changes in the set of links (MPRs for OLSR) that are presented in the partial topology. These changes should be disseminated to all nodes in the network and will increase the traffic overhead, contention and congestion in the network, and can also cause disruption in forwarding of the packets.

Our design methodology is based on the selected performance metric; links are divided into three groups based on the probability of success in sending the HELLO messages,  $s_{ij}$ . If  $s_{ij} \leq P_l$ , we assume that the link is not stable and detection probability should be close to zero. If  $s_{ij} \geq P_u$ , then the link is stable and detection probability should be close to 1. The threshold values  $P_l$  and  $P_u$  should be set based on the overall performance of the wireless network protocols. The status of links in the transition region, i.e. links with  $P_l < s_{ij} < P_u$  is still unpredictable, but for other regions link status are with high probability predictable. Thus, the transition region should be as narrow as possible. In section 7, we will illustrate our design methodology through a simple example.

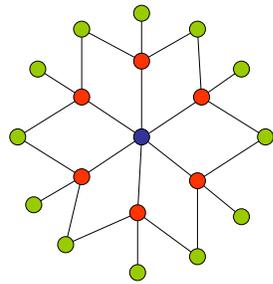
In OLSR, we can only control the  $D$  parameter and  $U$  is always one. Hence, as we demonstrate in section 7, with respect to the link status predictability, OLSR does not provide us enough control.

## 5. STIDC MODELING

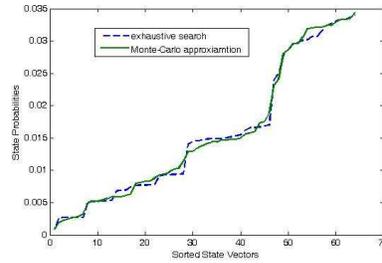
This component is responsible for selection of the links that are represented in the partial topology that network nodes use to estimate their distances and next-hops toward desired destination nodes. In section 3, we explained that in OLSR, the links are selected through the selection of the MPR nodes. Every node selects a subset of its detected neighbors as its MPRs. Node  $i$  selects its MPRs such that all its second order neighbors (neighbors of neighbors) are covered by them, i.e. there is at least one MPR node that connects node  $i$  to each one of its second order neighbors. In the current implementation, we consider a simple greedy algorithm for MPR selection, which is commonly used for MPR selection and introduced in section 3; however alternative algorithms can also be implemented with the same methodology.

The selected MPR nodes depend on the detected neighbors and their detected neighbors which are random variables. Therefore, the selected MPR nodes are also random processes. Our goal is to derive the statistic characteristics of this random process. Recall that  $C_i$ , state of node  $i$ , is a binary vector with  $\Delta_i$  (degree of node  $i$ ) elements. Each element of the state vector is 1 if its corresponding neighbor is selected as an MPR, and it is 0 otherwise. The output of this model is the pmf of the state vector for every node  $i$ . Let assume that the link detection probabilities are mutually independent. Hence, for each possible combination of detected links, we can find its probability and the selected MPRs, and in this way, we can derive the state vector pmf. However, this approach is not scalable in the number of links. For each node  $i$ , we have to consider all possible combinations of detected links of  $i$  and neighbors of  $i$ . If degree of node  $i$  and its neighbors are  $\Delta$ , then we have to consider  $O(2^{\Delta\Delta})$  combination of detected links, which is not scalable. However, in practice most of these combinations have negligible probability and we do not need to consider them. Hence, instead of exhaustive search, we use Monte-Carlo simulation to generate the combination of detected links and from that we approximate the MPR state vector probabilities.

The MPR selection at each node is based on the local information; hence, in the Monte-Carlo simulation for each node



(a) The network topology with 0.7 link detection probability.



(b) The computed state probabilities for the center node using Monte-Carlo approximation and exhaustive search.

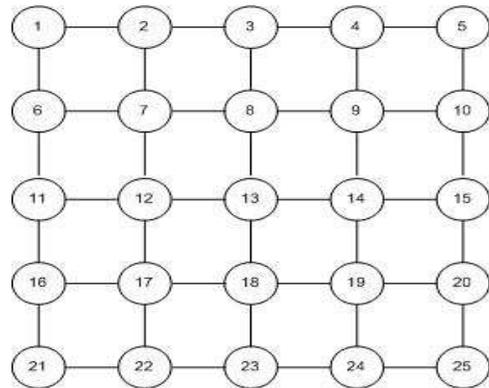
**Figure 3: The star network topology for testing the MPR selection component**

we only need to simulate its links and its neighbor node links. Therefore, our approach is scalable and its complexity increases linearly in the size of the network, as long as the degree of the nodes is fixed. Furthermore, the Monte-Carlo simulations for the network nodes can run in parallel. For the star topology network shown in Figure 3(a), where the probability of detecting every link is set to 0.7, we have computed the probability of MPR state vectors for the center node using exhaustive search over all possible combination of detected links in the network and by using Monte-Carlo simulation with 10,000 iterations. The results are plotted in Figure 3(b). On the x-axis we have all possible 64 states (there are 6 neighbor nodes for the center node) sorted by their probability and on the y-axis we have the state probabilities. It is clear that the Monte-Carlo approximation is very close to the result of exhaustive search.

## 6. RSC MODELLING

This component is responsible for computation of the routing tables entries at every node, which specifies the next hop neighbor for all possible destinations in the network. In OLSR, the routes are selected based on the partial topology information that is broadcasted in the network by the selected MPRs. Every node, advertises that it has a link to the nodes that have selected it as one of their MPRs. Note that directions of advertised links are from MPR to the MPR selector. Every node obtains the partial network topology from this information and uses it to compute the routes and populates its routing table. We assume that the nodes are using minimum hop routing on the partial topology to select their next-hop. However, anyone of the detected neighbors can be selected as the next hop. Thus, the topology that is used for routing by every node is different and consists of the links in the partial network topology that is advertised by the STIDC, plus the bidirectional links that are in the HELLO messages received from the neighbor nodes. This combination of local and global information makes the OLSR routing protocol resilient to link failures and should be considered in the RSC modelling.

We divide the RSC modelling into three stages. The first stage is estimation of the *average* distance computed at every node to the destination. In the second stage, we use the computed average distance and the minimum possible



**Figure 4: A simple 5x5 Grid Topology.**

distance that can be computed from the network topology to estimate the probability mass function (pmf) of the distances computed at a node. The network topologies that are used by the nodes to estimate the distance are random processes, hence the computed distances are also random process. In the first stage, we estimate the average of these random processes and in the second stage we estimate their pmf. Finally in the third stage, we compute the probability of selecting a neighbor node  $j$  as the next-hop for node  $i$  in forwarding the packets to destination  $k$ .

### 6.1 Average Distance Estimation

We use a probabilistic version of the Bellman-Ford (BF) dynamic programming computation to compute the average distance of the nodes on the partial topology. An important fact that needs to be taken into account is that an MPR for a node is a neighbor node that forwards packets to it and not viceversa. This interpretation, as provided in the RFC of OLSR [2], has prompted us to do the average distance computations first for a *reverse* version of the partial topology network. Thus, we compute the average distances  $h(i, k)$  from node  $i$  to node  $k$  in the reverse network using the dynamic programming equations and compute the average distances in the forward network  $D(i, k)$  using the fact that  $D(i, k)$  must be equal to  $h(k, i)$ .

For the dynamic programming equations we use the MPR

state probabilities that are computed in the STIDC performance model. However, we have to modify these probabilities for each destination to take into account the fact that in OLSR every node can select any of its neighbors to forward the packet and not just the links to the MPR selectors. Hence, in the reverse network, we modify the probabilities of the MPR configurations, so that every node that is connected to the final destination can forward its traffic to it directly, even if the destination node is not its MPR. In the forward network, this modification is equivalent to the assumption that the source node can use any of its neighbors (and not only its MPR selectors) with minimum distance to forward the packet.

Let  $h(i, k, C_i)$  denotes the average hop count from  $i$  to  $k$  at state  $C_i$  (MPR configuration) in the reverse network, and  $h(i, k)$  the average hop count over all states  $C_i$ .  $MPR(C_i)$  denotes the MPR set of node  $i$  at state  $C_i$ . For every destination, we have to first modify the MPR state vector probabilities for the neighbors of the destination, so that probability of having connection to the destination is equal to probability of having a detected link (and not having destination being selected as an MPR). After this modification in the state probabilities, to compute average hop counts to the destination  $k$  in the reverse network, we use the following DP equations:

$$\begin{aligned} h(i, k, C_i) &= 1 + \min_{j \in MPR(C_i)} (h(j, k)) \\ h(i, k) &= \sum_{C_i} P(C_i) h(i, k, C_i) \end{aligned} \quad (5)$$

After convergence,  $D(i, k)$ , the average hop count in the forward network is set equal to  $h(k, i)$ .

## 6.2 Distance Distribution Estimation

We explained that every node uses the partial topology and its local neighbor topology information to estimate its distance to every other node in the network. Due to the randomness of the network topology the estimated distances are random variables too. In the previous section, we estimated the average of these random variables. It is also easy to estimate the minimum of these random variables,  $m(i, k)$  from the network topology. We also assume that the maximum distance between two nodes is,

$$M(i, k) = \max(2D(i, k) - m(i, k), 2m(i, k)) \quad (6)$$

Then, we approximate the pmf of this r.v. using maximum entropy method. Let  $p_j$  be the probability that the distance from  $i$  to  $k$  is  $j$  hops. In order to approximate  $p_j, j = m(i, k), \dots, M(i, k)$ , we solve the following optimization problem:

$$\begin{aligned} \min \quad & \sum_{j=m(i,k)}^{M(i,k)} p_j \log p_j \\ \text{s.t.} \quad & \\ & \sum_{j=m(i,k)}^{M(i,k)} p_j = 1 \\ & \sum_{j=m(i,k)}^{M(i,k)} j p_j = D(i, k) \end{aligned} \quad (7)$$

Using the lagrange multipliers, we reach the conclusion that

$$p_j = e^{-\mu j} / \sum_{x=m(i,k)}^{M(i,k)} e^{-\mu x}, \quad (8)$$

where  $\mu$  is the solution of the following equation:

$$\sum_{j=m(i,k)}^{M(i,k)} j e^{-\mu j} / \sum_{x=m(i,k)}^{M(i,k)} e^{-\mu x} = D(i, k). \quad (9)$$

## 6.3 Next-hop Probability Estimation

After computation of the distances pmf, we can approximate the probability of selecting node  $j$  as the next hop for node  $i$  to destination  $k$ . For every node, we first list all possible combination of its detected links. If a node has  $\Delta$  node in its communication range, there are  $2^\Delta$  possible combinations of its detected links. The probability of each combination can be computed using the information provided from the NDC assuming that link status are independent random variables. For each combination, probability of having a particular detected neighbor  $j$  be the next hop is proportional to the probability of  $j$  having minimum distance among all detected neighbors. If more than one neighbor has the minimum distance, we assume that the neighbor with minimum index number is selected. We approximate the probability of  $j$  being selected as the next-hop from the distance pmf that we approximated before.

To illustrate our methodology consider the simple 5x5 network grid in Figure 4. Assume that the detection probability for every link is 0.6. We focus on the routing table entry of node 13 for the destination node 1. There are 4 neighbors of node 13 (8, 12, 14, 18) that can be used for packet forwarding. Using the iterative DP equations, we can find the expected delay of the 4 neighbor nodes. These are approximated by the Matlab code and are (4.70, 4.74, 7.42, 7.39) respectively. Because of the symmetry, distances of 8 and 12 should be equal, and the numerical difference is due to the finite number of iterations in the software implementation.

In the next step, we have to approximate the distance pmf for the four neighbor nodes using non-linear equations (8, 9). The minimum distance for node 8 is 3 and maximum is assumed to be 6 and the probabilities for distances from 3 to 6 are (0.19, 0.23, 0.27, 0.31). Similarly, for node 12 the probabilities are (0.18, 0.22, 0.27, 0.33). For nodes 14 and 19 the minimum is 5 and the maximum is 10 and the probabilities are: (0.18, 0.17, 0.17, 0.16, 0.16, 0.16) and (0.18, 0.17, 0.17, 0.16, 0.16, 0.16) respectively.

Using the pmf, we can compute probability that neighbor  $i$  is selected over neighbor  $j$  as the next hop. In this step, we assume that if two neighbors have the same hop count, we select the one with lower index. These probabilities are given in Table 1 for our example.

At any time, a subset of the four links are active and detected in the network. In total there are 16 different possible combination of the active links. For each combination, probability of selecting a particular detected neighbor node  $i$  is proportional to the product of those elements in row  $i$  of Table 1 that corresponds to the detected neighbors (note

**Table 1: Probability of selecting node  $i$  over node  $j$  at node 13 as the next-hop to the destination node 1**

Node $i$	Node $j$			
	8	12	14	18
8		0.64	0.94	0.94
12	0.36		0.94	0.94
14	0.06	0.06		0.58
18	0.06	0.06	0.42	

that the computed products for each combination of the detected neighbors should be normalized so that they sum up to 1).

For instance, let's consider the combination that all links are detected. Probability of this combination is  $0.6^4$ . Probability of selecting each neighbor is proportional to the product of all entries of the corresponding row. Thus, we can compute the probability of having each neighbor selected as the next hop, which are: (0.637, 0.359, 0.002, 0.002). These probabilities are for the combination that all 4 neighbor links are detected. For the final result, we have to compute these probabilities for all possible 16 combinations of detected links, and take the weighted average of them.

## 7. NETWORK MODELING AND DESIGN

Our component based modelling can be used for design and enhancement of the routing protocols performance. In this section, we first illustrate how we can use our models and methodology to enhance and modify the NDC component of the OLSR. Next, we demonstrate how we can study and evaluate the proposed methods impact on the overall performance of the routing protocol.

### 7.1 Enhancement and modification of NDC

In section 4, we presented our Markov Chain Model for NDC. Recall that the input to the Neighborhood Discovery Component (NDC) model is,  $f_{ij}$ , the probability of transmission failure of hello messages between two nodes. The output of this model is probability of having a bidirectional link between two nodes,  $p_{ij}$ . The performance metric for this component is the link detection entropy that we define as:

$$H_{ij} = -p_{ij} \log p_{ij} - (1 - p_{ij}) \log(1 - p_{ij}). \quad (10)$$

It is desirable that the links have predictable and stable status, i.e. the link detection probability should be close to zero or close to one depending on the transmission success probability for the link, or the link entropy should be close to zero for both stable (good) and non-stable (bad) links as defined in section 4. We say that a link is good (bad) if probability of HELLO messages success is above(below) a threshold  $p_u(p_d)$ . Consider that we set  $p_u$  and  $p_d$  to 0.7 and 0.6 respectively. Figure 5(a) and 5(b) show the detection probability for the bad and good links as a function of  $U$  and  $D$  respectively.  $U$  and  $D$  are the NDC component design parameters and should be set appropriately to achieve the desired performance. From the figures it can be seen that  $(U = 20, D = 9)$  is a good design point.

Figure 6(a) and 6(b) show the link detection probability and entropy as a function of success probability for this set of design parameters ( $U=20, D=9$ ). Note that the link entropy always has a maximum equal to one (corresponding to link detection probability equal to 0.5), but it is desirable to have abrupt transition from the maximum value to the values close to zero and the maximum should occur in the transition region between the bad and good link's region, so that entropy is close to zero for both good and bad links. From the figures, it is clear that these goal are achieved by our design.

In OLSR, the NDC detects a link after only one HELLO message is successfully received ( $U = 1$ ). For comparison with the OLSR NDC, we have also plotted the link detection probability and entropy for ( $U=1, D=9$ ), which can be realized with OLSR, in Figures 7(a) and 7(b) respectively. Clearly, the region with large entropy is wider; hence, the probability of having unpredictable links is higher. Furthermore, this region does not separate good and bad links.

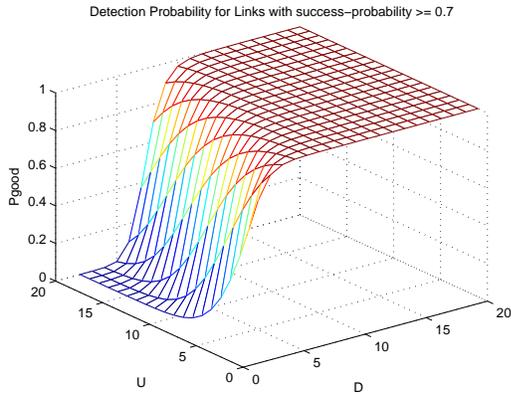
### 7.2 Network level performance evaluation

So far, we have illustrated that the proposed component model can be used to modify and enhance the performance of the NDC component. In this section, we illustrate how the models can be used to study the overall performance of the routing protocol.

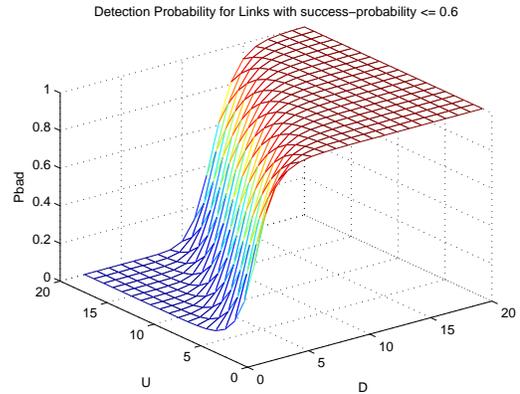
We consider throughput, which is one of the main and primary performance metrics of wireless networks. We illustrate how our models can be used to evaluate and estimate network throughput. We define throughput from node  $i$  to node  $k$ ,  $t_{ik}$  as the ratio of the packets received to the packets transmitted from source  $i$  to destination  $k$ . Recall that  $s_{ij}$  is the probability of successful transmission from node  $i$  to node  $j$ , and  $\alpha_{ij}(k)$  is probability of selecting node  $j$  as the next-hop of  $i$  for destination  $k$ . Thus, for every destination  $k$  and source node  $i$ ,

$$\begin{aligned} t_{ik} &= \sum_{j \neq i} \alpha_{ij}(k)(1 - f_{ij})t_{jk}, \\ t_{kk} &= 1. \end{aligned} \quad (11)$$

We are assuming that there is no MAC layer re-transmission at every node in 11. If the MAC layer retransmits the failed packet up to  $m$  times, then  $f_{ij}$  should be replaced with  $f_{ij}^m$ . We can use a simple fixed point iteration to find the solution of the above equations. We consider again the 5x5 grid network shown in Figure 4; however we increase the number and type of the links that are present in the network. We assume that there 3 types of the links in the network, with successful transmission probabilities of 0.9, 0.6, 0.5 respectively. The corresponding links for node 13 are shown in Figure 8(a), where links to nodes (8, 12, 14, 18), (7, 9, 17, 19), and (3, 11, 15, 23) are type, 1, 2, and 3 respectively. Since type 1 links are more reliable, our design objective is to only use these links with high probability for routing. In section 7.1, we set the NDC parameters  $U = 20, D = 9$ , so that the detection probability for links with success probability lower than 0.6 is close to zero and for links with success probability larger than 0.7 is close to 1. Thus, the same set of NDC parameters are appropriate for this example too, i.e., using only type 1 links for routing with high probability. We used  $(U, D) = (20, 9)$  and for comparison  $(U, D) = (1, 9)$  in our

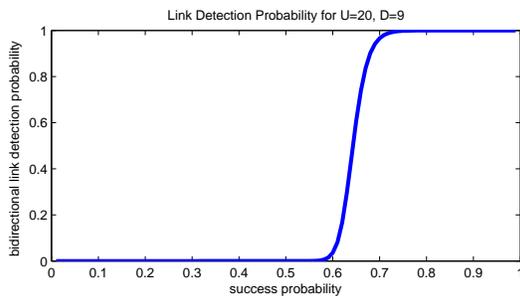


(a) Detection probability for good links v.s.  $U$  and  $D$ .

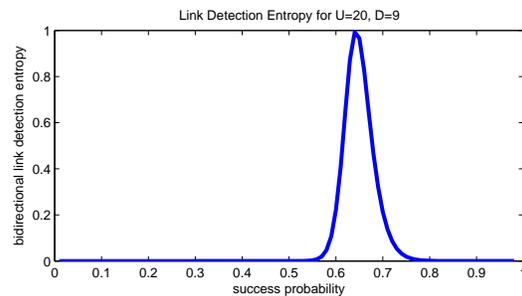


(b) Detection probability for bad links v.s.  $U$  and  $D$ .

**Figure 5: Design curves for  $U$  and  $D$  based on the link detection probabilities.**



(a) Link detection probability.



(b) Link detection probability.

**Figure 6: Link Predictability Metrics Curves for ( $U = 20, D = 9$ ).**

model and derived the next-hop probabilities,  $\alpha_{ij}(k)$ . Next, we used equations 11 to derive the network throughput between every pair of nodes. Figure 8(b) shows the throughput from every node in the network to destination node 25. As we expect, the first set of parameters ( $U, D$ ) = (20, 9) provides better end-to-end result, while the other design that resembles OLSR NDC functionality suffers from using bad links in the design.

## 8. SUMMARY AND FUTURE WORK

In this paper, we presented a modular and component based model for the wireless proactive routing protocols. We used OLSR routing protocol as an example and developed a component based model for this protocol. For the design, we focused on the Neighborhood Discovery Component (NDC) and provide a methodology for design and modification of this component that results in a routing protocol with reliable and predictable performance. We also illustrate how we can use our models to derive and approximate network throughput. We used this methodology to derive the performance of a modified version of OLSR that works based on our proposed NDC and compared it performance with regular OLSR protocol.

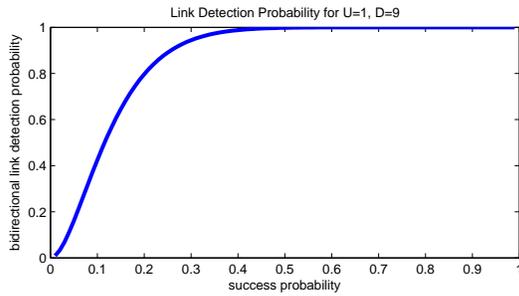
The component based modelling for the routing that we pre-

sented here is one of the main blocks that we are developing for modelling and performance study of wireless networks. The other main blocks are PHY, MAC, scheduling [1] and mobility models. The overarching objective of these projects are development of efficient approximation models for wireless network protocols that can be used for the analysis, design, and optimization of cross-layer protocols and algorithms.

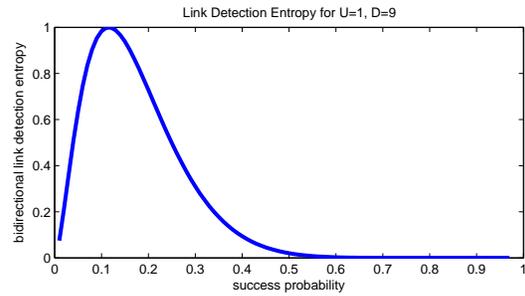
Currently, we are comparing results of our models with simulation models in order to study and enhance the accuracy of our models. As illustrated in Figure 1, we are also in the process of integrating the MAC and PHY models with the routing protocol models that we discussed in this paper. For the design, optimization and sensitivity analysis, we need to compute derivative of the performance metrics with respect to the inputs and network parameters. We have used Automatic Differentiation for that purpose in [1] and we intend to integrate AD software tools with our routing protocol models too.

## 9. REFERENCES

- [1] J. S. Baras, V. Tabatabaee, G. Papageorgiou, and N. Rentz. Modelling and Optimization for Multi-hop Wireless Networks Using Fixed Point and Automatic

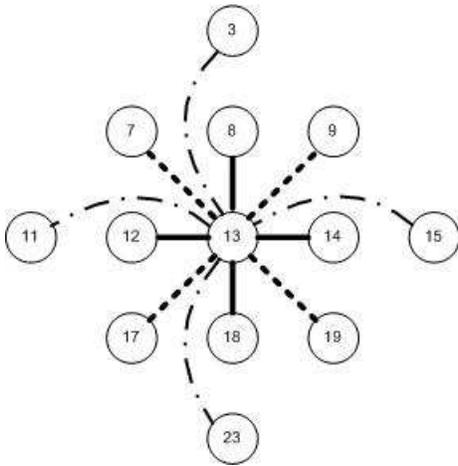


(a) Link detection probability.

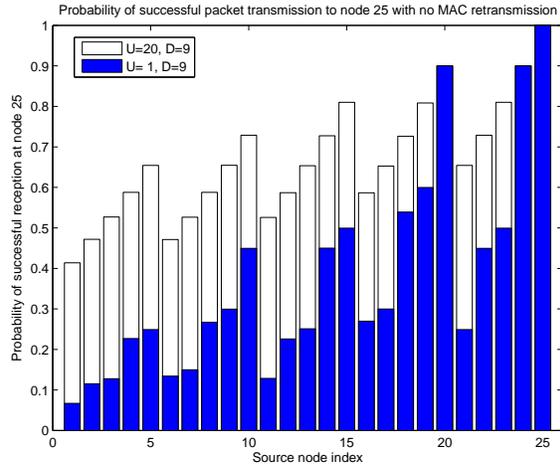


(b) Link detection probability.

Figure 7: Link Predictability Metrics Curves for ( $U = 1, D = 9$ ) that can be implemented with OLSR protocol.



(a) Node 13 Links.



(b) Probability of successful transmission of packets to destination 25.

Figure 8: Connections and performance for the Grid with 3 types of links.

Differentiation. In *Int. Symp. on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks*, 2008.

- [2] T. Clausen and P. Jacquet. Optimized Link State Routing (OLSR). In *IETF RFC 3626*, 2003.
- [3] H. Huang and J. S. Baras. Component Based Routing: A New Methodology for Designing Routing Protocols for MANET. In *Proceedings of the 25th Army Science Conference*, Nov. 27-30, 2003.
- [4] A. Qayyum, L. Viennot, and A. Laouiti. Multipoint Relaying for Flooding Broadcast Messages in Mobile Wireless Networks. *Annual Hawaii Int. Conf. on Systems Science HICSS'02*, 09, 2002.