This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the IEEE ICC 2009 proceedings

Component Based Performance Modelling of Wireless Routing Protocols

John S. Baras , Vahid Tabatabaee, Punyaslok Purkayastha, Kiran Somasundaram
Department of Electrical and Computer Engineering
and The Institute for Systems Research
University of Maryland
College Park, MD 20742
Email: baras, vahidt, punya, kirans@umd.edu

***Abstract*—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, or software. The feasibility of the component based design relies heavily on the compositionality property (i.e. system-level properties can be computed from properties of components). To provide a component based design methodology and to test compositionality for routing protocols, we have to develop a component based model of the wireless network. We present the main components of the routing protocol that should be modelled and focus on 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. Throughout the paper, we use the Optimized Link State Routing (OLSR) protocol as a case study to demonstrate the effectiveness of our approach. Using the neighborhood discovery component, we present our design methodology and design a modified enhanced version of this component, and compare its performance to the original OLSR design.**

I. Introduction

We consider a systems engineering approach based on component based modelling and design principles for wireless networks. This method provides a systematic approach that can be used in the study, analysis, design and optimization of wireless networks. In this approach, the wireless network is modelled as a complex distributed system of systems. The complex system is divided into separate components that are divided into sub-components themselves. The main objective of a component based design is *separation of concerns*, which is achieved by breaking the system into components that overlap as little as possible in functionality. The design principles, interfaces and performance metrics for each component should be specified so that it can be studied and designed separately.

One may view cross-layer design as a component based design of wireless communication networks. However, a fundamental problem with the proposed cross-layer solutions is lack of modularity, which is essential for the design of any complex system. Furthermore, cross-layer design may lead to unintended interaction between layers that can result in instability and unpredictable behavior of the system [4].

In fact, component based design can be considered as a design methodology that addresses these common problems with cross-layer design. We will use the main layers of the communication network as the primary components of the system, and for each layer, we specify its sub-components. We pay special attention to specify the components functions, interfaces, performance metrics, and the relation between the network and the components performance metrics. Note that the main objectives here are: (i) to design components modularly and separately; (ii) to understand and predict network performance based on the characteristics and performance of the components. These objectives are fundamentally different and cannot be met by the recently proposed cross-layer design methodologies [3], [5] that look at wireless network design as an abstract integrated cross-layer optimization problem.

The primary challenges in component based design are: (i) selection of the sub-components and their metrics so that *compositionality* holds, i.e. system-level properties can be computed from local properties of components; (ii) specification of the relation between whole network and individual components performance metrics in the form of deterministic and/or stochastic bounds. The complex distributed nature of wireless networks and the inherit interaction between the cyber system (hardware and software modules) on one hand, and the unpredictable dynamics of the underlying physical system (communication channel) on the other, makes it impossible to develop explicit functional relations between component and system performance metrics to test compositionality.

To test compositionality and perform system analysis and synthesis for wireless networks, we require component based models. It may be possible to develop packet level discrete event simulation models 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 the relation between the network and components parameters and performance metrics. Component based approaches to design can yield both formal and performance models for a system. Formal models can be used to identify protocol defects such as the existence of loops and occurence of deadlocks. Performance models provide a quantitative description of the protocol behaviour. In this work we primarily focus on component based performance models for wireless routing protocols.

The main components of our wireless network model are

978-1-4244-3435-0/09/$25.00 ©2009 IEEE

routing, node scheduling, MAC and PHY layer models. Each component consists of sub-components (i.e. components contained in the original component) and interfaces that can be used to control (inputs), observe internal state of a component (outputs), or evaluate its performance. Each component software is executable code with well defined inputs and outputs that can be connected to other components (perhaps through interface translators) to build up a system software. We use deterministic and probabilistic analytical and numerical methods to model sub-components. The performance metrics of the components are not necessarily the same as the system (network) performance metrics, but they should be related. The relation between network and components performance could be explicit, stochastic, or in the form of upper and lower bounds.

In this paper, we focus on routing and present our method for component based performance modelling of routing protocols. We first identify the main components of wireless routing protocols. We then propose methodologies for modelling each one of these components for the OLSR protocol.

The paper is organized as follows: In section II, we introduce the main components and the high-level model architecture. In section III, we review OLSR, introduce its main components and present our methodology for modelling them. In section IV, we use the component models for the design of routing protocols, and to evaluate performance (throughput) of the designed routing protocols. In section V, we summarize the results and discuss future research directions.

II. Performance Model Architecture

In this section, we introduce the main components of our performance model, their inputs, outputs, and interconnections. Consider a network that consists of N nodes, $i = 1, \cdots, N$. The probability of successful packet transmission from node i to node j is s_{ij}, and $f_{ij} = 1 - s_{ij}$ is the 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 five main components in our performance model for routing protocols: (i) Neighborhood Discovery Component (NDC), (ii) Selector of Topology Information to Disseminate Component (STIDC), (iii) Topology Dissemination Component (TDC), (iv) Route or Path Selection Component (RSC), Packet Forwarding Component (PFC). In this paper, we present methodologies for modelling the three components NDC, STIDC, 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.

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 nodes i and j, for $i, j = 1, \cdots, N$. 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 $i, j = 1, \cdots, N$, which is the detection probability of bidirectional links among nodes i and j.

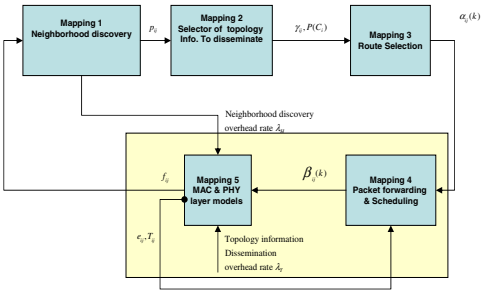


Fig. 1. Architecture of the Component Based Model

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, which 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 the states 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 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 selects a subset of the connected links of a node to disseminate their information. We say the 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. 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. The main task of the 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. 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 $\alpha_{ij}(k)$.

III. OLSR Components Modelling

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. For details on the operation of OLSR, refer to [2]. In this section, we describe the OLSR functionality based on the

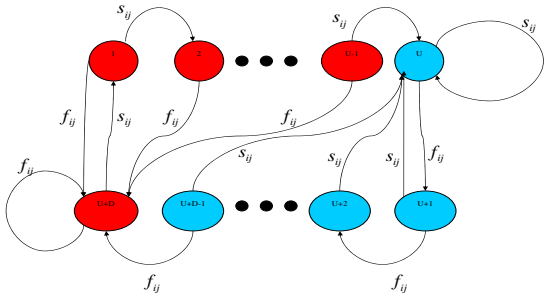


Fig. 2. Markov chain model for the NDC detection mechanism

component based architecture that was presented in section II, and use it to illustrate our modelling approach for each one of the main components studied in this paper.

A. NDC Modelling

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 is 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 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's ID, 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.

We model the NDC as a Finite State Machine (FSM), so that we can use Markov chain methods 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 the 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. Therefore, in OLSR we can only control the D parameter by changing NHT.

Let π_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:

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

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:

$$p_{ij} = q_{ij}^2. \tag{2}$$

The details of the global balance equations that are used to derive the steady state probabilities are given in [7]. The design (or control) parameters for NDC are the 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 examples. We can use 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 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 traffic overhead, contention and congestion in the network, and can also cause disruption in forwarding 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 protocol. 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 links status are with high probability predictable. Thus, the transition region should be as narrow as possible. In section IV, we 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 IV, with respect to link status predictability, OLSR does not provide us enough control.

B. STIDC Modelling

The STIDC 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 generate 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 [6]: At every node i, the algorithm starts with an empty set $MPR(i)$. It first selects as MPR the neighbors, 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 the largest number of not covered second order neighbors to its MPR list.

The selected MPR nodes depend on the detected links, which are random variables. Therefore, the selected MPR nodes are also random processes. The state of node i is C_i, which is a binary vector with Δ_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 component model output 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. 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 performance of the Monte-Carlo based estimator is compared with exhaustive search in [7].

C. 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 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 processes. 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 k.

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 in 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. Hence, we 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 [7].

Let $h(i,k,C_i)$ denote 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 the probability of having connection to the destination is equal to the probability of having a detected link (and not having the 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:

$$\begin{aligned} h(i,k,C_i) &= 1 + \min_{j \in MPR(C_i)} (h(j,k)) \\ h(i,k) &= \sum_{C_i} P(C_i) h(i,k,C_i) \end{aligned} \tag{3}$$

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

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 subsection, 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 assume that the maximum distance between two nodes is,

$$M(i,k) = \max\left(2D(i,k) - m(i,k), 2m(i,k)\right) \quad (4)$$

Then, we approximate the pmf of this r.v. using maximum entropy method [7]. Let p_j be the probability that the distance from i to k is j hops. Using Lagrange multipliers, we have

$$p_j = e^{-\mu j} / \sum_{x=m(i,k)}^{M(i,k)} e^{-\mu x}, \quad (5)$$

where μ is the solution of the following equation:

$$\sum_{j=m(i,k)}^{M(i,k)} j e^{-\mu j} / \sum_{x=m(i,k)}^{M(i,k)} e^{-\mu x} = D(i,k). \quad (6)$$

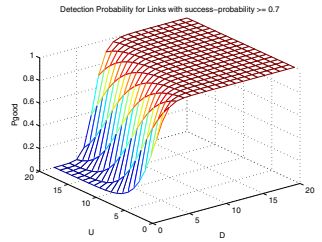
3) Next-hop Probability Estimation: After computation of the distances pmf, we can approximate the probability of selecting node j as the next hop for node i to destination k. For every node, we first list all possible combinations of its detected links. If a node has Δ nodes in its communication range, there are 2^{Δ} possible combinations of its detected links. The probability of each combination can be computed using the information provided from the NDC assuming that links status are independent random variables. For each combination, the 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.

IV. Network Modeling and Design

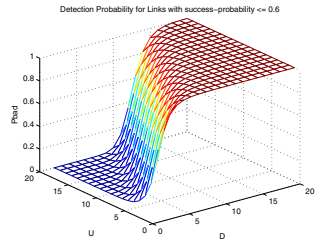
Our component based modelling can be used for design and enhancement of routing protocols performance. In this section, we first illustrate how we can use our models and methodology to enhance and modify the NDC component of the OLSR. Next, we study and evaluate the proposed method's impact on the overall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

In section III-A, we presented our Markov chain model for NDC. Recall that the input to the NDC model is, f_{ij}, the probability of transmission failure of HELLO messages between two nodes. The output of this model is the probability of having a bidirectional link between two nodes, p_{ij}.

It is desirable that the links have predictable and stable status, i.e. the link detection probability should be close to zero for bad links and close to one for good links. We say that a link is good (bad) if the 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 3(a) and 3(b)

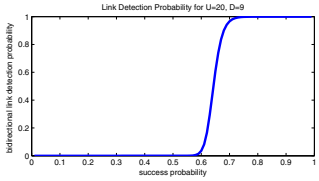


(a) Detection probability for good links v.s. U and D.

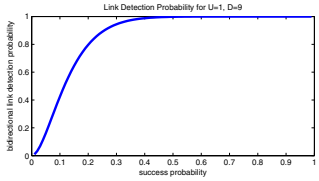


(b) Detection probability for bad links v.s. U and D.

Fig. 3. Design curves for U and D based on the link detection probabilities.



(a) Design parameters $(U = 20, D = 9)$



(b) Design parameters $(U = 1, D = 9)$

Fig. 4. Link Detection Probability

show the detection probability for the bad and good links as a function of U and D respectively. U and D are the NDC design parameters and should be set appropriately to achieve the desired performance. From the figures above it can be seen that $(U = 20, D = 9)$ is a good design point.

Figure 4(a) shows the link detection probability for $(U = 20, D = 9)$. Note that there is always a transition region from link detection probability 0 to 1, but it is desirable to have abrupt transition and a narrow transition region. From the figures, it is clear that these goals are achieved by our design $(U = 20, D = 9)$. 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 for $(U = 1, D = 9)$, realized with OLSR, in Figure 4(b). Clearly, the transition region is

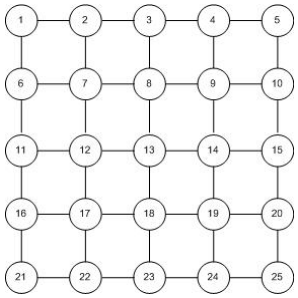


Fig. 5. A simple 5x5 Grid Topology

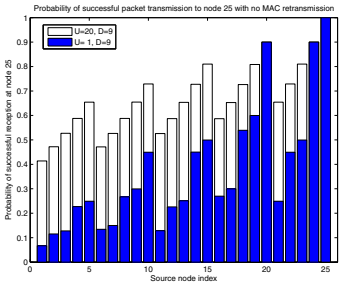


Fig. 6. Connections and performance for the Grid with 3 types of links

wider; hence, the probability of having unpredictable links is higher. Furthermore, this region does not separate good and bad links.

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

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. It can be computed recursively for every destination k and source node i as follows:

$$t_{ik} = \sum_{j \neq i} \alpha_{ij}(k)(1 - f_{ij})t_{jk}, \quad t_{kk} = 1. \tag{7}$$

We assume that there is no MAC layer re-transmission at every node in eq. (7). If the MAC layer retransmits the failed packets 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 eq. (7).

We consider the 5x5 grid network shown in Figure 5. We assume that there are 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 links to nodes (8, 12, 14, 18), (7, 9, 17, 19), and (3, 11, 15, 23), which are type, 1, 2, and 3 respectively. Type 1 links are more reliable, and our objective is to only use these links with high probability. Recall that we have already shown that for the NDC parameters $U = 20, D = 9$, 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. Hence, this set of NDC parameters are appropriate for this example too. We used $(U, D) = (20, 9)$, and for comparison $(U, D) = (1, 9)$ in our model and derived the next-hop probabilities, $\alpha_{ij}(k)$. Next, we used equations (7) to derive the network throughput between every pair of nodes. Figure 6 shows the throughput from every node in the network to destination node 25. As we expected, the first set of parameters $(U, D) = (20, 9)$ provides a better end-to-end result, while the other design that resembles OLSR NDC functionality suffers from using bad links in the design.

V. Summary and Future Work

In this paper, we presented a modular and component based model for wireless proactive routing protocols. We used the OLSR routing protocol as an example and developed a component based model for it. For the design, we focused on the Neighborhood Discovery Component (NDC) and provided a methodology for design and modification of this component that results in a routing protocol with reliable and predictable performance. We illustrated how we can use our models to derive and approximate network throughput. The component based modelling for routing we presented 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. Our overarching objective is the 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.

Acknowledgment

Research supported by the Communications and Networks Consortium sponsored by the U.S. Army Research Laboratory under the Collaborative Technology Alliance Program Cooperative Agreement DAAD19-01-2-001.

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 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] L. Georgiadis, M. Neely, and L. Tassiulas. Resource Allocation and Cross-Layer Control in Wireless Networks. In *Foundations and Trends in Networking, NOW Publishers Inc.*, March 2006.

[4] V. Kawadia and P. R. Kumar. A Cautionary Perspective on Cross-layer Design. In *IEEE Wireless Communication Magazine*, volume 12, pages 3–11, Feb., 2005.

[5] X. Lin, N. Shroff, and R. Srikant. A Tutorial on Cross-Layer Optimization in Wireless Networks. In *IEEE Journal on Selected Areas in Communications*, pages 1452–1463, Aug, 2006.

[6] 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.

[7] V. Tabatabaee, J. S. Baras, P. Purkayastha, and K. Somasundaram. Component Based Modeling and Design of Wireless Ad-hoc Routing Protocols. In *ISR technical report, University of Maryland*, 2008.