Component Based Modeling for Cross-layer Analysis of 802.11 MAC and OLSR Routing Protocols in Ad-hoc Networks

John S. Baras, Vahid Tabatabaee and Kaustubh Jain
Department of Electrical and Computer Engineering
University of Maryland
College Park, MD 20742
Email: baras@isr.umd.edu, vahidt@umd.edu, ksjain@umd.edu

***Abstract*—We present a complete scenario driven component based analytic model of 802.11 MAC and OLSR routing protocols in MANETs. We use this model to provide a systematic approach to study the network performance and cross-layer analysis and design of routing, scheduling, MAC and PHY layer protocols. The routing protocol is divided into multiple components. Componentization is a standard methodology for analysis and synthesis of complex systems. To provide a component based design methodology, we have to develop a component based model of the wireless network that considers cross-layer dependency of performance. The component based model enable us to study the effect of each component on the overall performance of the wireless network, and to design each component separately. For the MAC layer, we use a fixed point loss model of 802.11 protocol that considers effects of hidden nodes and finite retransmission attempts. We have also considered simple models for PHY and scheduling. The main focus of this paper is on integration of these models to obtain a complete model for wireless networks.**

In several scenario driven studies, with user-specified topologies and traffic demands, we study the performance metrics - throughput and delay. By analyzing the performances under varying network scenarios, we are able to identify a few sources of performance degradation. We also study the effect of certain design parameters on the network performance. Thus, demonstrating the ability of the model to quickly identify problem components and try alternative design parameters.

I. INTRODUCTION

A large number of routing protocols have been proposed for MANET. However, to date most of performance studies are based on discrete event simulations, with very few exceptions. Routing protocols are commonly implemented as large monolithic software, which are very difficult to adapt to varying conditions and mission scenarios. Identifying the performance bottlenecks by these simulations is very time consuming, since we have to run a large number of scenarios to pin-point the bottleneck. Because of the complex nature of the simulations, they provide little insight on design parameter sensitivities or on how we can improve performance and adaptivity of these routing protocols. Furthermore, performance of routing protocol depends a lot on the MAC protocol. Since the process of gathering topology information depends on topology packet flooding, performance evaluation of a routing protocol cannot be done correctly without considering the underlying MAC. Similarly, the performance of a MAC protocol depends on the traffic load offered and the routing used to handle the traffic. Hence, to model the wireless network performance correctly, cross-layer interaction between the routing and MAC protocols has to be considered.

The interaction and overall performance of the MAC and routing protocols in an ad-hoc network can be modeled by dividing the various parts of the protocols into components. This approach was introduced in [1]. In this approach a proactive routing protocol is divided into three main components - (i) Neighbor Discovery Component (NDC), (ii) Selector of Information for Dissemination Component (STIDC), and (iii) Route Selection Component (RSC). These routing components are coupled with the Scheduler and the MAC components. Figure 1 illustrates the interdependence of the various components. While the performance of the components are coupled, each of them can be independently evaluated and designed for better performance.

In [1], Baras et. al. describe and develop the routing components - namely NDC, STIDC and RSC, based on the OLSR protocol [2]. This approach of component based routing was first introduced in [3]. In his seminal work, Bianchi [4] gave an analytic model for the throughput performance of 802.11 MAC, but considered saturated users with ideal channel conditions and no hidden node problems. In [5] and [6], Hira et. al. extend the work of [4] to address the above issue, but consider traffic only along disjoint paths. In [7] and [8], Baras et. al. generalize the MAC models to enable throughput and delay computation along multiple paths with common nodes. In this paper, we combine the different components of [1] and [7], and focus on the interaction between them. While [1] considers fixed MAC loss probabilities for the links, [7] assumes pre-specified static routes. In this paper we specify appropriate modifications to the components and close the loop, taking cross-layer effects into account. We then describe a fixed point algorithm to find a consistent set of solution.

The combined component models *approximate* the per-

978-1-4244-5239-2/09/$26.00

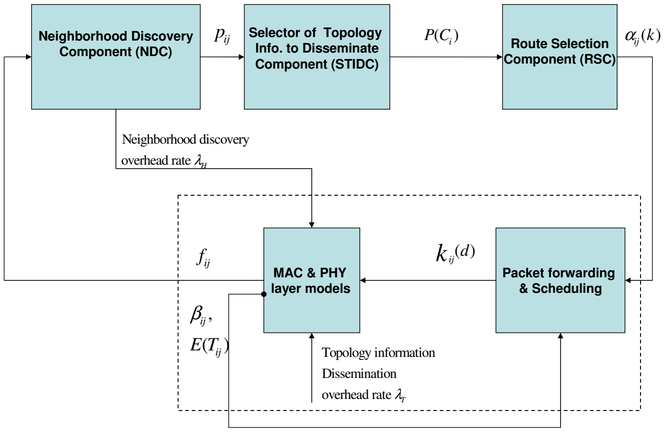


Fig. 1: Components for topology driven performance modeling

formance of routing and MAC protocols, but enable us to evaluate and analyze the cross-layer interaction under various network topologies. For specified static network topology, traffic demands and physical layer losses, the models evaluate the performance metrics of the various components and find out network performance metrics - average throughput and average end-to-end delay for each connection. By analyzing the performances under varying network scenarios, we can identify the sources of performance degradation and try to improve the corresponding components. We can also study the trade-offs and choose the component design parameters appropriately. To illustrate, we study the performance of a few network topologies under varying design parameters for NDC.

The paper is organized as follows: In section II, we describe the routing components based on OLSR and their extensions. In section III, we describe the 802.11 MAC models. Section IV gives the details about the fixed point algorithm to integrate the MAC and routing performance models. In section V, we use the component models to evaluate network performance and study the effects of certain design parameters. We conclude with a summary in section VI.

II. THE ROUTING COMPONENTS

A proactive routing protocol in MANET can be divided into three tasks - (i) gather local neighborhood information, (ii) flood a pruned version of the local information, and (iii) select routes based on the available information. These parts are abstracted into - (i) Neighbor Discovery Component (NDC), (ii) Selector of Information for Dissemination Component (STIDC), and (iii) Route Selection Component (RSC), respectively.

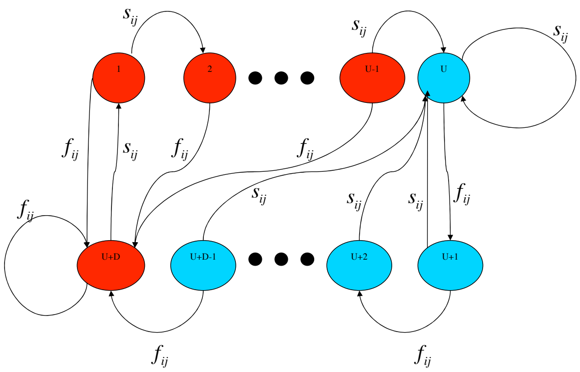


Fig. 2: FSM of neighbor detection mechanism

In this section, we describe the NDC performance model, how it depends on outputs from the MAC model, and describe the design parameters for the NDC. We refer the reader to [1] for a detailed exposition of modeling approaches to STIDC and RSC components, but give a short description of their interplay with the MAC model.

A. NDC

The neighbor discovery methods in ad hoc networks are usually driven by proactive HELLO packet broadcasts. Based on the reception of these HELLO packets, node identify their neighbors. Since these packets are susceptible to channel and contention losses, the local neighborhoods as seen by the nodes depends upon the underlying link loss probabilities.

Based on the link detection criteria and link loss probabilities, a state space model for these neighbor discovery methods has been introduced in [1]. A node declares a new unidirectional link as *up* if it receives U consecutive HELLO messages and declares a unidirectional link as *down* if it does not hear D consecutive HELLO messages. In OLSR [2], the values of U and D will correspond to 1 and 3 respectively.

The Finite State Machine (FSM) for the neighbor discovery algorithm is shown in Figure 2. Each station i in the network runs the FSM for every station j in its radio range. Whenever this station receives a HELLO packet from j it corresponds to a decision edge $s_{i,j}$ and when the HELLO packet is lost in transmission it corresponds to a decision edge $f_{i,j}$. Station i declares a unidirectional link $i \rightarrow j$ if it is in any one of states from U to $U + D - 1$. Otherwise it declares a unidirectional link failure. This FSM then forms the *executable model* for the NDC in system design.

The corresponding *performance model* which captures the steady state behaviour of the FSM can be obtained using Markov chain analysis. The inputs to this model are the probabilities of success ($s_{i,j}$) and failure ($f_{i,j} = 1 - s_{i,j}$) in transmission of HELLO messages. These inputs, which represent the link loss probabilities, come from the MAC model.

Let $\pi_k(f_{i,j})$ be the steady state probability that the NDC is in state k of the Markov chain, defined as a function of $f_{i,j}$. We can use the generalized global balance equations to derive the steady state probabilities. One of the main NDC performance metrics is the probability of detecting a directional link to node j at node i and is given by:

$$q_{i,j} = \sum_{k=U}^{U+D-1} \pi_k(f_{i,j}) \tag{1}$$

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:

$$p_{i,j} = q_{i,j} \cdot q_{j,i} \tag{2}$$

Thus, even for given link loss probabilities, the performance metric - probability of detecting a bidirectional link, depends on the the choice of U and D. Hence they can be set to achieve the desired performance and form the design (or control) parameters for NDC.

B. STIDC and RSC

The STIDC specifies the topology graph that is presented to the network nodes. For each node, STIDC determines a set of important links which are broadcast in the network. Because of losses in HELLO packets, the local neighborhood information changes even in a static topology and so do the links that are broadcast by STIDC. RSC then uses this subset of links to find out routes to the various destination nodes. Again, because of changing topology information, the routes keep changing.

In terms of the performance models, the probabilities of bidirectional link detection from NDC form the input to STIDC. The output of the STIDC is the probability mass function $P(C_i)$ for the state vectors C_i determining the set of important links for node i. Outputs from STIDC and NDC are used by the RSC to find routing probabilities at the nodes for various destinations. The routing probability $\alpha_{i,j}(k)$ specifies the probability of node i choosing node j as the next hop to destination k. In a way, $\alpha_{i,j}(k)$ is the fraction of time node i sees a graph in which j is the next hop on the shortest path to k. Depending on the outputs of NDC and STIDC, a node i may not find any path to the destination k. This amounts to $\sum_{\forall j} \alpha_{i,j}(k) \neq 1$, and the difference $1 - \sum_{\forall j} \alpha_{i,j}(k)$ representing the fraction of time, there is no path from i to k. Similarly, the NDC output may be such that there is no path from i to k at any time, making $\alpha_{i,j}(k) = 0, \ \forall j$. We refer the reader to [1] for details about these component models.

The performance models of STIDC and RSC do not depend directly on MAC, but are affected indirectly through NDC. Depending upon the MAC layer losses, the probability of detecting bidirectional links at NDC will change, thereby changing the performance of STIDC and RSC. For example, if all the links have high MAC layer losses, the network as seen after NDC and STIDC will seem disconnected most of the time, and the routing probabilities will be low.

The cross-layer effect takes place in the reverse direction through the routing probabilities. The output from RSC is an input to the Scheduler and MAC models, and hence directly affect their performance. We describe the Scheduler and MAC models in the next section.

III. THE MAC COMPONENTS

The performance of any MAC protocol can be broadly divided into (i) Scheduler modeling, which determines the amount of traffic demand on each link, and (ii) MAC modeling which determines the actual traffic flow, link delay and link losses, etc. Detailed Scheduler and MAC models for 802.11 MAC are developed in [7]. However, they assume pre-specified static routes. Since in our model, the routes are indirectly specified in-terms of routing probabilities, we need to modify the scheduler and MAC equations. In this section we briefly describe the models focusing on the relevant part, while referring the reader to the originals for more details.

A. Scheduler Modeling

Given the input traffic demand for various source-destination pairs (identified by unique connection IDs), and the routing probabilities for each destination (from RSC), the scheduler model finds out the average traffic at each node and the fraction of time spent by the MAC serving a particular connection traffic.

The average rate of packets that are served at source node i for connection d using node j as the next hop is denoted by $k_{i,j}(d)$ and is given by the following relationship:

$$k_{i,j}(d) = \begin{cases} \lambda_{i,j}(d) & \text{if } \sum_{\forall j} \sum_{\forall d} k_{i,j}(d) \cdot E(T_{i,j}) \leq 1 \\ \frac{\lambda_{i,j}(d)}{\sum_{\forall j} \sum_{\forall d} k_{i,j}(d) \cdot E(T_{i,j})} & \text{otherwise} \end{cases} \tag{3}$$

where $\lambda_{i,j}(d)$ denotes the arrival rate of packets at node i for connection d that will use node j as the next hop. $E(T_{i,j})$, is the average service time to send a packet from node i to node j. We have two possibilities when scheduling packets at node i: Either the utilization of node i is less than 1, and we can serve all incoming packets (as is described in the first line of Equation (3)) or we have to normalize the scheduler coefficients by the utilization of the node (as is described in the second line of the equation). Here we have assumed that all outgoing traffic receive proportional link resources. Then, the total rate of packets served at source i for next hop j (for all connections) is given by:

$$k_{i,j} = \sum_{\forall d} k_{i,j}(d) \tag{4}$$

The net arrival rate of packets at node i for connection d is given by:

$$\lambda_i(d) = \begin{cases} \mathit{input\ rate} & \text{if } i \text{ is the source for connection } d \\ \sum_{\forall l} k_{l,i}(d) \cdot (1 - \beta_{l,i}^{m}) & \text{otherwise} \end{cases} \tag{5}$$

where $\beta_{i,j}$ is the probability of transmission failure at the MAC layer for the link $i \rightarrow j$. m represents the maximum number of retries at the MAC before the packet is dropped at the MAC. Each term in the second line of Eq. (5) gives the traffic received at node i from node l, which is the scheduler rate at node l times the probability of a successful transmission over the link (l, i). Summing over all such nodes l, we get the total traffic at node i. Now, depending on the routing probabilities, node i will route the packets through node j. Thus, the arrival rate of packets at node i meant for next hop j is given by:

$$\lambda_{i,j}(d) = \alpha_{i,j}(dest(d)) \cdot \lambda_i(d) \tag{6}$$

where as described in section II-B, $\alpha_{i,j}(dest(d))$ is the probability of node i choosing node j as the next hop for destination $dest$ of connection d.

The fraction of time that a node i is serving a packet going through node j for connection d is:

$$\rho_{i,j}(d) = k_{i,j}(d) \cdot E(T_{i,j}) \tag{7}$$

And, the fraction of time node i is serving packets for next hop j:

$$\rho_{i,j} = \sum_{\forall d} \rho_{i,j}(d) \tag{8}$$

B. MAC Modeling

Depending upon the traffic on each link (given by $k_{i,j}$ and $\rho_{i,j}$) the MAC equations find out the probability of MAC layer transmission failure $\beta_{i,j}$, and packet service time $E(T_{i,j})$. The MAC equations model the actual traffic on the link by taking into account different parameters like contention for channel, transmission by hidden nodes, back-offs on collision, etc. We use the same equations as in [7], except for a slight change in notation. While [7] indexes each of the link variables by current node i and path p, we index the link variables by the directional link (i, j). This is required due to the modified routing model - the routes are no longer specified by pre-defined paths, but defined through the routing probabilities from RSC. The MAC component currently does not take contention and collision, due to topology control and HELLO packets, into account. We assume these control packets do not cause any significant increase in contention and collision. This assumption is justified as they cause insignificant amount of traffic for high capacity MAC and PHY models such as 802.11.

C. MAC FPA

The various MAC variables are defined implicitly. The traffic demand on each link $k_{i,j}(d)$ in the scheduler component, depends on the link metrics $\beta_{i,j}$ and $E(T_{i,j})$, while the link metrics themselves depend on traffic demand on the links. To find a consistent solution for the MAC variables, we use a fixed point algorithm. We call this the *MAC FPA*. Given the input traffic demand $\lambda_i(d)$ for the source nodes for each connection d and the routing probabilities $\alpha_{i,j}(k)$, we initialize the MAC variables appropriately and iterate over the MAC model equations till we reach a fixed point. Again, we refer the reader to [7] for more details.

D. Performance metrics

$\lambda_i(d)$, $\beta_{i,j}$ and $E(T_{i,j})$ form the main performance metrics of the MAC model. We use $\lambda_i(d)$ and $E(T_{i,j})$ to find out network performance metrics - throughput and delay (described later in Section IV). The link loss probability $\beta_{i,j}$, although determined for data packets, gives a good approximation for the link loss probability for HELLO packets because they both use the same MAC protocol. Hence in our model, we currently assume $f_{i,j} = \beta_{i,j}$, that is, the probability of failure of HELLO packet transmission over a link is the same as the probability of MAC layer transmission failure. Thus, the performance metric of MAC, $\beta_{i,j}$, forms the input $f_{i,j}$ to NDC and closes the loop (Fig. 1).

IV. INTEGRATION OF ROUTING AND MAC COMPONENTS

In the previous sections, we described how outputs from one component are fed to others (Fig. 1). In particular, we described how the routing components take inputs from the MAC components and the MAC components in turn, take inputs from routing components. To determine a set of consistent solutions for the various component outputs, we run another fixed point algorithm on this outer loop. We call this *Outer FPA*.

We start the loop assuming zero link loss probabilities $\beta_{i,j}$ for nodes within radio range. Given the link loss probabilities, we calculate the routing probabilities $\alpha_{i,j}(k)$ from the routing components. With input traffic demand and these routing probabilities, the MAC components (Inner FPA) calculate the new $\beta_{i,j}$, which form the input $f_{i,j}$ to the NDC. We iterate over this loop till we reach a fixed point. We say a fixed point is reached when link loss probabilities for all the links have converged. To avoid oscillations, we use some memory η on $\alpha_{i,j}(k)$ and $\beta_{i,j}$, and update the values in each iteration as:

$$\alpha_{i,j}^{new}(k) = \eta \cdot \alpha_{i,j}^{old}(k) + (1 - \eta) \cdot alpha_{i,j}^{current}(k) \tag{9}$$

$$\beta_{i,j}^{new} = \eta \cdot \beta_{i,j}^{old} + (1 - \eta) \cdot \beta_{i,j}^{current} \tag{10}$$

After reaching the fixed point for the Outer FPA, we get the desired performance metrics of the various components. Additionally, we obtain the throughput and end-to-end delay for each source-destination pair. Throughput is calculated as $\lambda_{dest}(d)/\lambda_{src}(d)$ where src is the source node and $dest$ is the destination node for connection d. To calculate the end-to-end delay, we first find average delays over each link $LD_{i,j}$ using the link layer statistics from the MAC component. (We assume an M/M/1 queuing model at each node, with arrival rate λ_i obtained from $\lambda_{i,j}(d)$ and service time obtained from $E(T_{i,j})$). Then for each connection, the end-to-end delay $D_{src,dest}$ is calculated using the equations :

$$D_{i,dest} = \sum_{\forall j} \alpha_{i,j}(dest) \cdot (D_{j,dest} + LD_{i,j}) \tag{11}$$

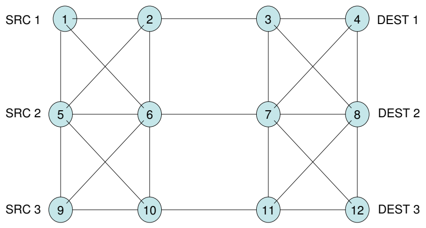


Fig. 3: Physical layer topology of the 12 node network

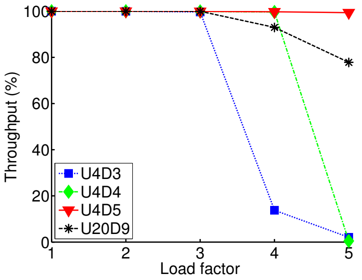


Fig. 4: Total Throughput for the 12 node network

V. Simulation Results

We run the FPA on two sample topologies for varying input parameters. The first set of simulations look at a simple network, but demonstrate the effect of NDC design parameters U and D. The second set of simulations look at a more complicated network and analyze the cross-layer effects.

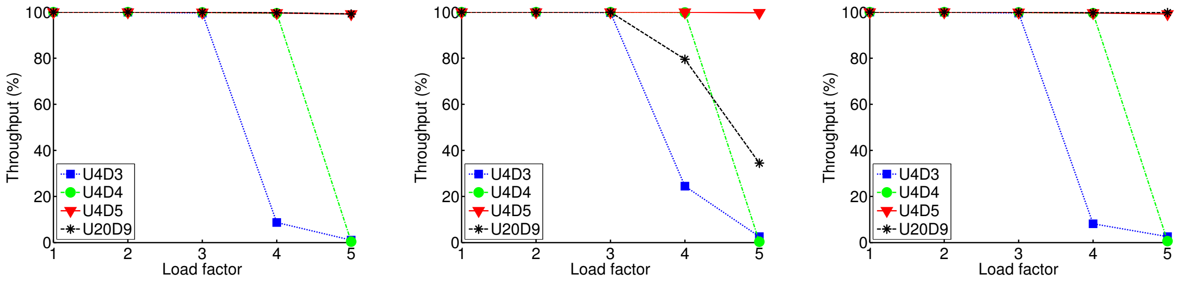
At first, we consider a 12 node topology with 3 connections (Fig. 3). For varying NDC parameters U and D, we run the simulations for increasing load on each connection. Fig. 4 shows the overall throughput for the network for increasing load factors and parameterized by U and D. Fig. 5 shows the throughput for each connection and Fig.6 shows the corresponding end-to-end delays. In these figures, a load factor of k corresponds to each connection having a traffic demand of $k{*}100\ kbps$. We see that as we go on increasing the offered load, the throughput for each connection goes on decreasing and delay keep increasing as expected. Furthermore, for some load factors, the network performance metrics are very sensitive to the values of U and D.

Focusing on the scenarios with NDC parameters $U = 4$ and $D = 3, 4, 5$, we recall that a low value of D corresponds to faster discarding of the existing bidirectional links, similarly a higher value of D corresponds to retaining the links for longer time. For a low value of D, even a slight increase in load will increase the MAC loss probabilities, and the existing links will be discarded quickly by NDC. This explains the faster drop in throughput for the low value of $D = 3$. Similarly, higher values of D will ensure that the links are retained for a longer time. Hence better throughput for high $D = 5$. [1] proposed setting $U = 20$ and $D = 9$, for detecting stable links and discarding unstable links with high probability. For $U = 20$ and $D = 9$, we find that the network is able to carry more load for connections 1 and connection 3, but the carried load for connection 2 drops significantly. This is because, the link loss probabilities from node SRC2 to its neighbors even though only slighty higher, are high enough for the links to be not discovered by its NDC. Performance degradation in end-to-end delays follows similar trends. As the offered load increases, the increase in contention and collision, increases the delays. The drastic increase in end-to-end delay for low throughput scenarios is due to large number of retries and significant buffering.

Next, we run the simulations for a more complicated 20 node scenario (Fig. 7) with 10 connections. Fig. 8 shows the throughput for 2 of the 10 connections and the overall throughput. As before, the scale load factor is used to uniformly increase the traffic demand on each connection. The values of U and D are fixed to be 2. The two connections chosen in Fig 8 achieve the best and the worst throughput across the various scenarios. Connection 1, which experiences the worst throughput corresponds to one of the longest connections in the network. Similarly connection 9, which experiences the best throughput corresponds to one of the shortest connections. Table I gives the connection throughputs for the scenario with scale load factor of 1.25. From the table, we can infer the following: short connections (connections 4 and 9) or connections which can route from low traffic region (connection 3) achieve high throughput; short connections in the heavy traffic region or long connections (connections 2, 5, 7, 8, 10) have moderate throughput; and long connections which pass through heavy traffic region (connections 1 and 6) achieve the lowest throughputs. These results are consistent with the expected network performance taking cross-layer effects into account. Such insights in the network performance would not have been possible by using just the routing layer models, with fixed MAC parameters.

VI. Summary and Future Work

In this paper, we presented a component based model for analysis and design of MAC and routing protocols for wireless networks. Taking cross-layer effects into account, we combined the performance models of 802.11 MAC [7] and generalized proactive routing protocol [1] based on OLSR. We proposed a fixed point algorithm to find a consistent solution

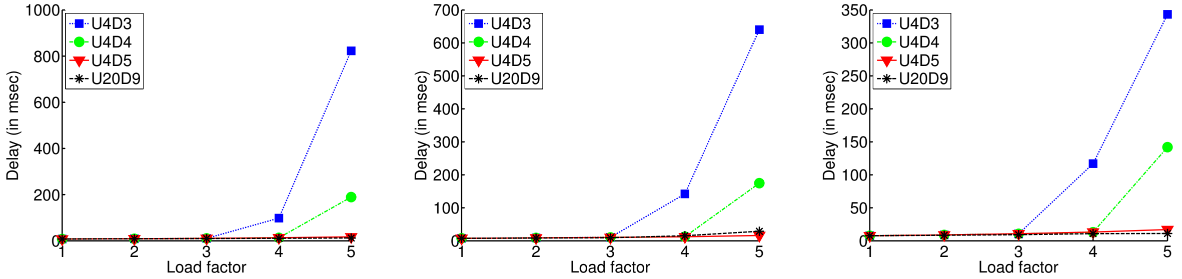


(a) Connection 1

(b) Connection 2

(c) Connection 3

Fig. 5: Connection Throughputs for the 12 node network



(a) Connection 1

(b) Connection 2

(c) Connection 3

Fig. 6: Delays for the 12 node network

Connection	**(Source, Destination)**	**Throughput (in %)**
1	(2,19)	8.1
2	(2,4)	69.9
3	(3,10)	90.5
4	(5,7)	95.8
5	(8,6)	57.1
6	(11,2)	12.1
7	(15,18)	37.3
8	(17,12)	43.3
9	(18,19)	99.6
10	(20,13)	42.8

TABLE I: Connection Throughputs for the 20 node scenario with scale load 1.25

for the interdependent component outputs and performance metrics. We studied the performance of the routing and MAC protocols under varying network settings and showed how the performance of one depends a lot on the other. Lastly, we demonstrated the effects of component design parameters - U and D for NDC on network performance. As a next step towards wireless network design, we plan to study the effects of other component design parameters and perform their sensitivity analysis. Using tools like Automatic Differentiation (AD), we can compute the gradient of the performance metrics along various design parameters and find

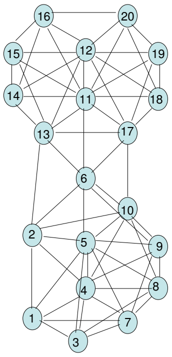


Fig. 7: Physical layer topology of the 20 node network

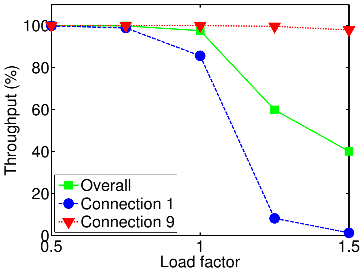


Fig. 8: Throughputs for the 20 node network

out their sensitivity in a much faster and insightful way. Using these generalized models, we can analyze the network performance under varying scenarios to understand the impact of the various components and work on better designing the components.

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. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U. S. Government.

REFERENCES

[1] J. S. Baras, V. Tabatabaee, P. Purkayastha, and K. Somasundaram, "Component based performance modeling of the wireless routing protocols," in *IEEE ICC Ad Hoc and Sensor Networking Symposium.*, 2009.

[2] T. Clausen, P. J. (editors), C. Adjih, A. Laouiti, P. Minet, P. Muhlethaler, A. Qayyum, and L.Viennot, "Optimized link state routing protocol (OLSR)," RFC 3626, October 2003, network Working Group. [Online]. Available: http://ietf.org/rfc/rfc3626.txt

[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] G. Bianchi, "Performance Analysis of the IEEE802.11 Distributed Coodination Function," *IEEE Journal on Selected Areas in Communications*, vol. 18, March 2000.

[5] K. Medepalli and F. Tobagi, "Towards Performance Modeling of IEEE802.11 Based Wireless Networks: A Unified Framework and its Applications," in *Proceedings of the IEEE Conference on Computer Communications (INFOCOM)*, 2006.

[6] M. Hira, F. Tobagi, and K. Medepalli, "Throughput Analysis of a Path in an IEEE 802.11 Multihop Wireless Netwoks," in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC)*, 2007.

[7] J. S. Baras, V. Tabatabaee, G. Papageorgiou, and N. Rentz, "Performance Metric Sensitivity Computation for Optimization and Trade-off Analysis in Wireless Networks," in *Proceedings of the IEEE Global Communications Conference*, 2008.

[8] J. S. Baras, S. Perumal, V. Tabatabaee, K. Somasundaram, and P. Purkayastha, "Loss Network Models and Multiple Metric Performance Sensitivity Analysis for Mobile Wireless Multi-Hop Networks," in *Fourth International Wireless Internet Conference*, 2008.