Pervasive and Mobile Computing 7 (2011) 611–626

Contents lists available at ScienceDirect

Pervasive and Mobile Computing

journal homepage: www.elsevier.com/locate/pmc

A quality based routing protocol for wireless mesh networks

Amitangshu Pal *, Asis Nasipuri

Electrical & Computer Engineering, The University of North Carolina at Charlotte, 9201 University City Blvd., Charlotte, NC 28223-0001, United States

ARTICLE INFO

Article history:
Available online 1 December 2010

Keywords:
Wireless mesh networks
On-demand routing
Quality based routing

ABSTRACT

Wireless mesh networks can provide low-cost solutions for extending the reach of wireless access points by using multi-hop routing over a set of stationary wireless routers. The routing protocol for these networks may need to address quality considerations to meet the requirements of the user. In this paper, we present a quality based routing protocol for wireless mesh networks that tries to maximize the probability of successful transmissions while minimizing the end-to-end delay. The proposed routing protocol uses reactive route discoveries to collect key parameters from candidate routes to estimate the probability of success and delay of data packets transmitted over them. To achieve accurate route quality assessments, a new route quality metric is proposed that uses performance models of data packet transmissions as opposed to estimating route quality from the transmission of control packets, which have different transmission characteristics. These models are developed after careful evaluations of multi-hop wireless transmissions and validated by computer simulations. Relevant parameters that can be used to assess the route quality metric using these models are explained. Extensive performance evaluations of the proposed quality based routing protocol are presented and its benefits in comparison to some other known routing protocols are discussed.

1. Introduction

Wireless mesh networks (*WMNs*) have emerged as a promising approach for extending the reach of wireless access points, making them attractive for providing reliable wireless Internet access across wide areas. A *WMN* typically consists of a set of static mesh routers that use multi-hop transmissions to form a backbone network for facilitating communications between mesh clients. A client can be a WiFi card, a WiFi phone, or any device that has WiFi connectivity built into it. One or more of the mesh nodes may have wired connections to the Internet, and serve as gateways to the mesh clients. Because of their dynamic self-configuration and self-organization capabilities, *WMNs* typically have low installation and maintenance costs, and provide reliable service [1]. *WMNs* have been adopted by numerous academic and industrial deployments, such as *Champaign-Urbana CommunityWireless Network (CUWiN)* [2], *SMesh* [3], *SolarMESH* [4], *Wireless Mesh Network for Next Generation Internet (WING) project* [5] etc. In home applications, *WMNs* can be used to create a low-cost, easily deployable, high performance wireless coverage throughout the home, eliminating radio frequency (RF) dead-spots. They are also effective in small and large offices, manufacturing plants, university campuses, government buildings, and health care establishments, where Ethernet cabling does not exist or its installation is economically prohibitive.

In recent times, rich media and multimedia applications such as voice over IP (*VoIP*) and video on demand (*VOD*) are becoming increasingly popular in mobile wireless devices. Consequently, in addition to the ease of deployment, *WMNs* must provide support for multimedia applications, which require that the multi-hop communications meet *quality* requirements. This motivates the development of routing protocols for *WMNs* that try to improve the quality of communications, such

* Corresponding author. Tel.: +1 980 229 3383.
E-mail addresses: apal@uncc.edu (A. Pal), anasipur@uncc.edu (A. Nasipuri).

1574-1192/$ – see front matter © 2010 Elsevier B.V. All rights reserved.
doi:10.1016/j.pmcj.2010.11.009

as the end-to-end probability of success (*POS*) and delay in the network. Since interference is a key factor that affects data transmissions in multi-hop wireless networks, there is a need for investigating mechanisms by which routing decisions are based on interference considerations in addition to the path length, which is often the primary factor considered for routing in dynamic multi-hop wireless networks.

In this paper, we present a *quality based* routing protocol that tries to optimize the end-to-end *POS* and delay in all active routes in the network by using a novel quality based routing metric. Although a lot of work has been reported on quality based routing for multi-hop wireless networks, most existing approaches rely on the usage of control packets to estimate the route quality. But control (broadcast) packets differ from actual data packets as they are smaller in size and are sent at a lower transmission rate than data packets. Consequently, the data transmission performance using routing protocols that estimate route quality from control packets only may be poorer than expected. To avoid this problem, we propose a scheme that tries to obtain the *predicted route quality* by applying interference models that are obtained using offline measurements of actual *data packet transmissions*. The proposed quality based routing protocol uses control packets to determine relevant parameters of candidate routes, such as hop count and node IDs, which are utilized by the routing metric to provide accurate estimates of the route quality. It is assumed that all communication requests are directed towards the gateway, which serves as the centralized manager for all routing decisions based on global knowledge of node locations and activities. We presented some initial results on this work in [6], where the basic IEEE 802.11 MAC was considered with *RTS/CTS* and *ACK* packets disabled for simplicity. Here, we derive the interfence models and the corresponding route quality metric for the more general case including *RTS/CTS* and *ACK* packets. The major contributions of this paper are as follows. Firstly, we develop mathematical models for estimating key factors that influence the quality of communication over a multi-hop wireless network, such as channel access probability, *POS*, and delay. These factors are obtained from careful evaluations of the effect of interference in 802.11 networks with and without the *RTS/CTS* option and *ACK* packets. Secondly, we propose a route quality metric that is based on the established models for *POS* and delay and can be evaluated by the gateway node under the knowledge of node locations. The proposed quality metric relies on quality estimates of actual data packet transmissions rather than control packets. We propose a new *Interference and Delay Aware Routing (IDAR)* protocol based on our proposed quality metric that improves the ratio of the end-to-end *POS* over delay. Finally, we perform extensive evaluation of *IDAR* under a variety of conditions and metrics. We demonstrate the benefits of the proposed approach using simulation experiments.

The rest of the paper is organized as follows. In Section 2, we discuss related work on quality based routing in mesh networks. In Section 3, we present the assumed network model, the problem statement, and explain our proposed approach for quality based routing. In Section 4, we present the analysis and modeling of wireless transmissions in 802.11 networks that forms the basis for the development of a route quality metric. Section 5 describes our proposed quality based routing protocol (*IDAR*). In Section 6, we present performance evaluations of *IDAR* and its comparison with a popular shortest-path based routing protocol (*AODV*) and another quality based routing protocol *MARIA* [7]. Conclusions are presented in Section 7.

2. Related work

A number of routing metrics have been proposed for achieving quality based routing in *WMNs*. In [8] the authors propose a metric named *expected transmission count (ETX)* that uses the expected number of transmissions a node requires to successfully transmit a packet to a neighbor. The *minimum loss (ML)* metric proposed in [9] computes the delivery ratio with the objective of choosing the route with the lowest end-to-end loss probability. The *expected transmission time (ETT)* metric proposed in [8] is based on the time a data packet requires to be transmitted successfully to each neighbor. In [10], the *modified ETX (mETX)* metric is proposed that computes the bit error probability using the position of the corrupted bit in the probe and the dependence of these bit errors throughout successive transmissions. The *interference aware (iAWARE)* metric in [11] uses *signal to noise ratio (SNR)* and *signal to interference and noise ratio (SINR)* to continuously reproduce neighboring interference variations onto routing metrics.

Several routing protocols have been proposed that try to improve network quality by estimating parameters related to wireless interference. In [7], the authors propose an interference aware quality based routing protocol *MARIA*, where nodes involved in a route discovery estimate the residual bandwidth in its neighborhood and forward the information over the route request packet (*RREQ*). *MARIA* uses conflict graphs to characterize interference. The destination selects the route based on the highest minimum residual bandwidth, i.e. the least interference. An algorithm that chooses the route with the minimum commitment period of the bottleneck node is presented in [12]. Commitment period is defined as the sum of the time the node spends in transmission/reception and the time a node has to reserve to be idle for enabling the flow of interfering traffic. Thus reducing the commitment period results in reduced interference. In the *DARE* protocol [13], all nodes in a path reserve time slots for flows and all nodes near the reserved path abstain from transmissions during the reserved time slots, thus minimizing the possibility of interference. The length and periodicity of reservation are chosen according to the needs of the application. In [14], the authors propose an algorithm where each mesh router periodically measures the *RSSI*, average *SINR*, average number of transmission rounds, average residual block error rate and the actual spectral efficiency of the transport channel. Four classes of services are defined where each class has its minimum tolerated level of each *QoS* metric. The scheme chooses the route that meets the minimum tolerable levels of all these metrics.

Other approaches to quality based routing have also been explored. In [15], a *Genetic Algorithm (GA)* for *QoS* multicast routing has been defined. In this proposed scheme a multicast tree from each source to a set of receivers is constructed

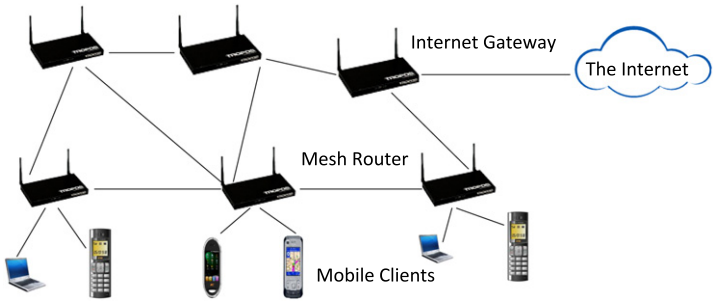

Fig. 1. Wireless mesh network architecture.

such that each edge in the tree has a minimum available bandwidth and the total delay of all the edges is less than the maximum delay that can be tolerated. A genetic algorithm is proposed to construct the multicast tree with minimum hop count. *QUORUM* [16] uses reactive route discovery and reservation based *QoS* provisioning. It estimates route robustness by counting the frequent *HELLO* packets that are received during a given time. It estimates the end-to-end delay by sending *DUMMY-RREP* packets, which have the same size, priority, and data rate, as the actual data packets. This helps in emulating real data traffic on a data path. The source selects the route for which the average delay of the *DUMMY* packets is within acceptable bounds and starts transmitting data traffic. *Wireless Mesh Routing (WMR)* [17] uses a novel bandwidth estimation algorithm where the required bandwidth and delay constraints are embedded in the route discovery message by the source. All the intermediate nodes forward the route discovery message if they can fulfill the bandwidth requirements and drop otherwise. Among all the routes satisfying the bandwidth requirement, the source chooses the shortest route that satisfies the maximum delay requirement. In [18], the authors propose an *Integrated QoS Routing (IQoSR)* procedure, where each intermediate node averages previous one-hop delay, link throughput and packet error rate measurements and piggybacks this information in the request packet. After getting the reply from destination, the source chooses the best route by calculating the integrated *QoS* performance metric.

Most of the above approaches use control packets for estimating the link quality, which can lead to inaccurate link quality estimation for data packet transmissions. For instance, a good link quality measurement method should be able to identify wireless link asymmetry that results from interference. If there is interference in the vicinity of node *A*, then the signals from node *B* to *A* might be disrupted, whereas signals from node *A* to *B* might be strong enough to overcome the interference. Many of the routing metrics in schemes that use control packets such as *ETX* (defined as $\frac{1}{p_f p_r}$, where p_f and p_r are the forward and reverse link packet reception ratios, respectively) give the same link quality in both directions, which is not true [19]. Our scheme is different from the perspective that we use control packets during route discovery to obtain relevant parameters from candidate routes, which are fed to a centralized route quality estimator for quality comparison and route selection. Since the estimator is based on performance models that are developed using offline measurements of *data packet* transmissions, the actual performance of data packet transmissions using this method would be closer to that predicted. We illustrated the principle using the basic IEEE 802.11 MAC without *RTS/CTS* and *ACK* packet options in [6]. In this paper we extend the quality metric proposed in [6] to support *RTS/CTS* and *ACK*. The contributions of this paper compared to [6] are as follows. First, we propose a new analytical model to measure the delay in the presence of *RTS/CTS* and *ACK*. Second, we develop a new interference aware *POS* model in the presence of control packets. Third, we propose a new route quality metric and finally validate the performance of our proposed quality model through simulations.

3. Problem formulation

3.1. Network model

We assume a network model that resembles a scenario where a *WMN* comprising of a set of static mesh routers is used to extend the reach of a wireless Internet gateway for a set of mobile users (see Fig. 1). The mesh routers dynamically form multi-hop routes between the mobile users (mesh clients) and the Internet gateway. Mesh routers have two interfaces, one for communicating with the mesh clients and another for communicating with other mesh routers. We focus on routing within the mesh routers only, i.e. the mesh clients do not participate in multi-hop routing. Only single channel operation is assumed, i.e. all mesh nodes operate on the same channel for transmitting mesh traffic. Although the use of multiple radios per node operating on multiple orthogonal channels would reduce interference effects, we focus on the quality of single channel networks in this work. It is assumed that the gateway is aware of the locations of the mesh routers, and keeps track of all active nodes and neighborhood information. In that sense, the assumed model for performing route selection in this paper is centralized, since we assume that the gateway node uses global information on node locations, their neighborhoods, and channel usages for routing. Such information can be obtained in a static network by employing appropriate channel probing techniques, which is not addressed in this paper.

Table 1
Simulation environment.

Parameter	Values used
Max node queue length	200
Propagation model	Two ray ground
Receiver antenna gain	0 dB
Noise floor	−101 dB m
SINRPreamblecapture	4 dB
Modulation scheme	BPSK
Data packets size	1000 bytes
Transmitter antenna gain	0 dB
Transmit power	20 dB m
SINRDatacapture	10 dB
PowerMonitor threshold	−86.77 dB m
Traffic generation	Exponential

3.2. Problem statement

With these assumptions, we consider that at any point of time a mesh router S seeks a multi-hop route to the gateway node. We consider an on-demand framework for routing, where S broadcasts a route request packet when it requires a route to the gateway. The route request packet reaches the gateway node via various routes, and carries relevant parameters of each of the paths traced. The gateway node considers these inputs and existing traffic conditions from all active nodes in the network to determine the best route for S. Note that all communications are directed to the same gateway node, and hence it is practically possible to implement a centralized routing solution under the assumed network model. Although technically any ad hoc routing protocol can be applied here, such routing protocols usually try to minimize path lengths, which does not necessarily give the best quality. The problem here is to determine the route that provides the best *communication quality*, in terms of the end-to-end *POS* and delay. Hence, the main problem addressed in this work is to determine a suitable routing metric that accurately captures the quality of communication over a candidate route.

3.3. Proposed approach for quality based routing

Our approach for solving the above problem is to develop accurate models for the *POS* and delay in multi-hop wireless networks using a simple set of measurable parameters, and incorporate these models into a route quality metric. Our goal is to implement an on-demand routing scheme that evaluates various routes based on the quality metric and selects the best. The proposed routing scheme is structured in a manner similar to *AODV* except that the control packets collect essential information to evaluate the quality metric, which forms the basis for route selection.

In order to develop a route quality metric, we start with extensive performance evaluations of a wireless link in a multi-hop network to determine important parameters that affect the characteristics of a link. We assume IEEE 802.11 as the underlying MAC protocol, and develop its performance models with and without the *RTS*/*CTS* option. Our objective here is to develop a performance model that is suitable for incorporating into a route quality metric, which can be evaluated for routing decisions by the gateway node. In particular, we show that when the 802.11 MAC is used without the *RTS*/*CTS* and *ACK* options, the primary factors influencing the throughput and delay in a test link at a given offered load can be effectively captured by two measurable quantities: (a) the number of active neighbors of the transmitter, and (b) the number of interferers of the receiver, along with a number of other parameters such as locations and interplay of neighboring nodes. These parameters can be easily obtained by the gateway node to evaluate the communication qualities of candidate routes. The evaluation is somewhat more complex when the *RTS*/*CTS* and *ACK* packets are enabled, since it involves more parameters. However, we obtain appropriate models for capturing the link performance using a measurable set of parameters for this case as well, which are described in Section 4.

4. Development of the route quality metric

We now present the characterization of wireless transmissions in multi-hop networks leading to the development of the proposed route quality metric. These are obtained from simulation experiments using the *network simulator*-2 (*ns*2) [20]. For the sake of explanations and performance evaluations, we consider a network where the nodes are placed on a uniform grid, as shown in Fig. 3. However, our analysis applies to any deployment scenario. The parameters used in the simulations are listed in Table 1. As stated before, we assume the IEEE 802.11 *CSMA*/*CA* MAC with and without the *RTS*/*CTS* option. We focus on the key link-level performance issues, which include (a) the channel access ratio (CAR), i.e. the ratio of the offered load that is actually transmitted, (b) the link-level probability of success, i.e. the probability that a transmitted packet is successfully received by the receiver, and (c) the average transmission delay of a packet in the MAC layer.

4.1. Channel access ratio

A transmitting node has to contend with its active neighbors to gain access to the channel. Consequently, the *CAR* for a transmitter depends on the number of active neighbors and their level of activity, which is dependent on the traffic load.

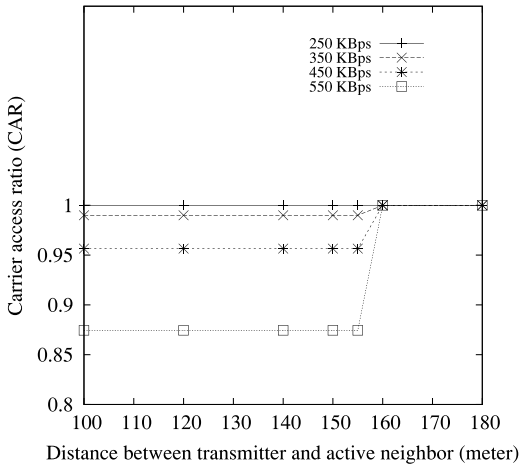

Fig. 2. Variation of CAR with respect to distance from an active neighbor.

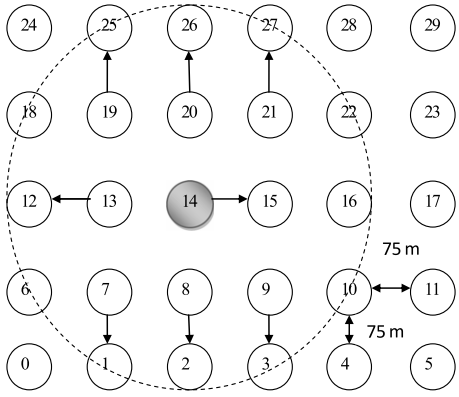

Fig. 3. Simulation environment to evaluate the effect of active neighbors on the test link 14 → 15. The dotted line shows the carrier sensing range.

It does not depend on the distance from the active neighbors as long as they are within the carrier sensing range. These observations are validated in Fig. 2, which depicts the variation of the *CAR* in a test link with respect to the distance from an active neighbor for different loads of the active neighbor. The figure also shows that with the chosen parameters, the carrier sensing range (*CSR*) is 155 m.[1]

When multiple active neighbors are involved, the *CAR* depends on a complex interaction of carrier sensing, back-offs, and transmission activities from all contending nodes, whose numbers vary from one node to another. Consequently, we attempt to determine the effect of the number of active neighbors on *CAR* from simulation experiments. We consider the test link 14 → 15 in a network of 30 nodes that are placed in a uniform grid as shown in Fig. 3 and determine the variation of *CAR* in the test link with increasing number of active neighbors (i.e. by incrementally activating transmissions from nodes 7, 8, 9, 13, 19, 20 and 21). The results, depicted in Fig. 4(a) for the case where *RTS/CTS* packets are disabled, indicate that the *CAR* is not noticably affected by the active neighbors for loads lower than 150 KBps, but it drops significantly and non-linearly at higher loads, especially for higher numbers of active neighbors.

Fig. 4(b) shows the same results when the *RTS/CTS* is enabled. The *CAR* in the presence of *RTS/CTS* is lower than without *RTS/CTS*, which is reasonable, since data packets are transmitted (i.e. the channel accessed) only when the channel is clear at both the sender and receiver node locations.

4.2. Transmission delay

The transmission delay in 802.11 channels depends on a number of components, of which the queuing and access delays are significant. The queuing delay Q_d is the property of the transmitting router, which is the time that a packet has to wait in its transmission queue before it actually reaches the head of the queue and starts contending for the channel. Q_d is directly

[1] The fact that the transmission range and *CSR* turn out to be be equal here is coincidental. Our analysis is quite general and is applicable even when they are different.

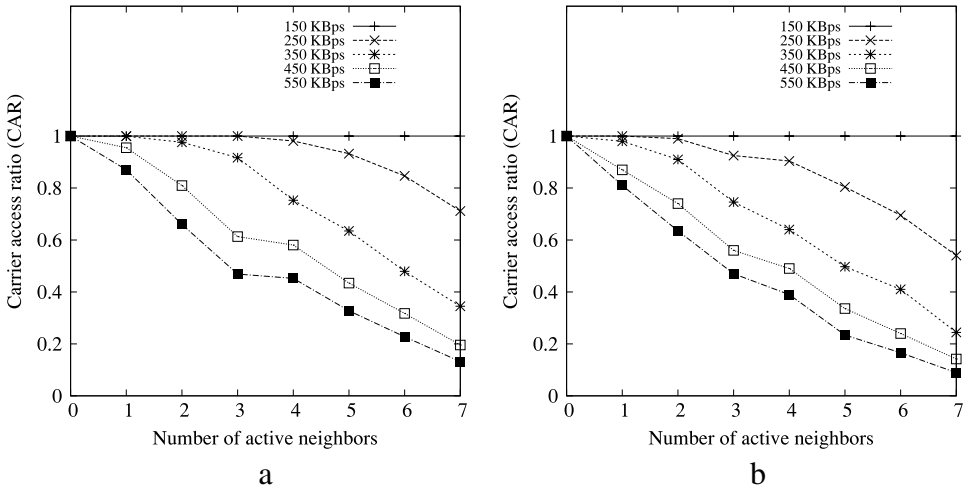

Fig. 4. Variation of CAR (a) without *RTS/CTS*, (b) with *RTS/CTS*.

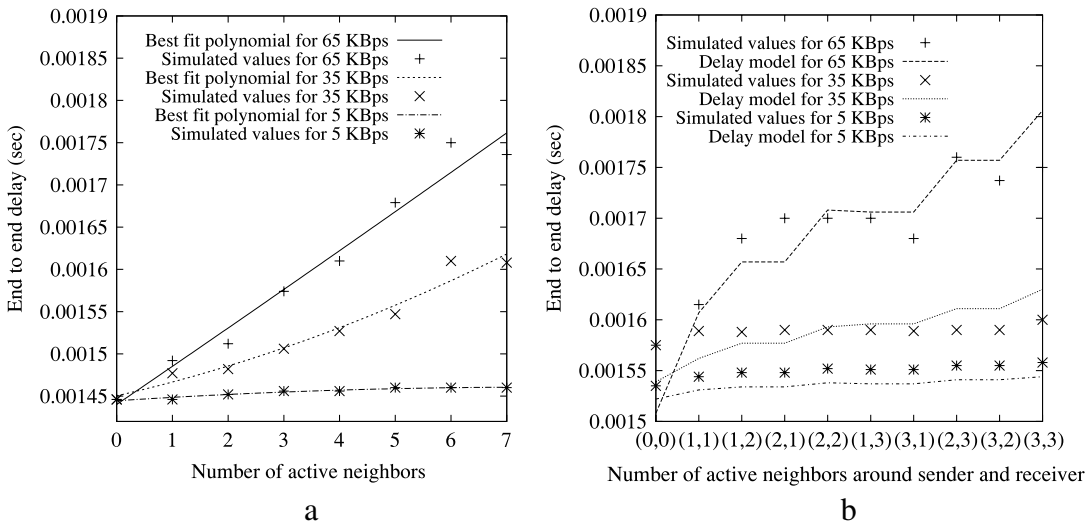

Fig. 5. Variation of delay (a) with number of active neighbors of sender without *RTS/CTS*, (b) with number of active neighbors of sender and receiver with *RTS/CTS*.

related to the length of the queue and the arrival rate of the packets entering in the queue. On the other hand, the access delay Q_a is the time that a packet at the head of the transmission queue has to wait before the contention in the channel is resolved by *CSMA/CA* and the packet gets access to the channel and starts transmission. The sum of the average queuing and access delays, referred to as total delay t_d, is an important factor affecting the quality of a communication link. Again, we consider the test link 14 $\rightarrow$ 15 and obtain the variation of the total delay in the test link with different numbers of active neighbors. When the *RTS/CTS* packets are disabled, the delay of the test link 14 $\rightarrow$ 15 is found to fit a quadratic polynomial:

$$T_d(n_a) = An_a^2 + Bn_a + C \tag{1}$$

where n_a is the number of active neighbors of the sender and A, B and C are the best fit coefficients that depend on the offered load. Simulations were run at several different offered loads, and the best-fit coefficients were found to be $A = -3.57 \times 10^{-7}$, $B = 4.814 \times 10^{-6}$, and $C = 0.001443$ for 5 KBps; $A = 1.88 \times 10^{-6}$, $B = 9.54 \times 10^{-6}$, and $C = 0.00146$ for 35 KBps; and $A = -7.023 \times 10^{-7}$, $B = 5.25 \times 10^{-5}$ and $C = 0.001425$ for 65 KBps. Delays obtained from simulations and the best fit curves described above are shown in Fig. 5(a), which validates the quadratic approximation.

If we enable *RTS/CTS* packets, then the time for a packet to reach the destination depends on the active neighbors of the sender (n_a) as well as of the receiver (which we denote by n_b) since a data packet is not transmitted unless the receiver has access to the channel to send the *CTS*. The total delay can be expressed as:

$$T_d^{RTS/CTS}(n_a, n_b) = T_{CA\_sender} + RTS_{period} + SIFS + T_{CA\_receiver} + CTS_{period} + SIFS + T_{Data\_Tx} \tag{2}$$

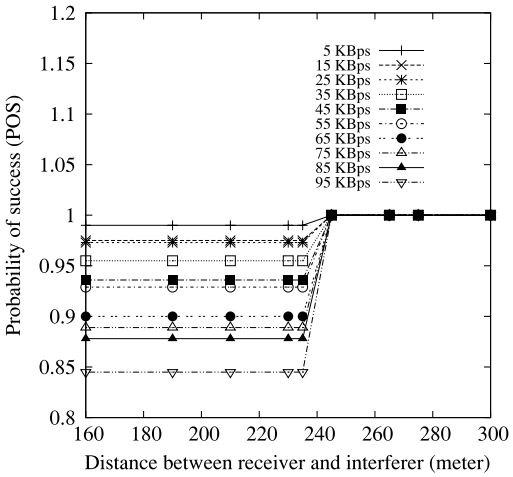

Fig. 6. Variation of POS with respect to distance from an interferer.

Table 2
Probability of overlapping transmission with the test packet.

Data rate (KBps)	Number of interferers				
	1	2	3	4	5
5	0.0079	6.29×10^{-5}	5×10^{-7}	3.96×10^{-9}	3.15×10^{-11}
35	0.048	0.0023	0.00011	5.36×10^{-6}	2.5×10^{-7}
65	0.09	0.00828	0.00075	6.85×10^{-5}	6.24×10^{-6}

where RTS_{period} and CTS_{period} are the transmission times of *RTS* and *CTS* packets, respectively, which can be calculated from their sizes (assumed to be 20 bytes and 14 bytes, respectively), and *SIFS* is the short interframe spacing length, which is taken as 16 μs. T_{CA_sender} is the time for the sender to get access to the channel, which, from the previous section, is expressed as $T_{CA\_sender} = An_a^2 + Bn_a$. T_{Data_Tx} is the data transmission delay, which is equal to C. Similarly, the channel access delay at the receiver $T_{CA\_receiver} = An_b^2 + Bn_b$. Then the expression for the total delay $T_d^{RTS/CTS}(n_a, n_b)$ in the presense of *RTS* and *CTS* can be written as:

$$T_d^{RTS/CTS}(n_a, n_b) = A(n_a^2 + n_b^2) + B(n_a + n_b) + 1.0324C + 0.000032. \tag{3}$$

We validate this model using simulations by evaluating the total delay with varying number of active neighbors. These results are shown in Fig. 5(b), where the sample point (i, j) on the x-axis implies that $n_a = i$ and $n_b = j$.

4.3. Probability of success

We now evaluate the probability of successful reception of a transmitted packet on the test link. Since the *POS* (defined as the fraction of the transmitted data packets that are received successfully) is very different for the cases when *RTS*, *CTS*, and *ACK* packets are disabled and when they are enabled, we consider these two cases separately.

4.3.1. POS with RTS/CTS disabled

With *RTS/CTS* packets disabled, the *POS* is only dependent on the probability of successful reception of the data packet at the receiver. A data packet is received correctly if its signal to interference-plus-noise ratio (SINR) does not fall below the minimum SINR threshold at the receiver at any time during the reception of the packet. When the receiver has only one interfering node, a packet transmission can be unsuccessful if the distance of the interferer is smaller than a limit, which is often termed as the *interfering range*. In Fig. 6, we depict the variation of the *POS* in the test link 14 → 15 with *RTS/CTS* turned off, with respect to the distance from an interfering node from the receiver. The figure shows that for the chosen parameters, the interfering range is 235 m. When an interferer is within this range, the *POS* depends on the load, which determines the probability that a transmission from the interferer overlaps with the test packet. Note that if the transmissions from multiple interferers overlap, the aggregate interference will increase, thereby causing the interfering range to increase. However, that probability is usually low unless the offered load is very high. For instance, the probabilities that transmissions from mutliple interferers overlap with a test packet at different transmission loads are shown in Table 2. Based on these data, we ignore the possibility of overlapping transmissions in our calculations. For the rest of the paper, we assume the grid spacing to be 150 m, with which a receiver can have up to 5 interferers.

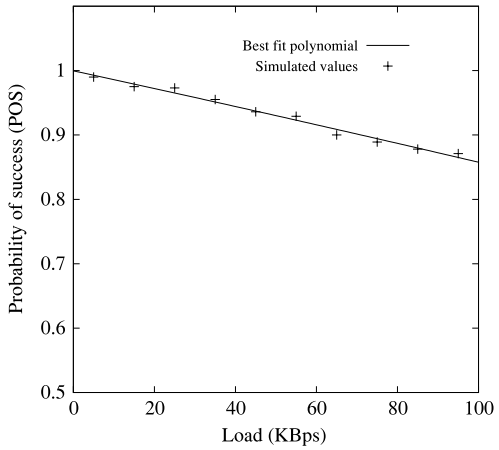

Fig. 7. Experimentally obtained POS versus load in the presence of one interferer.

Generally, a link in a wireless network comes under the influence of a number of interferers whose transmissions may be dependent or independent of one another. *Independent interferers* are those whose transmissions are not in any way affected by one another, i.e. each node's transmissions occur independently of those from the others. So the combined interference from a set of independent interferers can be calculated easily. On the other hand, if the transmissions of any interfering node are in some way dependent on transmissions from other nodes that are also located within the interfering range of the test node, then the combined interference is more difficult to model. We term such nodes *dependent interferers*, which are addressed later.

If S is the transmitter and D is the receiver in a test link, then the *POS* of the link $S \rightarrow D$ in the presence of a set of N independent interferers I with transmitted load L (L is given by $CAR \times$ offered load) can be written as:

$$P_S(I) = \prod_{k=1}^{N} P_S(i_k) \tag{4}$$

where $I = \{i_1, i_2, \ldots, i_N\}$ is the set of N interferers of D and $P_S(i_k)$ is the probability of success of the test link when i_k is transmitting.[2] Our simulation experiments demonstrate that the probability of success in the presence of a single interferer can be modeled by a quadratic function of L which is very consistent over a wide range of loads. Thus, we write

$$P_S(i_k) = Q.L^2 + R.L + T \tag{5}$$

where $Q = -1.49184 \times 10^{-6}$, $R = -0.00128499$ and $T = 0.998588$ are the coefficients that best fit the results obtained from simulations. The simulation results and the best fit curve are shown in Fig. 7. As expected, the *POS* decreases consistantly with increasing load, which is due to an increasing amount of interference from transmitting nodes.

It must be noted that although the set I can be estimated by the set of nodes that are located within the interfering range of the receiver, some additional factors affect the accuracy of Eq. (4). For instance, wireless propagation can be highly non-isotropic because of shadowing, multipath reflections, and other long term fading effects. This can make it difficult to estimate the actual interference from a source from its distance from the receiver. However, because of the threshold effect of the interference from any source, we find that using the interfering range to identify interferers is generally acceptable. This issue needs additional considerations if *RTS*/*CTS* and *ACK* packets are assumed, which is discussed later.

To validate the *POS* model in Eq. (5), we compare results obtained from Eq. (5) with those obtained from simulations in Fig. 8, where all the nodes have the same load and, hence, the same $P_S(i_k)$ for all i_k. The actual *POS* values obtained from simulations closely match the values obtained from the model. This confirms our claim that the *POS* of a test link for a given load can be approximately estimated from the number of active neighbors of the transmitter and the number of interferers of the receiver using the models developed above.

We now address the issue of independence of transmissions from interferers in the set I. We show with an example that not all nodes located within the interference range of a test receiver can transmit independently. For instance, consider

[2] Note that $P_S(i_k) = 1 - P_t(i_k)$, where $P_t(i_k)$ is the probability that a transmission from interferer i_k overlaps with the test packet from S and depends on the transmitted load L.

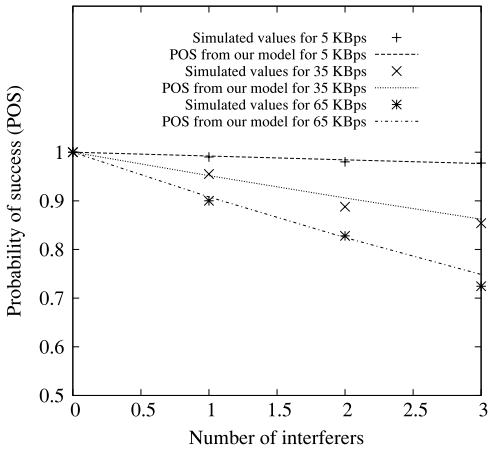

Fig. 8. POS versus number of interferers of D: model and simulation results.

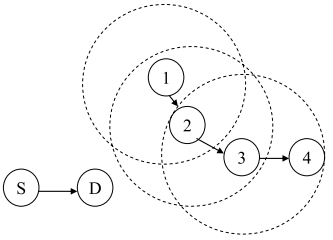

Fig. 9. An example for getting the number of independent interferers for test link $S \rightarrow D$: the dotted circles show the carrier sensing range.

Fig. 9, where the test link $S \rightarrow D$ has three active interferers, nodes 1, 2, and 3. Now, with the *RTS/CTS* option disabled, the maximum number of *independent* interferers is two, as only nodes 1 and 3 can transmit data packets at the same time. Hence, for this case, although there are three active nodes within the interfering range of D, we assume that the *effective number of interferers* is two, since one of them is dependent on the others. The case is slightly different when the *RTS/CTS* option is used, since a successful *RTS/CTS* exchange between either nodes 1 and 2 or nodes 3 and 4 will silence the other nodes through the duration of that transmission. So, the number of independent (or effective) interferers for this case is one. From the above discussion, it is clear that for analyzing the *POS* we only need to consider the number of effective interferers. From this section onwards, the term interferer means effective number of interferers.

4.3.2. POS with RTS/CTS and ACK enabled

When the *RTS/CTS* option is enabled, a data packet is only transmitted when the *RTS/CTS* exchange is successful, i.e. the channel is found to be clear both at the transmitting and receiving nodes. However, the transmitted data packet can still be lost due to interference caused to the data packet or the *ACK* packet. Here, we analyze the possible events that can cause these transmission failures, which are explained with the help of Fig. 10. In the figure, $S \rightarrow D$ represents the test link, Δ_{RTS} and Δ_{CTS} denote the regions where the *RTS* and *CTS* packets for the test link can be received, and Δ_{CS} denotes the area around S where nodes can sense the transmission from S. For any node i, $I(i)$ denotes the area from where a transmission from any node $j \in I(i)$ can interfere with a packet being received at i. Our approach is to explore various cases where events can lead to the loss of the *DATA* or the ACK packet, both of which can cause the data transmission from $S \rightarrow D$ to be unsuccessful. For each of these cases, we evaluate the factors that affect the *POS*, as outlined below.

Case-1. The transmitted data packet is unsuccessful due to interference from nodes that are within the interference range of the receiver but outside its transmission range, i.e. range of reception of the *CTS* packet. These nodes are marked as PC_i in Fig. 10, of which we assume p nodes are sending and r nodes are receiving. Since both events can generate interfering packets, we consider the probability of success of the test data packet in the presence of both these events. Note that the sending nodes can interfere by transmissions of either *RTS* or data packets. However, since the length of the *RTS* packet is much smaller than that of the data packets, its effect on the *POS* of the data packet at D will be much smaller, and so we only consider the interference of data packet transmissions from the p sending nodes among PC_i. In the absence of any other interferer that can affect the reception of the test data packet, the effect of a single interfering node among PC_i can be evaluated as follows. We assume that the length of data packet is *DLEN* (in bits) and all the nodes generate packets based

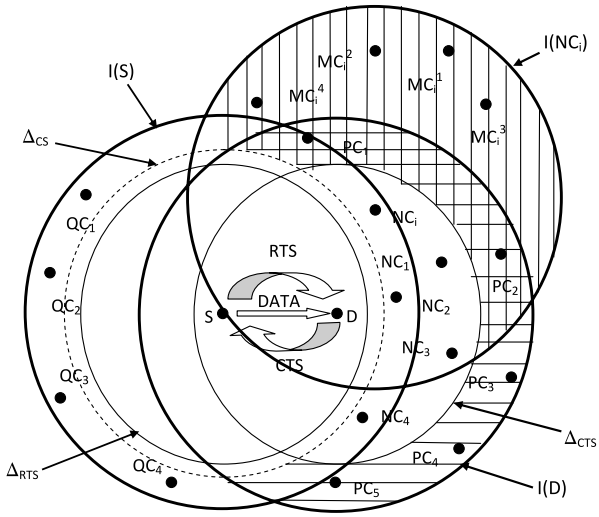

Fig. 10. Effect of interferers in the presence of *RTS*/*CTS* for test link $S \rightarrow D$.

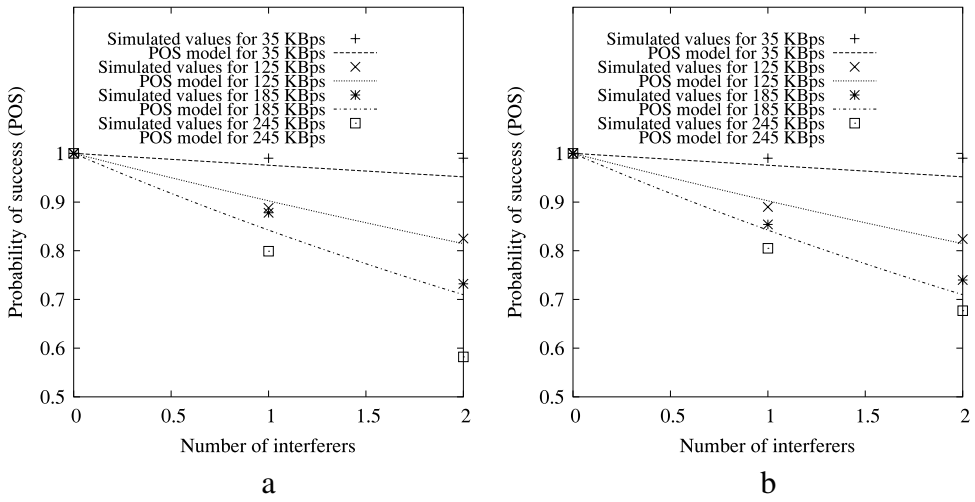

Fig. 11. *POS* with (a) number of sending nodes among $PC_i(p)$, (b) number of receiving nodes among $PC_i(r)$.

on a Poisson process. B is the bandwidth of the channels in bits/seconds or bytes/seconds and λ is the arrival rate of the data packets.

$$
\begin{aligned}
&P(DATA \text{ is received successfully } \mid DATA \text{ is transmitted})\\
&= P(DATA \text{ is received successfully } \mid RTS \text{ is received successfully at } D)\\
&= \frac{P(DATA \text{ and } RTS \text{ are received successfully})}{P(RTS \text{ is received successfully at } D)}\\
&= \frac{P\left(PC_i \text{ does not send } DATA \text{ in vulnerable period } \left(\frac{2\times DLEN}{B}\right) \text{ of } S\right)}{P\left(PC_i \text{ does not send } DATA \text{ in vulnerable period } \left(\frac{DLEN}{B}\right) \text{ of } S\right)}\\
&= \frac{e^{\frac{-2\times\lambda\times DLEN}{B}}}{e^{\frac{-\lambda\times DLEN}{B}}} = e^{\frac{-\lambda\times DLEN}{B}}. \qquad (6)
\end{aligned}
$$

For p independent senders $(PC_1, PC_2, \ldots, PC_p)$ in this region, the probability of success of the *DATA* packet is given by $e^{\frac{-\lambda\times DLEN\times p}{B}}$.

The receiving nodes among PC_i can interfere by the transmission of *CTS* packets during the transmission of the test *DATA* packet; thus the probability of success in the presence of r such nodes is $e^{\frac{-\lambda\times DLEN\times r}{B}}$.

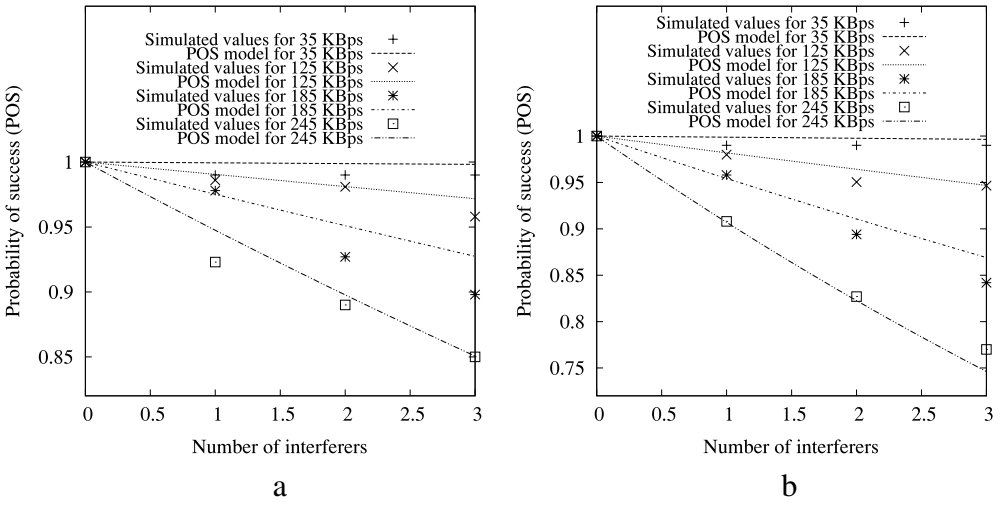

Fig. 12. POS with number of active nodes among NC_i (n) (a) with $m = 1$, (b) with $m = 2$.

To validate these models, we perform simulations to study the effect of p senders among PC_i and r receivers among PC_i independently. These are depicted in Fig. 11(a) and (b), respectively, which show that our models are reasonably accurate.

Case-2. The test data packet is unsuccessful due to interference from nodes that are within the transmission range of D but fail to receive the *CTS* packet. A node located in the Δ_{CTS} that is outside the Δ_{CS} of S (marked as NC_i in Fig. 10) may not receive the *CTS* from D correctly due to an overlapping transmission from MC_j^i (refer to Fig. 10). The probability of this event is $1 - e^{\frac{-\lambda \times DLEN}{B}}$, which is the probability that MC_i^j transmits in the vulnerable period ($\frac{DLEN}{B}$) of the *CTS* transmission from D.[3] In general, if there are m such interferers among MC_i^j, then the probability that the *CTS* is not received by NC_i is $1 - \prod_{i=1}^{m} e^{\frac{-\lambda \times DLEN}{B}} = 1 - e^{\frac{-\lambda \times DLEN \times m}{B}}$.

The probability that NC_i, having failed to receive the *CTS* from D, interferes with the reception of the *DATA* packet at D is then given by $(1 - e^{\frac{-\lambda \times DLEN}{B}})$, which is the probability that an *RTS* transmission from NC_i overlaps with the test *DATA* packet at D. Consequently, the probability that the *DATA* transmission from S to D is successful in the presence of unsuccessful reception of the *CTS* packet at NC_i is given by $1-(1-e^{\frac{-\lambda \times DLEN}{B}})\,(1-e^{\frac{-\lambda \times DLEN \times m}{B}})$. If there are n such nodes $(NC_1, NC_2, \ldots, NC_n)$, then the *DATA* transmission will be successful with a probability of $\prod_{i=1}^{n} 1 - (1 - e^{\frac{-\lambda \times DLEN}{B}})\,(1 - e^{\frac{-\lambda \times DLEN \times m}{B}})$.

To validate the above *POS* model, we perform simulations by first varying n keeping $m = 1$, i.e. assuming that each of NC_i, $i = 1, 2, \ldots, n$, has one interferer MC_i^j only. These results are compared with the proposed *POS* model in Fig. 12(a). Fig. 12(b) depicts the results where the number of interferers for each NC_i is doubled, i.e. $m = 2$. These results are in close agreement with our *POS* model.

Case-3. The transmitted *ACK* packet is unsuccessful due to interference from nodes that are within the interfering range of S. This interference can be caused by a transmitted *RTS* or *DATA* packet. A node in this region may send an *RTS* packet during the transmission of the *ACK* packet on the test link, if it has missed the *RTS* packet from S. But as the packet sizes of both *ACK* and *RTS* are very small in comparison to a *DATA* packet, the vulnerable period is also small. So, the probability of collision of an *RTS* and *ACK* is negligible and we ignore this possibility.

We next consider the possibility of interference of a data transmission from a node located within the interference range of S on the *ACK* being received at S. In particular, we are interested in the interference from the nodes marked as QC_i in Fig. 10. Now, the transmission of an *ACK* packet from D to S implies that the corresponding *RTS*/*CTS* exchange was successful, which implies that the nodes QC_i, $i = 1, 2, \ldots,$ did not transmit during the vulnerable period of the *CTS* transmission. These nodes would be unsuccessful in exchanging *RTS*/*CTS* packets during the following period of *DATA* transmission from S. Consequently, after successful completion of the *DATA* transmission from S, the only packets that can be transmitted from a node QC_i that can interfere with the reception of the *ACK* packet at S are *RTS* or *CTS* and not *DATA*, which results in a small probability of overlap. Hence, we can ignore the effect of this case as well.

By taking into account all the factors described above, the probability of success of a transmitted data packet using the *RTS*/*CTS* handshake is given by

$$POS = \left\{ \prod_{i=1}^{n} 1 - (1 - e^{\frac{-\lambda \times DLEN}{B}})(1 - e^{\frac{-\lambda \times DLEN \times m}{B}}) \right\} \times e^{\frac{-\lambda \times DLEN \times p}{B}} \times e^{\frac{-\lambda \times DLEN \times r}{B}}. \quad (7)$$

[3] As before, we ignore the effect of interference of the smaller *RTS* and *CTS* packets in favor of a data packet from MC_i^j, since those probabilities are comparatively smaller.

In order to evaluate the *POS* using the above expression, we note that for any given load, the *POS* in the presence of a single interferer with the *RTS/CTS* disabled is given as:

$$P_S(i_k) = P\left(i_k \text{ does not send DATA in vulnerable period } \left(\frac{2 \times DLEN}{B}\right) \text{ of } S\right) = e^{\frac{-2\times\lambda\times DLEN}{B}}. \tag{8}$$

Consequently, we have $e^{\frac{-\lambda\times DLEN}{B}} = \sqrt{P_S(i_k)}$, where $P_S(i_k)$ can be evaluated using Eq. (5). Hence, Eq. (7) may be evaluated from

$$POS = \left\{\prod_{i=1}^{n} 1 - (1 - P_S^{1/2}(i_k))(1 - P_S^{m/2}(i_k))\right\} \times P_S^{p/2}(i_k) \times P_S^{r/2}(i_k). \tag{9}$$

It must be noted that some additional factors affect the *POS* and hence, the accuracy of Eq. (7). Firstly, Eq. (7) is based on the assumption that all interferers in case-1 transmit independently of interferers in case-2 and case-3. But this is not exactly true as there are dependencies among these interferers. The number of interferers whose transmissions are independent of each other is often hard to obtain. Secondly, in Eq. (7) we assume that the arrival rates of all the interferers are the same. However, the rate of transmissions of *DATA* packets depends on successful reception of *CTS* at the sender, which depends on the nodes that interfere with it. Again these interferers depend on other interferers as well. Thus, obtaining an accurate estimate of the *POS* becomes intractable. Nevertheless, our model gives a good estimate of the *POS* for a test link considering the various measurable parameters in a static multi-hop wireless network, such as the number of interferers m, n, p, and r in different scenarios.

4.4. Route quality metric

We now apply the above models of the estimated *POS* and delay of a test link to define an end-to-end route quality metric. We consider that the end-to-end *POS* of a multi-hop route is given as the product of the *POS* of every individual link on the route and the end-to-end delay is given as the sum of the individual link delays. Consequently, based on the objective of maximizing the end-to-end *POS* and minimizing the end-to-end delay, we define the route quality $Q(r)$ metric for route r of length v operating at load L as follows:

$$Q(r) = \left(\prod_{f=1}^{v} P_S(I_f)\right) \Big/ \sum_{f=1}^{v} T_d(n_{af}) \quad \text{without } RTS/CTS \tag{10}$$

$$= \left(\prod_{f=1}^{v} P_S(I_f)\right) \Big/ \sum_{f=1}^{v} T_d(n_{af}, n_{bf}) \quad \text{with } RTS/CTS. \tag{11}$$

Here, f is a link on the route from source to destination, $P_S(I_f)$ is the *POS* of link f, I_f is the set of interferers, and $T_d(n_{af})$ is the delay experienced by a packet with n_{af} active neighbors at the sender. Similarly, $T_d(n_{af}, n_{bf})$ is the delay with n_{af} and n_{bf} active neighbors at the sender and the receiver end respectively.

5. Interference and delay aware routing

In this section we describe the proposed quality based routing protocol *IDAR* that uses the quality metric derived in the previous section. *IDAR* is a reactive routing protocol that tries to select routes with the highest ratio of the end-to-end *POS* and delay based on parameters collected and conveyed by *RREQ* packets. We present two versions of *IDAR*, which differ in the contents of the propagating *RREQ* packets and how the quality metric is calculated. These are described in detail below:

IDAR_v1: Protocol functionality of our proposed routing protocol *IDAR_v1* can be divided into the following different phases.

- Route discovery. When the source does not have a route to the destination, it broadcasts a route request packet (*RREQ*) to its neighbors. In addition to many other fields, the *RREQ* contains a *RREQ ID*, the destination address, the source address, the number of active neighbors of the sender (A), the accumulated *POS* on the current route (P_S), the accumulated delay in the current route (T_d), and a timestamp. These quantities A, P_S, T_d and the timestamp are initialized at the source to the number of active neighbors of the source, $P_S = 1$, $T_d = 0$, and timestamp = the time when the *RREQ* packet was generated. Every intermediate node updates the accumulated *POS* and delay based on the number of active neighbors of the previous node and its active interferers before forwarding it. The *RREQ ID* combined with the source address uniquely identifies a route request. This is required to ensure that the intermediate nodes rebroadcast a route request only once in order to avoid broadcast storms. If any intermediate node receives a *RREQ* more than once, it just discards it. All intermediate nodes do the same thing until the *RREQ* reaches the destination. The timestamp is used to reduce unnecessary flooding of *RREQ* packets throughout the network.

Table 3
Best routes on both calculated and simulated scenarios.

Order of *RREQ* packets for $0 \rightarrow 29$	Calculated quality for $0 \rightarrow 29$	Simulated quality for $0 \rightarrow 29$
1	4	4
2	3	1
3	5	3
4	2	2
5	1	5

- Route selection. For every *RREQ* packet, the destination calculates the quality metric $Q = P_S/T_d$. The destination waits for the first ten packets and forwards a route reply packet (*RREP*) back on the route that has the highest Q value. All the intermediate nodes forward the *RREP* back to the source and update their routing table entry. The source then starts sending the data packets via this route.
- Route maintenance. If a routing table entry is not used for a long time, that entry is erased. This is required as the network scenario changes with time, thus after a long time if a source needs a route to the gateway, it has to start a route discovery to get a good quality route.

In *IDAR_v*1, the intermediate nodes are required to calculate the *POS* and delay, for which the nodes must know their active neighbors and interferers. One way to achieve this is for the gateway to forward this information to all nodes at periodic intervals, which causes additional overhead.

IDAR_v2: In order to avoid the overhead problem mentioned above, we propose another version of the *IDAR* routing protocol, named *IDAR_v2* and the different phases are described as follows.

- Route discovery. Here, instead of carrying A, P_S and T_d as in *IDAR_v*1, the *RREQ* simply carries the sequence of nodes that it has traversed. The rest of the route discovery procedure is similar to *IDAR_v*1.
- Route selection. The destination (gateway) uses the node location and neighborhood information to calculate the end-to-end *POS* and delay, and hence the Q for each route. In addition to solving the problem of providing all nodes with node location information, *IDAR_v2* also calculates the route quality more accurately because it can use global location information to determine dependent and independent interferers based on the information conveyed by each *RREQ* packet.
- Route maintenance. Route maintenance is the same as for *IDAR_v*1.

But the disadvantage of this scheme is that as the intermediate routers have to append their own IDs, the size of the *RREQ* packet gets larger as it propagates along the network, which can be a problem for large networks.

6. Performance evaluation of *IDAR*

We perform extensive performance evaluations to determine the effectiveness of the proposed route quality metric and the *IDAR* routing schemes. The quality of the routes are determined by obtaining the average *UDP* end-to-end packet delivery ratio and delay using *ns*-2 and comparing them with a traditional shortest-path reactive routing protocol (*AODV*) and the quality based routing scheme *MARIA* [7].

6.1. Validating the new quality metric for a route

We first validate the benefits of using our new quality metric $Q(r)$ with the help of a specific simulation experiment. For this experiment, we disabled the *RTS/CTS* option and *ACK* packets in the *MAC*. We consider the same grid network of 30 nodes (Fig. 3) and apply our quality metric to multiple candidate routes from $0 \rightarrow 29$ when the flow $5 \rightarrow 24$ is active and *fixed* as depicted in Fig. 13. The chosen candidate routes are those that are traced by the 1st, 2nd, 3rd, 4th and 5th *RREQ* packets that reach node 29 from a route discovery initiated at node 0, as obtained from an *ns*-2 simulation using the *AODV* routing protocol. We assume that both flows are running *UDP* with a transmission rate of 35 KBps. The relative orders of these routes in terms of the calculated values of the proposed route quality metric as well as their qualities obtained from simulations of *UDP* data flows on these routes are shown in Table 3, with the highest values at the top. The simulated qualities are obtained by taking the ratio of the achieved end-to-end *POS* over the delay on each route. Column 2 shows the relative order of the values of $Q(r)$ of $0 \rightarrow 29$ *as calculated* and column 3 shows the order of the achieved quality along the corresponding routes as obtained from simulations. From Table 3 we can see that the order of best routes as calculated from the proposed quality metric is very different from the order in which they arrive at the destination. Consequently, while the *AODV* routing protocol would use the first route, the route that would be selected using the proposed *IDAR* protocol in this scenario is route 4, i.e. the route taken by the fourth *RREQ* packet. Also, the relative order of the qualities of these routes as estimated by the proposed $Q(r)$ metric is very similar to the order of quality performance figures actually obtained from data transmission simulations. This indicates that the proposed quality metric $Q(r)$ is effective in determining the route that has the highest end-to-end *POS* over delay ratio.

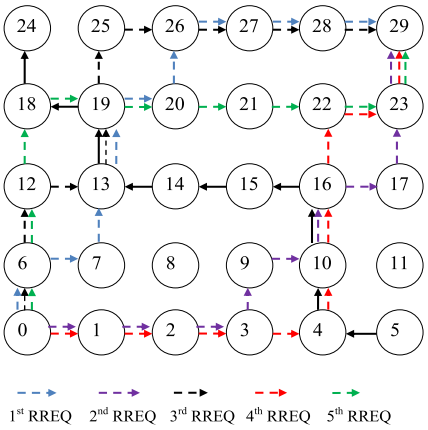

Fig. 13. Routes taken by the 1st, 2nd, 3rd, 4th and 5th *RREQ* to reach their destination for the pair 0 → 29 when 5 → 24 (shown as solid black arrows) is fixed.

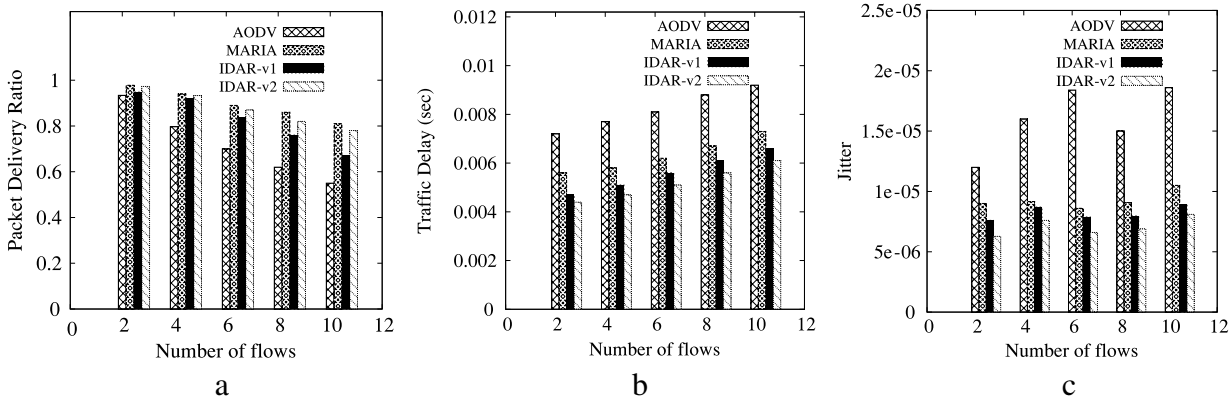

Fig. 14. (a) Comparison of packet delivery ratio. (b) Comparison of delay. (c) Comparison of jitter.

6.2. End-to-end packet delivery ratio and delay performance

We next present the packet delivery ratio and delay performance of the proposed *IDAR* routing protocol in a more general simulation scenario. We consider the same grid network as shown in Fig. 3, and consider the case where all nodes are communicating with a common destination, node 29 (representing the Internet gateway). The sources are selected randomly. Each flow runs *UDP* with a transmission rate of 35 KBps. Each flow is alive for 200 s and the average delivery ratio, delay, and jitter of the data flows are averaged over 10 runs for four different routing protocols: *AODV*, *MARIA*, *IDAR_v*1, and *IDAR_v*2. Jitter is measured by the variance of the delay obtained from multiple simulations. The results, obtained with *RTS/CTS* and *ACK* disabled, are shown in Fig. 14(a)–(c). It is observed that both *IDAR_v*1 and *IDAR_v*2 provide significantly better performance than *AODV* in terms of average delivery ratio, delay, and jitter. *IDAR_v*2 performs better than *IDAR_v*1, as *IDAR_v*2 requires the destination to measure the quality of a route after getting all the intermediate nodes in the route, while in *IDAR_v*1 the quality is calculated in the intermediate routers which only know the *existing* active nodes and query packet information. While *MARIA* gives a slightly higher delivery ratio than *IDAR-v2*, *IDAR_v*2 provides a significant improvement in delay and jitter over *MARIA*. The reason is that *MARIA* only chooses the route based on higher residual bandwidth i.e. lesser interference, without considering the delay. However, delay is an important parameter for determining quality in many applications.

We next present the performance of *IDAR-v2* with *RTS/CTS* while varying the number of flows. Fig. 15(a)–(c) show the comparison of average delivery ratio, delay and jitter of *AODV* and *IDAR_v*2 with *RTS/CTS* when the transmission rate is set to 65 KBps. It is observed that *IDAR_v*2 with *RTS/CTS* gives a higher delivery ratio and lower delay and jitter than *AODV*. In Fig. 16(a)–(c), we fix the number of flows to 10, and vary the data rate. The results show that *IDAR_v*2 gives better delivery ratio, delay, and jitter than *AODV* at all data rates. This is because of the ability of *IDAR* to choose higher quality routes than the shortest routes to gateways. From Fig. 16(b) and (c) we can observe that for lower load the improvement of delay and jitter in *IDAR* over *AODV* is not very significant, but for higher load, the difference is significant. Thus our proposed scheme is more efficient in heavy loads.

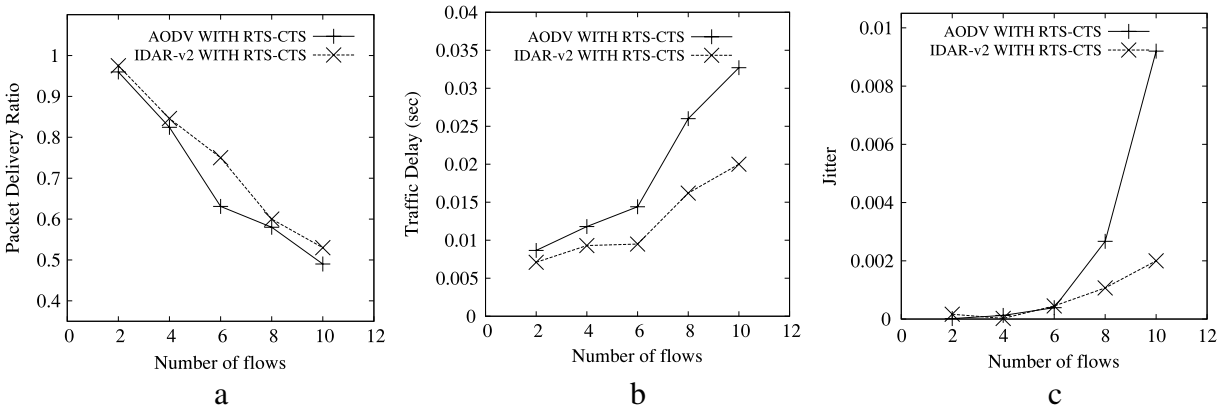

Fig. 15. (a) Comparison of packet delivery ratio. (b) Comparison of delay. (c) Comparison of jitter.

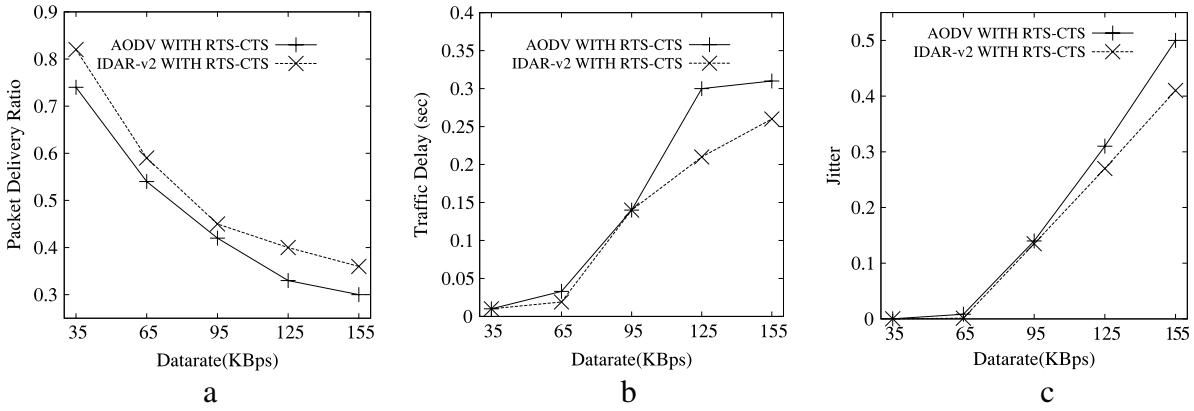

Fig. 16. (a) Comparison of packet delivery ratio. (b) Comparison of delay. (c) Comparison of jitter.

Table 4
Comparison of packet delivery ratio.

Load (KBps)	*MARIA*	*IDAR_v2*
5	0.988	0.9799
15	0.969	0.96
25	0.945	0.936
35	0.923	0.917

Table 5
Comparison of delay (s).

Load (KBps)	*MARIA*	*IDAR_v2*
5	0.0176	0.0058
15	0.0176	0.0058
25	0.01765	0.0058
35	0.01769	0.0058

Finally, we take a specific example to demonstrate the potential of *IDAR* in controlling the end-to-end delay, which is not achieved in *MARIA*. We consider the scenario shown in Fig. 17, where node 16 has already established the route $16 \rightarrow 22 \rightarrow 28 \rightarrow 29$ when node 5 starts looking for a route to 29. In this scenario, because of the existing interference from the nodes on the route from 16 to 29, *MARIA* chooses a really long route ($5 \rightarrow 4 \rightarrow 3 \rightarrow 2 \rightarrow 1 \rightarrow 7 \rightarrow 13 \rightarrow 19 \rightarrow 25 \rightarrow 26 \rightarrow 27 \rightarrow 28 \rightarrow 29$), suffering from a very high delay in comparison to the route chosen by *IDAR_v2* ($5 \rightarrow 11 \rightarrow 17 \rightarrow 23 \rightarrow 29$), whose delivery ratio is only slightly smaller. These results are shown in Tables 4 and 5. Thus, the proposed *IDAR* routing protocol does not sacrifice delay in trying to achieve a high delivery ratio, which is important in many applications.

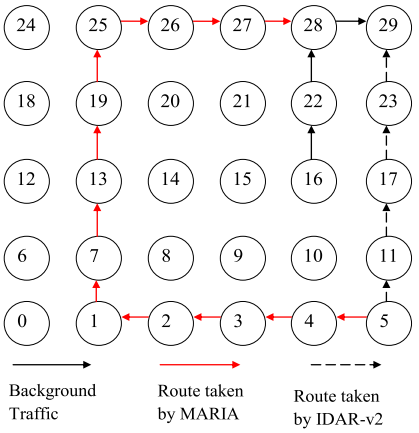

Fig. 17. Routes chosen by *MARIA* and *IDAR_v2* when the source–destination pair is 5 → 29 and 16 → 22 → 8 → 29 is the background traffic.

7. Conclusion and future work

We propose a quality based routing for wireless mesh networks to improve the packet delivery and delay performance of multi-hop communications. The proposed routing protocol *IDAR* is interference-aware and relies on a quality metric that is built on offline measurements of characteristics of data packet transmissions in a multi-hop network. *IDAR* takes two quality parameters into account: *POS* and the end-to-end delay. Performance evaluations obtained from simulation experiments demonstrate that the proposed routing protocol is effective in improving both the packet delivery and delay performance in multi-hop environments where the route selection is done by a gateway node that has global activity information of the network.

Since the proposed routing protocol relies on precalculated models of *POS* and delay, it is somehwat constrained by the parameters under which such models are based. This includes the assumed channel capacity, the offered transmission loads, etc. Although for the ease of representation, we assumed equal loads for all nodes, in practice loads may differ from node to node. This can be considered by the gateway node for incorporation into the corresponding quality metric as long as it has the models for those loads and the specific load information in the network.

The proposed quality based routing scheme can be extended to a network model that involves multiple gateway nodes, requiring anycast routing. In addition, our future work on this topic includes the extension of this quality based routing approach to incorporate multiple channels with multiple radios for each mesh router to reduce co-channel interference.

References

[1] I.F. Akyildiz, X. Wang, W. Wang, Wireless mesh networks: a survey, Computer Networks 47 (4) (2005) 445–487.
[2] Cuwin webpage, http://www.curwireless.net/.
[3] Smesh webpage, http://smesh.org/.
[4] Solarmesh webpage, http://owl.mcmaster.ca/todd/SolarMESH/.
[5] Wing webpage, http://www.wing-project.org/.
[6] A. Pal, S. Adimadhyam, A. Nasipuri, Qosbr: a quality based routing protocol for wireless mesh networks, in: ICDCN, 2010.
[7] X. Cheng, P. Mohapatra, S.-J. Lee, S. Banerjee, Maria: interference-aware admission control and qos routing in wireless mesh networks, in: ICC, 2008, pp. 2865–2870.
[8] R. Draves, J. Padhye, B. Zill, Routing in multi-radio, multi-hop wireless mesh networks, in: MOBICOM, 2004, pp. 114–128.
[9] D. Passos, D.V. Teixeira, D.C. Muchaluat-saade, L.C.S. Magalhes, C.V.N. Albuquerque, Mesh network performance measurements, in: International Information and Telecommunication Technologies Symposium, 2006.
[10] C.E. Koksal, H. Balakrishnan, Quality-aware routing metrics for time-varying wireless mesh networks, IEEE Journal on Selected Areas in Communications 24 (11) (2006) 1984–1994.
[11] A.P. Subramanian, M.M. Buddhikot, S. Miller, Interference aware routing in multi-radio wireless mesh networks, in: IEEE Workshop of Wireless Mesh Networks, 2006, pp. 55–63.
[12] V. Kolar, N.B. Abu-Ghazaleh, A multi-commodity flow approach for globally aware routing in multi-hop wireless networks, in: PerCom, 2006, pp. 308–317.
[13] E. Carlson, H. Karl, A. Wolisz, C. Prehofer, Distributed allocation of time slots for real-time traffic in a wireless multi-hop network, in: European Wireless, 2004.
[14] L. Romdhani, C. Bonnet, Cross-layer qos routing framework for wireless mesh networks, in: International Conference on Wireless and Mobile Communications, 2008, pp. 382–388.
[15] Z. Ke, L. Li, Q. Sun, N. Chen, A qos multicast routing algorithm for wireless mesh networks, in: SNPD vol. 1, 2007, pp. 835–840.
[16] V. Kone, S. Das, B.Y. Zhao, H. Zheng, Quorum — quality of service in wireless mesh networks, MONET 12 (5–6) (2007) 358–369.
[17] Q. Xue, A. Ganz, Qos routing for mesh-based wireless lans, IJWIN 9 (3) (2002) 179–190.
[18] C.H. Liu, K.K. Leung, A. Gkelias, A novel cross-layer qos routing algorithm for wireless mesh networks, in: Information Networking, 2008. ICOIN 2008. International Conference on, 2008, pp. 1–5.
[19] K. han Kim, K.G. Shin, On accurate measurement of link quality in multi-hop wireless mesh networks, in: In ACM MobiCom 06, 2006, pp. 38–49.
[20] The network simulator — ns-2, http://www.isi.edu/nsnam/ns/.