Determining the End-to-end Throughput Capacity in Multi-Hop Networks: Methodology and Applications

Yan Gao, Dah-Ming Chiu
Department of Information Engineering
The Chinese University of Hong Kong
Email: {ygao4, dmchiu}@ie.cuhk.edu.hk

John C.S. Lui
Department of Computer Science Engineering
The Chinese University of Hong Kong
Email: cslui@cse.cuhk.edu.hk

ABSTRACT

In this paper, we present a methodology to analytically compute the *throughput capacity*, or the maximum end-to-end throughput of a given source and destination pair in a multi-hop wireless network. The end-to-end throughput capacity is computed by considering the interference due to neighboring nodes, as well as various modes of hidden node interference. Knowing the throughput capacity is important because it facilitates the design of routing policy, admission control for realtime traffic, as well as load control for wireless networks. We model location-dependent neighboring interference and we use a contention graph to represent these interference relationships. Based on the contention graph, we formulate the individual link capacity as a set of fixed point equations. The end-to-end throughput capacity can then be determined once these link capacities are obtained. To illustrate the utility of our proposed methodology, we present two important applications: (a) *route optimization* to determine the path with the maximum end-to-end throughput capacity and, (b) *optimal offered load control* for a given path so that the maximum end-to-end capacity can be achieved. Extensive simulations are carried out to verify and validate the proposed analytical methodology.

Categories and Subject Descriptors

I.6.5 [**Simulation and Modelling**]: Model Development

General Terms

Performance

Keywords

Multi-hop ad hoc wireless networks, Throughput capacity, Analytical model for 802.11 protocols

1. INTRODUCTION

In a wired network, it is relatively straight-forward to allocate bandwidth and select routes for a set of flows in a way that is feasible for the network to support. Therefore, it is possible to study the bandwidth allocation problem as an optimization problem, and design distributed algorithms to achieve the desired goals. For a wireless multi-hop network, in particular, network which is based on the 802.11 protocol, the bandwidth allocation problem is much harder. Based on the relative positions of the transmitting and receiving nodes (or links), the mapping of flows (hence rates) to use these links introduce complex interference relationships between the links that any bandwidth allocation must obey. While classic papers[1, 2] established certain basic limits for the capacity of such wireless networks as a function of the number of nodes, the rules for bandwidth allocation and routing with a practical MAC protocol such as IEEE802.11 are poorly understood.

In this paper, we derive an analytical model for the following problem. Consider a multi-hop wireless network with a given set of flows each with a known path and a known end-to-end constant bit rate (CBR) in transmission, what is the maximum achievable throughput (also referred to as throughput capacity below) of a path if we inject a new flow onto that path without affecting the throughput of the existing flows? Such a model would be useful in various ways:

- Given different alternative paths between a source and destination, determine whether at least one of the paths would meet the throughput demand of a newly arriving flow, hence admit that flow if the throughput requirement is satisfied.
- Given a set of alternative paths, determine the best path if the newly arriving flow is elastic.
- Evaluate various rules for designing an efficient routing policy, for example, whether distance (e.g., hop count) between forwarding nodes is a good measure for a simple routing protocol, on other routing measures such as end-to-end throughput capacity.

Recently, other researchers[3] have studied ways to evaluate different paths in a multi-hop wireless network for the same purpose. The methodologies used are mainly experimental and of heuristic in nature. The difference, hence contribution of our work, is that we provide an *analytical methodology* of evaluating the throughput capacity which can be used for route optimization and load control for elastic traffic and admission control for inelastic traffic. Note that there will be more multimedia applications and one has to carefully consider how to manage this form of traffic for multi-hop wireless networks.

SIGMetrics/Performance'06, June 26–30, 2006, Saint Malo, France.

Our methodology is based on extending the model[4] for analyzing the maximum throughput of flows in *linear* 802.11 wireless network. The major result of [4] is that there is an *optimal* hop distance that can be used in a simple static routing protocol, which can simplify the design of routing policy in a 802.11 wireless ad-hoc network and at the same time, achieve good performance. Unlike the work in [4] which only considered symmetric flows, our new methodology considers *all* the links in use by *many different* flows in a wireless network. In particular, each link has two kinds of contention relationships with other links:

- direct or neighboring node contention,
- hidden node contention.

Such relationships can be represented by a contention graph (V, E, E') where wireless links are represented by nodes in V, and the neighboring node contention relationships are represented by undirected edges E, while the hidden node contention relationships are represented by directed edges E'. Each link's activity can be completely characterized by three variables: (i) *self air time*, which represents that a successful or a collided transmission is going on the link, (ii) *other's air time*, which represents when transmissions are going on for other contending links, and (iii) *idle time*, which is the time that no transmission is occurring from that link's view. The contention graph allows us to express a set of equations between these three variables for all links and solve them. The solution gives the throughput for each link in the newly arriving flow. Assuming each link belongs to only a single path, we can then readily derive the maximum throughput of each path (hence flow), as the throughput of the most constraining link on that path.

The balance of this paper is as follows. In Section 2, we first review the basic model of IEEE802.11's DCF, and introduce basic terminology and concepts of a multi-hop wireless network, in particular, the problem with different forms of hidden node interference. In Section 3, we develop the mathematical model for evaluating the end-to-end throughput capacity of a given flow in a wireless network. In Section 4, a detailed example is used to illustrate how the model works, and how the model can be applied to important applications like (1) routing optimization, (2) optimal offered load control. In Section 5, we compare the results of the analytical model with simulation experiments. In Section 6, we discuss how our work is related with existing literature. Lastly, Section 7 concludes.

2. SYSTEM MODEL

In this paper, we consider an ad-hoc network in which the underlying communication protocol is based on the 802.11 protocol. All nodes communicate using identical, half-duplex wireless radio based on the IEEE 802.11 DCF mode. The aim is to evaluate and determine the maximal end-to-end throughput capacity for a given source-destination pair. To include the possibility of communication contention, we assume that the carrier sensing range of each node is about two times of its transmission range (e.g., the transmission range is 250m and the carrier sensing range is 550m). The signal propagation is represented using the two-ray ground reflection model. Lastly, TCP has the built-in congestion control which may limit the potential end-to-end throughput capacity, therefore we assume that all data sources are UDP traffic streams with fixed packet size. In the following, we briefly explain the DCF mode and elaborate clearly the various *hidden* nodes problem in a wireless ad-hoc network.

2.1 DCF Model for IEEE802.11 Node

For the IEEE802.11 protocol, the fundamental mechanism to access the channel is based on the distributed coordination function (DCF). There are two access modes used in DCF, namely, the basic access mode and the RTS/CTS access mode. In this paper, we model the system for the basic access mode only because when the carrier-sensing range is larger than two times the transmission range, RTS/CTS is no longer needed or effective[5].

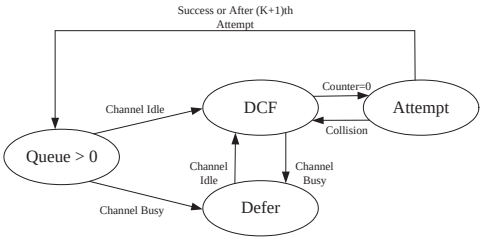

Figure 1: The State Diagram for an IEEE 802.11 Node.

Figure 1 depicts the DCF under the basic access mode. When a node has some packets to transmit, it needs to first sense the transmission medium, if the medium is busy then the node defers from transmission. If the medium is free for a specified time (which is the distributed interframe space, or DIFS), then the node enters the DCF state, in which the node initializes the backoff counter and resumes its cycle of sensing and count down. When the backoff counter reaches zero, the node makes an attempt to transmit its packet. This transmission may either succeed, or result in a collision. In the former case, a new packet will be selected from the queue and will start a new round of transmission attempt. In the latter case, it returns to the DCF state with new backoff timer value which is randomly chosen value between 0 and CW (contention window). If the maximum attempts K is reached, the node discards the packet and restarts at checking its data queue.

In general, a node with an nonempty queue (i.e., the node is operating at the saturated load) spends its time in one of three states. The time spent in the "DCF" state corresponds to the channel idle time; the time spent in the "Defer" state corresponds to the channel busy time due to other nodes transmission; the time spent in the "Attempt" state corresponds to the time the node itself is transmitting the packet.

Note that DCF adopts a binary exponential backoff scheme. At each packet transmission attempt, the backoff value is uniformly chosen in the range (0, CW-1). Under the 802.11 standard, the value CW (called contention window) depends on the number of failed attempts for the packet transmission. At the first transmission attempt, CW is set equal to the value CW_{min}, which is called the minimum contention window. After each unsuccessful attempt, CW is doubled according to the rule of $CW = 2^k CW_{min}$, where k denotes retransmission attempt with value up to K, after the K^{th} retransmission, the packet succeeds in transmission or is discarded.

According to these properties, Bianchi[6] first provided a Markov chain model for DCF behavior and applied it to analyze *single-cell* 802.11 networks. Later, authors of [7] derived a general formula relating the collision probability γ to the attempt rate per idle slot[1] by a node. This is denoted as $G(\gamma)$ and is represented as:

$$G(\gamma) = \frac{1 + \gamma + \gamma^2 \cdots + \gamma^K}{b_0 + \gamma b_1 + \gamma^2 b_2 \cdots + \gamma^K b_K}. \quad (1)$$

The numerator is the expected number of attempts of transmitting a single packet. In the denominator, b_k denotes the mean backoff duration (in time slots) at the k^{th} retransmission for a packet, $0 \leq k \leq K$; Therefore, the denominator represents the expected total back-off duration for a packet. For the 802.11 protocol, $b_0 = 16$, which is the expectation of a random valuable uniformly distributed in the range of 1 to 31 ($CW_{min} - 1$). Similarly, $b_1 = 32$, $b_2 = 64$, ..., $b_6 = 1024$. So the model of IEEE802.11 DCF using binary exponential algorithm yields attempts per idle slot in term of collision probability as:

$$G(\gamma) = \frac{1 + \gamma + \gamma^2 \cdots + \gamma^6}{16 + 32\gamma + 64\gamma^2 \cdots + 1024\gamma^6}. \quad (2)$$

2.2 Hidden Node Problem

In analyzing the performance of wireless multi-hop networks, one always needs to consider the impact of hidden nodes. Hidden nodes are the possible interfering nodes which *cannot* be sensed by the sender. The RTS/CTS mechanism was introduced in IEEE802.11 to deal with this problem. However, the use of RTS/CTS does not eliminate the hidden node problems completely in multi-hop networks [5, 8]. Due to the significant impact of hidden node interference, let us summarize the issues into two basic types of hidden nodes, namely, (i) *physical hidden nodes* and (ii) *protocol hidden nodes*.

To understand the "physical hidden node" problem, let us consider an example depicted in Figure 2.

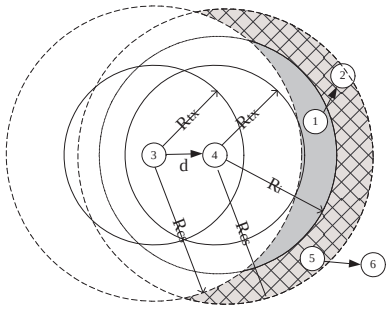

Figure 2: Physical and Protocol Hidden Node Problems

When node 3 transmits to node 4 at a distance d away, the received power at node 4 is proportional to $(1/d)^4$. Another node at distance r away from node 4 will cause interference unless the signal to interference power ratio (SIR) exceeds certain threshold. Assume the desired SIR threshold is 10, this implies that to avoid interference at node 4, the following condition needs to be satisfied:

$$SIR = P_r/P_i = (\frac{r}{d})^4 \geq 10$$

where P_r denotes the received power and P_i denotes the power of the interfering signal. This equation gives a *lower bound* on the distance r so that no interference will occur. Conversely, we can define an *interference range* as a distance R_i from the receiver such that nodes falling within that range may cause interference. Using the above SIR threshold, we have

$$R_i = \sqrt[4]{10} * d. \quad (3)$$

In Figure 2, given that node 3 is d away from node 4, any node within the interference range, which is represented by the circle centered at node 4 with radius R_i, may potentially interfere with the transmission from node 3 to node 4.

In general, two mechanisms can be used to protect the transmission from the physical hidden nodes interference: (i) the CTS sent from the receiver (node 4); and (ii) the sensing of node 3's transmission by the potential interferer. Protection mechanism (i) covers all the nodes in the circle centered at receiver (node 4) with radius R_{tx}, the transmission range of node 4 in sending the CTS. Protection mechanism (ii) covers all the nodes in the circle centered at the sender (node 3) with radius R_{cs}, the sensing range of the transmitter. Note that there is a shaded area in Figure 2, which is the area *within* the interference range but *outside* of both protection ranges, thus represents the area where potential physical hidden nodes reside. For instance, when node 3 is transmitting to node 4, node 1's transmission to node 2 will cause a collision due to the physical hidden node problem.

For the protocol hidden nodes problem, it occurs because the sender cannot hear as far as the receiver. To illustrate, consider the same situation in Figure 2, the cross-lined area which can be heard by node 4 is out of the sensing range of node 3. When a transmission from node 5 to node 6 is started first, any node hearing this transmission will be "frozen" (this is based on the IEEE 802.11 protocol). This implies that node 4 shuts itself down from receiving. But in this case, the sender (node 3) has no idea about what is taking place at node 5 (the interfering hidden node). To node 3, the channel is idle. Therefore, node 3 would transmit to node 4 while the transmission of node 5 is in progress. A protocol hidden node collision will occur, since no ACK will be sent by node 4 to node 3 because node 4 shuts itself down from receiving the data packet. Since this type of hidden node problem is caused by the limitation of the protocol, we name it the protocol hidden node problem.

It is important for us to point out the subtle difference between physical hidden node and protocol hidden node. In particular, the collision caused by physical hidden node may happen only if the hidden node transmits *after* the interfered node. Otherwise, if the hidden node started transmitting before the interfered node and a collision resulted, then it would be considered as a protocol hidden node induced collision.

To derive the throughput capacity of a given source-destination pair, one has to consider the neighboring nodes interferences, as well as the above mentioned hidden node interference. In the following section, we present the methodology to derive the throughput capacity.

[1]A slot is a unit of backoff time under the 802.11 protocol.

3. A METHODOLOGY TO COMPUTE END-TO-END THROUGHPUT CAPACITY

In this section, we present a methodology to compute the end-to-end throughput capacity of a given flow in an ad-hoc wireless network. We first show how to map an ad-hoc wireless network into a contention graph. Based on the contention graph, one can determine the potential interference between nodes. After that, we present an analytical model of an 802.11 DCF node, and show how to derive the channel idle probability and collision probability to yield the final end-to-end throughput capacity. To illustrate the methodology, we use an example to show how to apply the methodology to analyze throughput capacity.

3.1 Contention Graph

Given a set of wireless nodes and a set of flows, a network can be mapped into a contention graph[2]. This contention graph is used to represent the interference, i.e. which node is interfering with which nodes, and the types of interference. In this work, we consider two types of interference, namely, (a) hidden node contention, and (b) neighboring contention. Hidden node contention was described in the previous section while neighboring contention is due to the presence of wireless nodes within the sensing range of the transmitting node, and these wireless nodes also want to transmit packets.

We now present a general framework for mapping a network topology into the contention graph. This framework is a 3-step process.

1. Given the network topology, we generate an *undirected* graph that captures the neighborhood property, that is, nodes that are within the carrier-sensing range of a given node are considered as neighbors of that node.

2. Given the constructed undirected graph from the previous step and the set of active links (i.e., an active link connects a pair of transmitting and receiving nodes), we construct a contention graph $G = (V, E)$ where active links are represented by nodes in V and contentions among links are represented by undirected solid lines in E. Note that this graph provides information on all possible neighboring contentions.

3. Next, we need to deduce all hidden node contentions based on the definitions given in Section 2.2. We represent the hidden node interference in the final contention graph $G = (V, E, E')$, where the hidden node interference is represented by directed dot lines in E'. A directed edge in E' represents that the pointed node, which is an active link in a wireless node, is under a hidden node interference by the associated link.

To illustrate this concept, consider the network topology given in Figure 3. The solid circles represent nodes in the wireless network, while the dotted circle represents the sense range of the wireless node which is centered at the dotted circle. There are two flows in the network, one is from the source node A to the destination node E. Packets of this flow have to go through $link_1$, $link_2$, $link_3$ and $link_4$. The other flow is from the source node F to the destination node G. The flow goes through $link_5$.

[2]Previous work on contention graph only considered the interference due to neighboring nodes, while hidden node interferences were not modelled

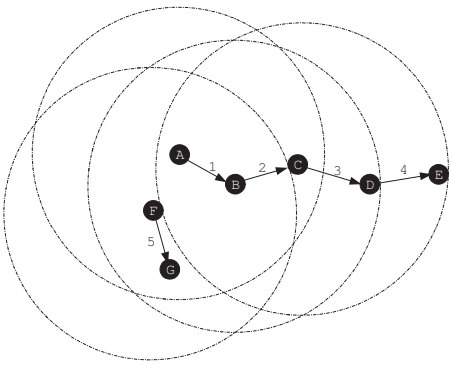

Figure 3: Network Topology with five active links

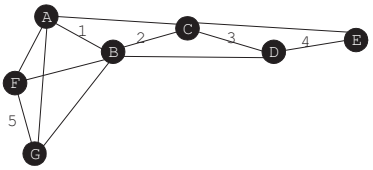

Figure 4: Undirected Graph

In Step 1, the network is represented as an undirected graph as shown in Figure 4. In this graph, node A is connected to node B, node C, node F and node G because these four nodes are within the carrier-sensing range of node A and they are considered as neighbors of node A (this can be observed from Fig. 3). Following the same argument, we create edges for each node in the undirected graph in Figure 4. Note that numbers in this graph are used to label all active links (i.e., $link_1$ to $link_5$) in the wireless network.

In Step 2, we transform all active links: $link_1$, $link_2$, $link_3$, $link_4$ and $link_5$, to nodes in the contention graph $G = (V, E)$ as shown in Figure 5. Each node in V represents a unique active link of the undirected graph in Figure 4. In particular, node 1 represents $link_1$, node 2 represents $link_2$, ..., etc. We connect node 1 to node 2, node 3 and node 5 because any transmission on these links can be sensed by the sender of $link_1$ (node A in this case). Note that this information can be deduced from the undirected graph in Figure 4 since node A and B are connected, this implies that when node A and node B transmit packets along $link_1$ and $link_2$ respectively, they will interfere with each other. Following the same argument, we can construct all edges for the contention graph in Figure 5.

In Step 3, we represent all hidden node contention. From the undirected graph in Figure 4, we observe that node D's

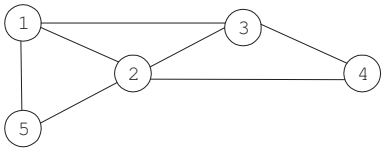

Figure 5: Contention Graph $G = (V, E)$

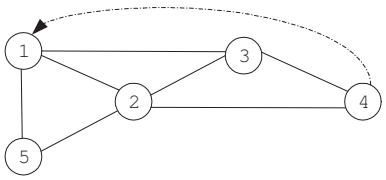

Figure 6: Contention Graph with Hidden Nodes $G = (V, E, E')$

transmission can be heard by node B (the receiver of $link_1$), but cannot be heard by node A (the sender of $link_1$). According to the discussion in Section 2.2, node D is the protocol hidden node of node A. This relation is indicated by a directed dot line from $link_4$ to $link_1$ as shown in Figure 6.

Using this procedure, one can generate the contention graph $G = (V, E, E')$. Note that this contention graph provides both active neighbor and active hidden node interference information. Based on the contention graph, we also define some useful notations which we will use in following subsection:

$\nu(i)$: the set of neighbors of $link_i$

$\mu(i, j)$: the set of common neighbors of $link_i$ and $link_j$

$\kappa(i)$: the set of hidden nodes of $link_i$

Consider the example in Figure 6, $\nu(1)$ is referred to node 2, node 3 and node 5; $\nu(5)$ is referred to node 1 and node 2; $\mu(1, 4)$ is referred to node 2 and node 3; $\mu(5, 3)$ is referred to node 1 and node 2; $\kappa(1)$ is referred to node 4 while $\kappa(5)$ is an empty set,..., etc.

3.2 Link Capacity under the 802.11 Model

Given a particular path for a source and destination node, the end-to-end throughput capacity is defined as the minimum link throughput capacity of this path. Therefore, in order to compute the throughput capacity of a path, we need to develop a methodology to compute an individual link capacity.

¿From a sending node's perspective, it's sending link, say link i, can be in one of three potential states: transmission state, channel busy state, and channel idle state. The activity of link i can be characterized by three variables:

(i) x_i: denoting the normalized "self" airtime, which includes the successful and collided transmission time.

(ii) y_i: denoting the normalized "busy" airtime, which is the time due to the transmission of contending links.

(iii) z_i: denoting the normalized "idle" time of link i, i.e., the time that the sender spends in counting down its backoff timer.

Consider a long stretch of time interval in [0, $Time$]. Let S_i be the transmission airtime within this interval that a "steady-state" node i transmits. Let $|S_i|$ be the length of this interval. This airtime includes the transmission times of data packets (PACKET), the transmission times of the acknowledgements (ACK), the durations of the distributed interframe space (DIFS) and the durations of the short interframe space (SIFS). The times used up for retransmission are also included in S_i.

From a particular node's perspective (i.e., node i), x_i is defined as follows:

$$x_i = \lim_{Time \to \infty} \frac{|S_i|}{Time}. \quad (4)$$

Due to the carrier-sensing property, any transmission within node i's carrier-sensing range leads to channel busy. The total airtimes used up by these transmissions is $\bigcup_{j \in \nu(i)} S_j$. Then we can define:

$$y_i = \lim_{Time \to \infty} \frac{|\bigcup_{j \in \nu(i)} S_j|}{Time}. \quad (5)$$

The channel is in idle state if there is no "self" transmission or neighbors' transmissions. We have

$$z_i = 1 - x_i - y_i. \quad (6)$$

Because we are interested in the link's capacity, we assume that the sender has a nonempty queue of packets. Thus, whenever the channel is sensed as idle, the sender will count down its back-off counter in order to transmit a packet. In Section 2.1, we represent the behavior of the DCF operation as a state transition diagram (Figure 1). Based on the state diagram, a node's view of channel is represented in Figure 7. From this figure, we see that as soon as the channel is

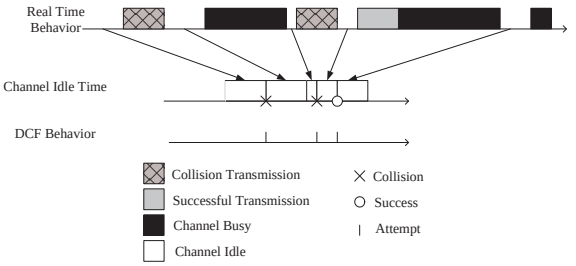

Figure 7: Real Time Sequence, and after removing the channel activities, the behavior of the DCF in channel idle slots

busy, DCF of the node will defer the count-down process; when the channel is idle, it will resume the process until the counter reaches zero. Then the node makes an attempt to transmit. It is clear that we can remove the channel activities (i.e., collision transmission, successful transmission and channel busy time), so that the DCF can be seen as an independent function running in the channel idle slots. Thus, when $G(\gamma_i)$ models the attempt rate per idle slot, with γ_i denoting the collision probability of transmission on link i, then the normalized "self" airtime x_i can be expressed as:

$$x_i = z_i \times G_i(\gamma_i) \times T \quad (7)$$

where T is the average packet length (in units of time slot). Since we consider a fixed sized UDP packets, T is a constant value which is equal to the packet size divided by the transmission rate. Note that z_i is the channel idle probability and $G_i(\gamma_i)$ is the attempt rate per idle slot, so $z_i \times G_i(\gamma_i)$ is the attempt rate per slot, because T denotes the time slots spent in each attempt, $z_i \times G_i(\gamma_i) \times T$ yields transmission time ratio which is equal to x_i.

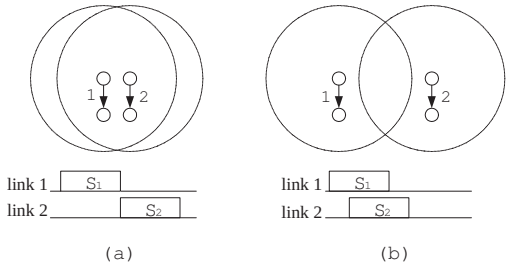

Figure 8: (a) when $link_1$ and $link_2$ can sense each other, then they do not overlap due to the 802.11 protocol; (b) when $link_1$ and $link_2$ cannot sense each other, then transmissions may overlap

Let E_i denote link throughput capacity of link i, by knowing x_i and γ_i, we can express E_i as

$$E_i = x_i \times (1-\gamma_i) \times \frac{T_1}{T} \times data\_rate. \quad (8)$$

Note that $1-\gamma_i$ is successful transmission probability, so $x_i(1-\gamma_i)$ is the fraction of the normalized airtime spent in successful transmitting. T_1 is the packet payload (in unit of time slot) and $data\_rate$ is transmission capacity of the 802.11 protocol. For instance, $data\_rate$ = 11 Mbps for 802.11b or $data\_rate$ = 54 Mbps for 802.11a.

3.3 Channel Idle Probability

To find E_i from Eq. (8), we need to compute x_i, which is a function of the channel idle probability z_i and the collision probability γ_i. From Eq. (6), the channel idle probability is

$$z_i = 1 - x_i - y_i.$$

From a node's perspective (i.e., node i), the channel busy time is the *union* of its neighbors' transmission airtimes $|\bigcup_{j\in\nu(i)} S_j|$. One can use the second order approximation to represent this union as

$$|\bigcup_{j\in\nu(i)} S_j| \approx \sum_{j\in\nu(i)} |S_j| - \sum_{m,n\in\nu(i)} |S_m \bigcap S_n| \quad (9)$$

Note that the second-order approximation is accurate enough because due to the property of carrier sensing of 802.11, the neighboring interference caused by *more* than two simultaneous transmissions is not significant.

However, the second-order intersection terms are still difficult to unfold because whether S_n and S_m overlap (simultaneously transmit without interfering each other) is location-dependent. For example, as shown in Figure 8(a) when $link_1$ and $link_2$ can sense each other, then they do not overlap due to the 802.11 protocol and so $|S_1 \bigcap S_2| = 0$; when $link_1$ and $link_2$ cannot sense each other as shown in Figure 8(b), then they may overlap and $|S_1 \bigcap S_2| > 0$. This implies that in order to complete the derivation, one must take the topology information into consideration and this topology can be obtained from the contention graph $G = (V, E, E')$.

To unfold the second-order intersection terms, we need to determine the intersections of airtimes used by any two nodes. To achieve this, we make the following assumptions.

1 If two nodes can sense each other, we assume:

$$|S_i \bigcap S_j| = 0, \quad i \in \nu(j);$$

2 If two nodes cannot sense each other and they have common neighbors, we assume:

$$|S_i \bigcap S_j| = \frac{|S_i||S_j|}{Time - \sum |S_c|} = \frac{x_i x_j}{1-\sum x_c} \cdot Time, c \in \mu(i,j);$$

3 Otherwise, $|S_i \bigcap S_j| = x_i x_j \cdot Time$.

The justification of assumption 1 is that if two nodes can hear each other, then their transmission airtimes will not overlap due to the carrier sensing property of 802.11 protocol. The justification of assumption 2 is that if two nodes cannot hear each other but have common neighbors, due to the carrier sensing property, the two nodes do not transmit within the airtime used by their common neighbors during $[0, Time]$, So the remaining fraction of airtime where the two nodes may overlap is $1-\sum x_c$. If two nodes cannot hear each other and have no common neighbor, we assume they will not interfere with each other. Following these assumptions, intersection terms of Eq. (9) can be unfolded when the topology information is given by the contention graph $G = (V, E, E')$. Therefore the channel idle probability is obtained as follows:

$$z_i = 1 - x_i - \sum_{j\in\nu(i)} x_j + \sum_{m\notin\nu(n)} \sum_{\bigcup n;\, m,n\in\nu(i)} \frac{x_m x_n}{1-\sum x_c} \quad (10)$$

3.4 Collision Probability

Remember that we need to know γ_i, the collision probability of link i, so that we can determine E_i, the capacity of that link. In this paper, we assume collisions are mainly due to hidden node interference because compared with other factors which contribute to collisions (e.g. more than one node attempts at the same slot), hidden node interference occurs with much higher frequency in multi-hop networks.

Let γ_{ik} denote the collision probability caused by the k^{th} hidden node of node i and γ_i denote the overall collision probability of node i. Let

$$a = \frac{\text{PACKET}}{\text{DIFS+PACKET+SIFS+ACK}}$$

be the fraction of time used for transmitting a data packet. So ax_i and ax_k are the normalized times spent in transmitting data packet for link i and k respectively. Let x_c be the normalized "self" airtime for node c, which is the *common neighboring node* for node i and k.

As we have discussed in Section 2.2, hidden nodes include both the *protocol* hidden nodes and the *physical* hidden nodes. Thus, the effect of hidden node node k can be classified in the following cases:

Case 1: *Node k is the protocol hidden node of node i:* As discussed in Section 2.2, protocol hidden node problem occurs if the hidden node starts transmitting before the interfered node and their transmissions overlap each other. Let A_1 be the event that ax_i overlaps ax_k, then the overlap probability $Prob\{A_1\} = ax_i + ax_k$. Let A_2 be the event that ax_k starts before ax_i, then $Prob\{A_1 A_2\} = ax_k$. But if there are common neighboring nodes between node i and node k, then ax_i and ax_k may overlap under the condition that these common neighboring nodes do not transmit. Let B be the event that these common neighboring nodes do not transmit, then $Prob\{B\} = 1-\sum x_c$. The collision probability

for this case is

$$\gamma_{ik}^{(1)} = Prob\{A_1 A_2 | B\} = \frac{a x_k}{1 - \sum x_c}, \quad k \in \kappa(i), \quad c \in \mu(i,k). \tag{11}$$

Case 2: *Node k is the physical hidden node of node i:* Node k may cause collision on node i when it starts transmitting after node i. Thus the collision probability caused by node k on node i is ax_i. Similarly, the airtime used by their common neighbors needs to be eliminated. The collision probability for this case is

$$\gamma_{ik}^{(2)} = \frac{a x_i}{1 - \sum x_c}, \quad k \in \kappa(i), \quad c \in \mu(i,k). \tag{12}$$

Case 3: *Node k is both the physical hidden node and the physical hidden node of node i:* Under this case, it does not matter whether node k starts transmitting before or after node i, it may cause collision on node i. Following similar derivation, the collision probability for this case is

$$\gamma_{ik}^{(3)} = \frac{a(x_k + x_i)}{1 - \sum x_c}, \quad k \in \kappa(i), \quad c \in \mu(i,k). \tag{13}$$

According to the above discussion, once we know the type of hidden node node k, we can determine the collision probability γ_{ik} caused by this node. The overall collision probability γ_i is the union of all individual collision probabilities ($\gamma_{ik},\ k \in \kappa(i)$). Since these probability may overlap each other, we also use the second-order approximation and the assumptions made in Section 3.3 to unfold the union expression. We have:

$$\gamma_i = \sum_{k \in \kappa(i)} \gamma_{ik} - \sum_{m \notin \nu(n) \bigcup n;\, m,n \in \kappa(i)} \frac{\gamma_{im}\gamma_{in}}{1 - \sum x_b}, \quad b \in \mu(m,n). \tag{14}$$

Now, we have obtained the expression of the channel idle probability z_i in terms of $x_j,\ j \in \nu(i)$ and collision probability γ_i in terms of $x_k,\ k \in \mu(i)$. Substituting these two results into Eq. (7), we have

$$x_i = \left(1 - x_i - \sum_{j \in \nu(i)} x_j + \sum_{m \notin \nu(n) \bigcup n;\, m,n \in \nu(i)} \frac{x_m x_n}{1 - \sum x_c}\right) G(\gamma_i) T. \tag{15}$$

This implies that we transform Eq. (7) into Eq. (15), in which we can use fixed point method to find the solution of all x_i.

Based on the above model, we can summarize the methodology to compute the link capacity as follows:

1. Given the network topology and the flow pattern, apply the framework proposed in Section 3.1 to constructing a contention graph $G = (V, E, E')$.
2. According to this contention graph, use Eq. (10) to formulate the channel idle probability and use Eq. (11), Eq.(12), Eq.(13) and Eq.(14) to formulate the collision probability for each individual link.
3. Substitute the channel idle probability and the collision probability into Eq. (15) to obtain a fixed-point equation.
4. Solve the fixed-point equation by the numerical method.

Note that in this paper, we are interested in the maximum throughput of the incoming flow. Thus there is only one independent source in a network and we assume that other interference traffic is given. That implies the fixed-point equation is one-dimensional, and all variables are known or associated with the same flow. In the following part, we are going to illustrate how to apply the methodology to computing the throughout capacity of a given path.

3.5 An Example of Computing The Throughput Capacity of a Path

To illustrate the methodology, we use an example to show how to compute the end-to-end throughput capacity for a given path. In this example, we use the same network as shown in Figure 3. The contention graph of this network is $G = (V, E, E')$, which is given in Figure 6. Based on this contention graph and Eq. (7), we derive a set of fixed point equations for each link.

Let x_1, x_2, x_3, x_4 and x_5 denote the normalized transmission airtime of $link_1$, $link_2$, $link_3$, $link_4$ and $link_5$ respectively. There are two flows in the network, flow 1 goes through $link_5$, flow 2 goes from $link_1$ to $link_2$ to $link_3$ and to $link_4$. Without loss of generality, we assume flow 1 is an existing traffic in the network and its traffic load is given. Flow 2, on the other hand, is a new incoming traffic. Our goal is to figure out the end-to-end throughput capacity that the flow 2 can achieve without affecting flow 1.

For flow 2, there are four links with four different link capacities. Thus the basic idea is to solve the four links individually and choose the minimum one as the end-to-end throughput capacity of this flow. Note that there is one protocol hidden node (i.e., node 4) in the network and according to Eq. (11), the collision probability γ_1 of link 1 is

$$\gamma_1 = \frac{a x_4}{1 - x_2 - x_3}. \tag{16}$$

Note that at the end only the minimum link capacity is regarded as the path capacity, so each link should have the same throughput as the end-to-end throughput of the flow. This implies that the links in the same path should have the same throughput and they must satisfy the *flow constraint*:

$$x_1(1 - \gamma_1) = x_2 = x3 = x_4. \tag{17}$$

Using Eq.(15), we can express the *link capacity equation* for each of the four links as follows:

$$\begin{aligned}
x_1 &= (1 - x_1 - x_2 - x_3 - x_5 + \frac{x_3 x_5}{1 - x_1 - x_2}) G(\gamma_1) T, \\
x_2 &= (1 - x_1 - x_2 - x_3 - x_4 - x_5 + \frac{x_1 x_4}{1 - x_2 - x_3} \cdot \\
&\qquad + \frac{x_5 x_3}{1 - x_1 - x_2} + \frac{x_5 x_4}{1 - x_2}) G(\gamma_2) T, \\
x_3 &= (1 - x_1 - x_2 - x_3 - x_4 + \frac{x_1 x_4}{1 - x_2 - x_3}) G(\gamma_3) T, \\
x_4 &= (1 - x_2 - x_3 - x_4) G(\gamma_4) T.
\end{aligned}$$

Then we substitute Eq. (17) (the *flow constraint*) and γ_1 Eq. (16) (since γ_2, γ_3 and γ_4 are equal to zero) into each equation above. By considering x_5 as a given parameter, we have four fixed point equations for four links.

For example, considering link 1, we have:

$$\begin{cases}
x_1 = (1 - x_1 - x_2 - x_3 - x_5 + \frac{x_3 x_5}{1 - x_1 - x_2}) G(\gamma_1) T, \\
x_2 = x_1(1 - \gamma_1), \\
x_3 = x_1(1 - \gamma_1), \\
x_4 = x_1(1 - \gamma_1), \\
\gamma_1 = \frac{a x_4}{1 - x_2 - x_3}.
\end{cases}$$

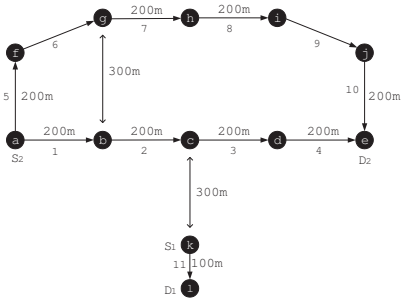

Figure 9: Example 2: Network Topology

After making substitutions, we get a fixed point equation set in terms of x_1 and γ_1:

$$\begin{cases} x_1 &= (1-3x_1+2x_1\gamma_1-x_5+\frac{x_1x_5(1-\gamma_1)}{1-2x_1+\gamma_1x_1})G(\gamma_1)T, \\ \gamma_1 &= \frac{ax_1(1-\gamma_1)}{1-2x_1(1-\gamma_1)}. \end{cases}$$

For these fixed point equations, we turn to numerical method for the solutions of x_i and γ_i, for $i \in \{1,2,3,4\}$. Then we can use Eq. (8) to compute the throughput capacity of each link. We have four candidate throughput capacities, and the minimum one is regarded as the bottleneck of the path throughput. As a result, it is the end-to-end throughput capacity of this flow. We will present the numerical solutions with different values of x_5 in Section 5, together with validation via simulation.

4. APPLICATIONS

In the previous section, we proposed an analytical model to determine the end-to-end throughput capacity of a path. To illustrate the utility of our proposed methodology, we provide two important applications, namely *optimal routing* and *optimal offered load control*.

To illustrate the utility of the model, we use an example whose topology is shown in Figure 9. There are two flows going through the network, flow 1 is generated by source S_1 (node k) to destination D_1 (node l); flow 2 is generated by source S_2 (node a) to destination D_2 (node e). In this example, flow 1 is regarded as an existing flow and the issue is how to perform optimal routing and load control on a new flow, namely, flow 2.

4.1 Routing Optimization

Given a path in a multi-hop network, the end-to-end throughput is determined by the minimum link capacity of the path. So if we can figure out each link's capacity of a given path, then the minimum one is the end-to-end throughput capacity of this path. By knowing all candidate paths' capacity, one can choose the path with maximum throughput so as to deliver packets.

Let E_i^k denote the throughput capacity of $link_i^k$ in $path_k$, the optimal path is determined by

$$OptimalPath = \{path_j | Capacity(path_j) = \max_k \{\min_i E_i^k\}\} \quad (18)$$

Traditional routing such as the shortest-path routing assumes that the link cost is fixed. In recent study, however, people found that the shortest path is far from the optimal in multi-hop wireless networks. For example, in [9] authors investigated the performance of multi-hop wireless networks and argued that shortest path is not enough to achieve high throughput and more attention should be paid to link quality when choosing ad hoc routes. Note that all these investigations are done in an experimental and heuristic manner. In here, we use the proposed methodology to determine the optimal path which has the highest throughput.

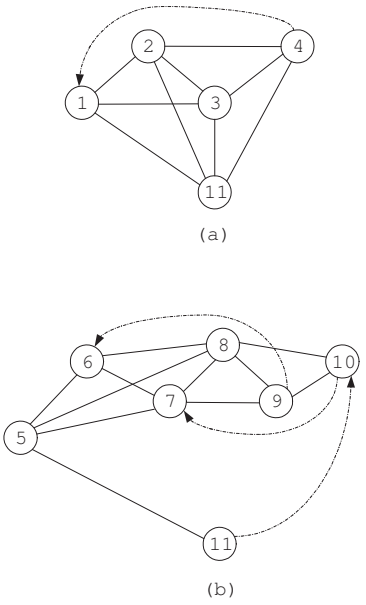

Figure 10: (a) Contention Graph for Path 1; (b) Contention Graph for Path 2.

For Figure 9, flow 2 (i.e. from S_2 to D_2) has two candidate paths: Path 1 is $a \to b \to c \to d \to e$, which is of four hops, and Path 2 is $a \to f \to g \to h \to i \to j \to e$, which is of six hops. If the shortest-path routing algorithm is used, then Path 1 will be used to deliver the packets from S_2 to D_2.

Now let us compute the capacity of the two paths by the proposed methodology. First, the topology can be mapped into a contention graph for Path 1 and 2 respectively, as shown in Figure 10. Figure 10 (a) depicts the contention graph for Path 1, while Figure 10 (b) depicts the contention graph for Path 2.

Assume the throughput of the existing flow 1 is 3Mbps, and other parameters[3] are $data\_rate = 11Mbps$, $T = 84$ and $T_1 = 55$, one can compute x_{11} based on Eq. (8) and we have $x_{11} = 0.42$. For Path 1, the *flow constraint* is:

$$x_1(1-\gamma_1) = x_2 = x_3 = x_4 \quad (19)$$

since $link_1$ will be affected by $link_4$ due to the hidden node d. Based on Eq. (11), the collision probability γ_1 is

$$\gamma_1 = \frac{ax_4}{1-x_2-x_3-x_{11}}. \quad (20)$$

Solving four sets of fixed point equations, we can obtain capacities of these four links and they are listed in Table 1.

For Path 2, the *flow constraint* is:

$$x_5 = x_6(1-\gamma_6) = x_7(1-\gamma_7) = x_8 = x_9 = x_{10}(1-\gamma_{10}) \quad (21)$$

[3]Note that the units of T and T_1 are in units of time slots, with each time slot being 20 μs, based on the 802.11 protocol.

$link_i$	Link Capacity Equation	Computed Capacity
$link_1$	$x_1 = (1 - x_1 - x_2 - x_3 - x_{11})G(\gamma_1)T$	1.17 Mpbs
$link_2$	$x_2 = (1 - x_1 - x_2 - x_3 - x_4 - x_{11} + \frac{x_1 x_4}{1-x_2-x_3-x_{11}})G(0)T$	1.49 Mpbs
$link_3$	$x_3 = (1 - x_1 - x_2 - x_3 - x_4 - x_{11} + \frac{x_1 x_4}{1-x_2-x_3-x_{11}})G(0)T$	1.49 Mpbs
$link_4$	$x_4 = (1 - x_2 - x_3 - x_4 - x_{11})G(0)T$	1.30 Mpbs

Table 1: Throughput Capacity of Path 1.

$link_i$	Link Capacity Equation	Computed Capacity
$link_5$	$x_5 = (1 - x_5 - x_6 - x_7 - x_8 - x_{11} + \frac{x_{11}(x_6+x_7+x_8)}{1-x_5})G(0)T$	1.40Mbps
$link_6$	$x_6 = (1 - x_5 - x_6 - x_7 - x_8)G(\gamma_6)T$	1.39Mbps
$link_7$	$x_7 = (1 - x_5 - x_6 - x_7 - x_8 - x_9 + \frac{x_9(x_5+x_6)}{1-x_7-x_8})G(\gamma_7)T$	1.37Mbps
$link_8$	$x_8 = (1-x_5-x_6-x_7-x_8-x_9-x_{10} + \frac{x_5 x_9}{1-x_7-x_8} + \frac{x_5 x_{10}}{1-x_8} + \frac{x_6 x_9}{1-x_7-x_8} + \frac{x_6 x_{10}}{1-x_8} + \frac{x_7 x_{10}}{1-x_8-x_9})G(0)T$	1.44Mbps
$link_9$	$x_9 = (1 - x_7 - x_8 - x_9 - x_{10} + \frac{x_7 x_{10}}{1-x_8-x_9})G(0)T$	1.78Mbps
$link_{10}$	$x_{10} = (1 - x_8 - x_9 - x_{10})G(\gamma_{10})T$	2.13Mbps

Table 2: Throughput Capacity of Path 2.

because $link_6$ ($link_7$ or $link_{10}$) will be affected by $link_9$ ($link_{10}$ or $link_{11}$) due to the hidden node problem. Based on Eq. (11), the collision probabilities are:

$$\begin{aligned} \gamma_6 &= ax_9/(1 - x_7 - x_8), \\ \gamma_7 &= ax_{10}/(1 - x_8 - x_9), \\ \gamma_{10} &= ax_{11}. \end{aligned}$$

Solving these six sets of fixed point equations, we can obtain capacities of these six links and they are listed in Table 2.

From Table 1, we see that the bottleneck is link 1 and the end-to-end throughput capacity of Path 1 is 1.17Mbps. From Table 2, we see that the bottleneck is link 7 and the end-to-end throughput capacity of Path 2 is 1.37Mbps. This shows that the longer path (Path 2) performs better than the shorter path. The analysis of this example theoretically supports the intuition that when the shortest path is under severe interference, this may have a adverse effect on the end-to-end performance[10]. On the other hand, some longer hop paths (e.g. Path 2 in our example) may achieve better performance by avoiding the interference hotspot. Note that this opinion was mostly investigated in a heuristic and experimental manner in previous work. In here, we proposed a *quantitative analytical methodology* to systematically evaluate the path performance.

One direct application of the above example is on performing *admission control* for multimedia traffic in wireless networks. Using the example above, if the throughput requirement of the multimedia application is 1.3 Mbps, then one has to choose Path 2 to deliver the information. On the other hand, if the throughput requirement of the multimedia application is 2.0 Mbps, then one should not admit the flow into the network since there is no available resource to satisfy the given requirement.

4.2 Offered Load Control

In multi-hop networks, offered-load control can significantly improve system performance. There are number of studies on this issue. In [10], authors investigated a linear network and observed that the maximum throughput is achieved when the offered load is controlled to a certain value. They gave the explanation that the source node injects more traffic than the subsequent nodes can handled. These packets are eventually dropped at the down stream nodes. Therefore, the time the source spends in these dropped packets is wasted and thereby reduces the end-to-end throughput. Subsequent work on this issue was carried out by [8], in which the authors proposed an analytical framework of load control for a linear network. However, for general case, that is, in what situation and to what extent the offered load should be controlled is still an open issue.

Let us illustrate the utility of our proposed methodology in solving the offered load control problem. In here, we use the network topology as shown in Figure 9. This time, we focus on Path 1 and Path 2's capacity without other neighboring interference. This can be achieved by setting the data rate of flow 1 to zero ($x_{11} = 0$). Using the methodology we described in Section 4.1, the link capacity of each link is shown in Table 3.

Path 1	$link_1$	$link_2$	$link_3$	$link_4$		
Capacity	1.72	1.78	1.78	2.24		
Path 2	$link_5$	$link_6$	$link_7$	$link_8$	$link_9$	$link_{10}$
Capacity	1.44	1.39	1.37	1.44	1.78	2.24

Table 3: Link Capacity (in Mbps) when x_{11}=0.

From the table, we see that in Path 1, $link_1$ has the minimum capacity and is the bottleneck of the path; in Path 2, $link_7$ has the minimum capacity and acts as the bottleneck. We carry out a simulation experiment to study this scenario. When unlimited traffic (or very high traffic rate) is generated from the source node along Path 1, most packets are dropped in $link_1$; whereas when unlimited traffic is generated from the source along Path 2, most packets are dropped in $link_7$. This experiment, to some extent, verifies our model. The simulation detail and the accuracy of our model will be presented in the next section. We also adjust the packet generation rate at the source node to the capacity of Path 1 and 2 respectively, and the analytical end-to-end capacity can be achieved. Lastly, an interesting phenomenon that we observed is that offered load control seems to be more beneficial for Path 2 than Path 1. In other words, the end-to-end throughput is significantly enhanced in Path 2 when offered load control is applied while little performance benefit is gained in Path 1.

Based on the analysis and the simulation study, we know that in order to achieve the optimal throughput, whether a flow needs the offered-load control or not depends on the *location* of the bottleneck link. If the capacity bottleneck is in the first link (i.e., $link_1$ of Path 1), then the offered-load control will not improve the throughput performance. Packets are dropped in the first link and it won't affect the performance of subsequent links. Whereas if the capacity bottleneck is in the subsequent link (i.e., $link_7$ of Path 2), then the source needs to perform the offered load control so as to achieve higher throughput. This suggests an alterna-

tive routing strategy and offered load control are important and may enhance the performance of multi-hop wireless networks.

5. SIMULATION EXPERIMENT

In this section, we validate our analytical results by network simulator ns-2. First, we examine the accuracy of our model. Then we observe the neighboring nodes effect on the throughput capacity and compare simulation results with the analytical results. Finally, we verify our results for the application on optimal routing and optimal offered load control.

Our simulation environment is created using the simulator ns2.28 [11], which simulates wireless networks based on IEEE802.11 standard. Table 4 shows the system parameters used in the simulation and the RTS threshold is set to 5000 so that nodes do not need to use the RTS/CTS handshake mechanism. According to these parameters, one can com-

Transmission range R_{tx}	250m
Carrier-Sensing range R_{cx}	550m
CPThreshold	10dB
Propagation model	TwoRayGround
Packet payload	1500 bytes
UDP header	20 bytes
MAC header	28 bytes
PHY header	24 bytes
ACK frame	38 bytes
Channel bit rate	11 Mbps
PHY header bit rate	1 Mbps
Slot time	20 μs
SIFS	10 μs
DIFS	50 μs
CW_{min}	31
CW_{max}	1023
Retransmission limit	7
Traffic pattern	CBR
Transport protocol	UDP
Routing protocol	DSDV

Table 4: Simulation Parameters

pute packet payload length $T_1 = (1500 \times 8/11)/20 = 54.55$ (time slots) and packet length[4] $T = T_1 + (24 \times 8 + ((28 + 20) \times 8)/11 + 10 + 38 \times 8 + 50)/20 = 84.09$ (time slots).

5.1 Single Flow Capacity

Now we simulate single-flow cases with 1-hop, 2-hop or 3-hop path length to validate our model. Let us consider a linear network as shown in Figure 11. Assume there is a single flow generated in node a and it goes to node b, node c or node d respectively. Thus we have three cases to simulate: 1-hop case (node a to node b), 2-hop case (node a to node c) and 3-hop case (node a to node d). For each simulation case, the sender delivers packet with unlimited traffic rate (or very high traffic rate) so as to achieve the end-to-end throughput capacity.

The analytical throughput capacity of 1-hop case is computed by our methodology as follows: Since there is no collision for a single link, collision probability $\gamma = 0$. The

[4]We consider a 802.11 packet frame including PHY header, MAC header, UDP header, data packet, SIFS, ACK frame and DIFS.

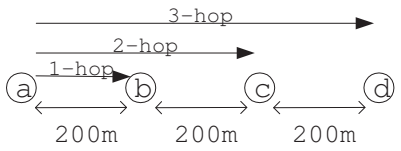

Figure 11: A Linear Network

normalized "self" airtime is $x = (1 - x)G(0)T$, then $x = \frac{G(0)T}{1+G(0)T}$. Substitute it into Eq. (8), we have throughput $E = \frac{G(0)T_1}{1+G(0)T} \times 11 = 6.05Mbps$. Similarly, one can compute throughput capacity of 2-hop case and 3-hop case. These computed results and the simulation results are illustrated in Figure 12 As one can observe, the throughput capacity of

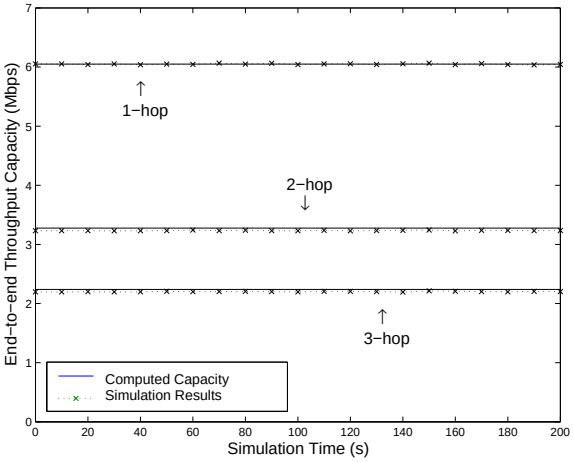

Figure 12: Single Flow Throughput Capacity

more hops is lower than fewer job. This is due to the interference of neighboring nodes. Lastly, this figure shows that the computed throughput capacity is well matched with the real end-to-end throughput capacity.

5.2 Neighboring Traffic Effect

This simulation is carried out to observe neighboring interference on the end-to-end throughput capacity. We use the same network as depicted in Figure 3. Flow 1 is regarded as neighboring traffic and we vary its traffic sending rate. For each traffic rate of flow 1, we inject unlimited traffic load into flow 2 so as to achieve the throughput capacity of flow 2. Each simulation runs for 200s simulation time. Figure 13 shows the simulation results and the computed throughput capacities.

We observe that:

- The computed end-to-end throughput capacity is well matched with the simulation results.
- As the neighboring traffic (i.e., flow 1) rate increases, the maximum end-to-end throughput of flow 2 decreases since the transmission on $link_5$ will affect the performance on $link_1$ and $link_2$.

This simulation demonstrates that the proposed methodology can accurately compute the end-to-end throughput capacity of a given flow in a multi-hop network.

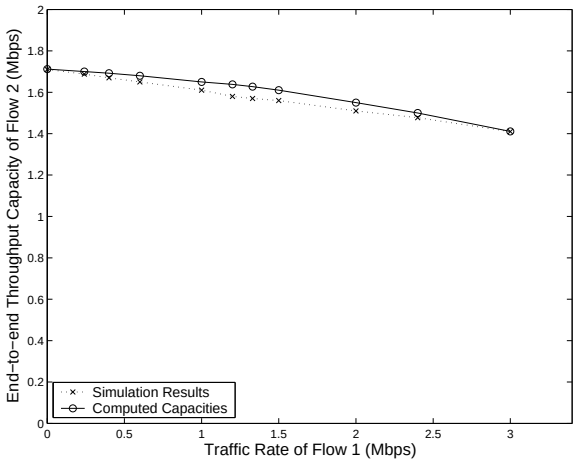

Figure 13: Neighboring Interference on Throughput Capacity of Flow 2

5.3 Routing Optimization

In this simulation, we validate the *route optimization* claim in Section 4.1 that when the shortest path is under severe interference, this may have an adverse effect on the end-to-end performance. On the other hand, some longer hop paths may achieve better performance by avoiding the interference hotspot.

We use the same network as presented in Figure 9. Flow 1 is regarded as an existing flow. Flow 2 has two candidate path: Path 1 and Path 2. We vary flow 1's sending rate and observe its effect on the throughput capacities of the two paths. The simulation is carried out in the following manner: given the sending rate of flow 1, flow 2 is manually routed to Path 1 and Path 2 in two different simulation settings. For each simulation setting, we adjust the sending rate of flow 2 until the maximum sustainable throughput is achieved and this maximum sustainable throughput is regarded as the end-to-end throughput capacity of this path. Figure 14 shows the simulation results and the computed throughput capacities.

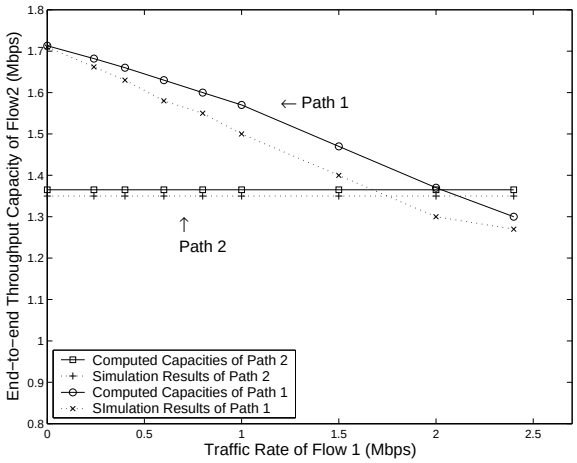

Figure 14: Throughput Capacities of Path 1 and Path 2 as a function of the sending rate of Flow 1

From Figure 14, we observe that:

- The end-to-end throughput capacity of Path 1 decreases as the traffic rate of flow 1 increases.
- The end-to-end throughput capacity of Path 2 remains unaffected by the traffic rate of flow 1.
- When the rate of flow 1 exceeds 1.7 Mbps, the throughput capacity of Path 1 is less than that of Path 2.
- The computed capacities of both paths closely match with simulation results.

The end-to-end throughput capacity of Path 1 is affected by flow 1 because flow 1 interferes with $link_1$, $link_2$, $link_3$ and $link_4$. Since $link_1$ is the bottleneck of Path 1, one should expect the degradation of its throughput capacity when flow 1 increases its rate. On the other hand, flow 1 can only interfere with $link_5$ for Path 2. However, $link_7$ is the bottleneck of this path, therefore, the interference of flow 1 on this path is negligible. This simulation confirms that when the shortest path is under severe interference, this may have an adverse effect on the end-to-end performance and one should consider other paths which may have higher capacity.

5.4 Optimal Offered Load Control

We carry out this simulation to illustrate that if the capacity bottleneck is in the first link, offered load control will not improve the throughput performance, whereas if the capacity bottleneck is in the subsequent link, offered load control will significantly improve the throughput performance.

In this simulation, we also use the network as present in Figure 9. This time we do not consider neighboring interference by setting the traffic rate of flow 1 to zero. We inject different traffic loads into Path 1 and Path 2 respectively and observe their end-to-end throughput. The results are shown in Figure 15 From this figure, one can observe that:

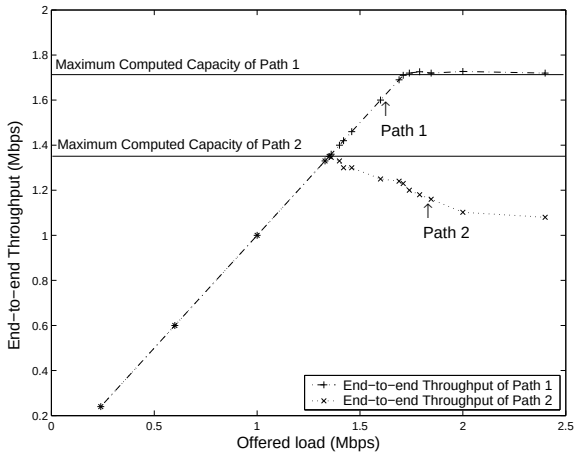

Figure 15: End-to-end Throughput vs. Offered Load

- Before reaching the maximum end-to-end throughput capacity, the end-to-end throughput of each path increases as the offered load increases.
- After reaching the maximum end-to-end throughput capacity, Path 1's throughput does not change as the offered load increases but Path 2's throughput decreases as the offered load increases.

- Either Path 1 or Path 2's maximum end-to-end throughput matches with the computed capacity very well.

Path 1's throughput performance does not degrade when the overloaded traffic is injected because the bottleneck of Path 1 is in the first link of this path. The redundant packets are dropped in the first link and it won't affect the performance of subsequent links. Path 2's throughput performance degrades when the overloaded traffic is injected because the bottleneck of Path 2 is in the subsequent link. The redundant packets are eventually dropped at the down stream nodes. The time the source spends in these dropped packets is wasted and thereby reduces the end-to-end throughput.

These results show that if capacity bottleneck is in the subsequent link, rather than transmitting information beyond the maximum throughput capacity, offered-load control will significantly improve throughput performance. And we believe our methodology can be used to predict the optimal offered load in multi-hop networks.

6. RELATED WORK

A number of papers have been published on the issue of determining the capacity of a multi-hop wireless network. For example, the seminal paper by Gupta and Kumar [1] derived theoretical bounds for the capacity of wireless networks. [2] subsequently analyzed the capacity of hybrid networks. Several researchers [6, 7, 12] analyzed the performance of IEEE802.11 DCF based network. But these papers considered that every node can hear every other nodes in the network (i.e., all nodes within a single cell), which is not the case for a multi-hop network. In [13, 14], undirected contention graph was proposed to study source allocation issues. But they only considered the interference due to neighboring node contentions, while hidden node interferences were not modelled. [15, 16, 9, 10] studied different heuristic routing policies based on prototype and measurements. Whereas we propose an analytical methodology which can systematically evaluate the end-to-end throughput capacity of a given flow in a multi-hop network. Recently, several analytical methods are proposed in [8, 4] but the analysis only applies on a linear wireless network. Our work provides a general analytical method that analyze a more general form of multi-hop wireless networks.

7. CONCLUSION

In this paper, we propose a methodology to analytically compute the *throughput capacity*, or the maximum end-to-end throughput of a given flow in a multi-hop wireless network. We considered two key factors which affect the end-to-end throughput capacity: (a) neighboring contentions, and (b) hidden node interference. The contributions of our work are: (1) We propose a contention graph to represent both neighboring interference and hidden node interference. (2) We consider neighboring interference not only by the number of neighboring nodes but also depends on the relative location between neighbors. (3) We propose a *fixed point functional model* for analyzing the link capacity, and thereby the end-to-end throughput capacity of a flow in a multi-hop wireless network. We illustrate the utility of our method in performing routing optimization, admission control and offered load control. We believe the proposed methodology can provide a systematic design of wireless networks.

8. ACKNOWLEDGMENTS

The work was partially supported by RGC grant 4232/04E (project 2150420).

9. REFERENCES

[1] P.Gupta and P.R.Kumar, "The capacity of wireless network," *IEEE Trans. On Information Theory*, vol. 46, pp. 388–404, March 2000.

[2] B. Liu, Z. Liu, and D. Towsley, "On the capacity of hybrid wireless networks," in *IEEE Infocom*, 2003.

[3] D. D. Couto, D. Aguayo, J. Bicket, and R. Morris, "High-throughput path metric for multi-hop wireless routing," in *ACM MobiCom'03*, September 2003.

[4] Y. Gao, D. M. Chiu, and J. C. Lui, "The fundamental role of hop distance in ieee802.11 multi-hop ad hoc networks," in *ICNP 2005*, November 2005.

[5] K. Xu, M. Gerla, and S. Bae, "How effective is the ieee 802.11 rts/cts handshake in ad hoc network," in *IEEE GlobeCom'02*, (Taipei, Taiwan), November 2002.

[6] G.Bianchi, "Performance analysis of the ieee802.11 distributed coordination function," *IEEE Journal on Selected Areas in Communications*, March 2001.

[7] A. Kumar, E. Altman, D. Miorandi, and M. Goyal, "New insights from a fixed point analysis of single cell ieee802.11 wireless lans," in *Proceedings of the IEEE Infocom*, March 2005.

[8] C. Ng and S. Liew, "Offered load control in ieee802.11 multi-hop ad-hoc networks," in *The 1st IEEE International Conference on Mobile Ad-hoc and Sensor System*, October 2004.

[9] D. D. Couto, D. Aguayo, B. Chambers, and R.Morris, "Performance of multihop wireless networks: Shortest path is not enough," in *In Proceedings of the First Workshop on Hot Topics in Networks (HotNets1)*, (New Jersey, USA), October 2002.

[10] J. Li, C. Blake, D. S. D. Couto, H. Lee, and R. Morris, "Capacity of ad hoc wireless networks," in *ACM MobiCom'01*, July 2001.

[11] *The Network Simulator-ns2, www.isi.edu/nsnam/ns.*

[12] G. Sharma, A. Ganesh, and P. Key, "Performance analysis of contention based medium access control protocols," in *INFOCOM 2006*, 2006.

[13] T. Nandagopal, T. E. Kim, X. Gao, and V. Bharghavan, "Achieving mac layer fairness in wireless packet networks," in *MobiCom*, 2000.

[14] M. Chiang, "Balancing transport and physical layers in wireless multi-hop networks: Jointly optimal congestion control and power control," *IEEE Journal on Selected Areas in Communications*, January 2005.

[15] J. Padhye, S. Agarwal, V. Padmanabhan, L. Qiu, A. Rao, and B. Zill, "Estimation of link interference in static multi-hop wireless networks," in *IMC*, 2005.

[16] K. Jain, J. Padhye, V. Padmanabhan, and L. Qiu, "Impact of interference on multi-hop wireless network performance," in *ACM MobiCom'03*, September 2003.