Model-based Bandwidth Estimation in IEEE 802.11 Wireless Networks

Yaling Yang, Jun Wang and Robin Kravets
University of Illinois at Urbana-Champaign
{ yyang8, junwang3, rhk@cs.uiuc.edu}

Abstract

Predicting the bandwidth achievable by a new flow is crucial for many bandwidth-aware services. In IEEE 802.11 based networks, the achievable bandwidth of a new flow is not necessarily the same as the free bandwidth on the channel due to the contention-based channel access method. In this paper, we develop a novel analytical model that provides closed-form calculation of bandwidth allocation in all possible operating states of single-hop IEEE 802.11 networks. Using this model, we design an algorithm that accurately predicts the achievable bandwidth of a new flow for single-hop wireless networks, such as wireless LANs, with or without service differentiation. In addition, we extend the bandwidth prediction model to multihop environments. Although the accuracy of the our bandwidth prediction model is degraded in multihop networks, our simulation results show that our algorithm still predicts more accurately than current existing methods.

I. Introduction

With the use of any network comes the desire to understand how much bandwidth is available to the flows in the network. Many services can be enabled when the network can expose information about the achievable bandwidth to a flow that is about to be added to the network or a flow that would like to increase its bandwidth usage. Such information can enable bandwidth-aware services such as admission control, load balancing, QoS management, bandwidth-sensitive applications or server selection. In any shared network using statistical multiplexing, such bandwidth prediction is challenging since existing flows may currently be using all of the bandwidth. However, the admission of a new flow can often "push" the competing flows to achieve its share of the bandwidth. Essentially, the problem of bandwidth prediction comes down to determining the achievable bandwidth of a flow, which is not necessarily the same as the free bandwidth in the network.

In current wired networks, all links are generally point-to-point with a fixed amount of bandwidth determined by the underlying physical medium. Since there is no competition with other nodes for link bandwidth, the management of bandwidth can occur locally at each node to support estimates of achievable bandwidth either on a per link or per flow basis. However, in wireless networks, all links are shared and there is no centralized control of how the channel bandwidth is shared between competing nodes. Predicting achievable bandwidth becomes much more challenging in wireless networks than in wired networks. The goal of our research, hence, is to model the competition for bandwidth between nodes so that accurate predictions of achievable bandwidth at individual nodes can be obtained in a shared medium wireless network. We mainly focus on one-hop IEEE 802.11 based wireless networks. An extension of our work to multi-hop networks is also discussed in Section VII.

The shared nature of wireless links has led to the use of CSMA/CA (carrier sense multiple access with collision avoidance) protocols like IEEE 802.11 [16] for communication in wireless LAN environments. Such protocols mediate access to the shared medium in an effort to provide competing nodes with fair access to the medium. Bandwidth allocation for competing nodes is achieved through competition using a contention resolution algorithm. Under such competition for bandwidth, a node cannot simply observe the unused capacity of a link to determine how much bandwidth it can achieve since competing nodes may currently be using all of the bandwidth. Instead, a node must understand the underlying competition between nodes vying for the bandwidth to predict the bandwidth it can achieve. For example, consider two competing nodes, A and B. If Node A always has data to send and Node B has no data, Node A uses all of the bandwidth. When Node B finally has data to send, although it observes a fully saturated link, the MAC protocol should be able to allocate bandwidth to Node B up to half of the channel bandwidth. In other words, although the observed available bandwidth is zero, knowledge of the competition between Nodes A and B can be used to predict that Node B can get half of the channel bandwidth.

Estimating achievable bandwidth in IEEE 802.11 based wireless networks can be very challenging. First, in IEEE 802.11 networks, bandwidth allocation is no longer a purely local operation, since competition for bandwidth occurs across multiple nodes. Therefore, it may be necessary to coordinate bandwidth estimates across competing nodes or, at a minimum, expose bandwidth usage patterns of the competing nodes to the estimating node. Second, as new flows are added into the network, the competition for bandwidth between nodes intensifies and bandwidth allocations may change, which essentially makes it difficult to estimate achievable bandwidth based on channel observation. Third, the recent introduction of different priority

classes in IEEE 802.11e [12] has further complicated the problem of achievable bandwidth estimation. Contrary to wired networks, where high priority packets are guaranteed to be sent before low priority packets, IEEE 802.11e does not provide such guarantees. Instead, IEEE 802.11e adjusts contention window sizes or inter-frame spacing to provide statistically differentiated services between flows with different priorities. Because of this statistical nature of IEEE 802.11e's service differentiation, even when the bandwidth is saturated with high priority traffic, some low priority traffic can still push its way through. Therefore, the bandwidth achievable by a flow of one priority is dependent on all competing flows from all priorities.

Current achievable bandwidth prediction algorithms take one of three approaches. The first approach, such as VMAC [4] and SWAN [2], uses a *free bandwidth model* where idle channel time is used to estimate achievable bandwidth at a node. While this approach does well in lightly-loaded networks, it does not capture the competition between nodes. In the second approach [15], [7], [9], [14], a *delay model* is used where the channel access time of a node's current traffic is used to predict the achievable bandwidth at the node. This model does not consider the fact that as more flows are added into the network, the competition for bandwidth intensifies and the channel access times change. Therefore, the channel access time before a node starts transmission does not reflect the achievable bandwidth after the node starts transmission. The third approach [5], [6], [3], [11], [13], [19] uses a *saturation model* where the competition between nodes is modeled assuming the network is fully saturated. However, this model does not capture the behavior of the network in an unsaturated or semi-saturated (i.e., some nodes of the network are saturated, some are not) states and therefore has limited applicability.

Because of the limitations of existing methods, our research focuses on designing a new achievable bandwidth prediction method based on the knowledge of the channel contentions between nodes. There are three major contributions of our work. First, we develop a bandwidth allocation model that captures the state of a single-hop network in an unsaturated, semi-saturated or saturated state. Despite the fact that our bandwidth allocation model is based on the complex relationship between competing nodes with multiple priority levels, our model is simple to calculate and hence can be used by individual nodes without imposing heavy computation overhead. Second, we show that given the priority levels of existing flows and their traffic demands, our model can be used to accurately predict the achievable bandwidth of a new flow in single hop IEEE 802.11 wireless networks under all network states. In comparison, the other techniques tend to only provide good estimates when the network is in the state at which they are

targeted. Third, although our single-hop bandwidth allocation model may not model a multihop wireless network accurately, simulation results show that it still provides better estimation than existing solutions regardless of the state of the network.

The remainder of this paper is organized as follows. Section II reviews the relevant components of IEEE 802.11 and its extensions for supporting service differentiation. Section III introduces our model of bandwidth allocation. Section IV shows how to use our model to predict achievable bandwidth in a single-hop network. Section V analyzes the three existing prediction methods using our bandwidth allocation model. Section VI evaluates our prediction method in comparison to the three existing methods through simulation in single hop networks. Section VII discusses and evaluates the extension of our model for achievable bandwidth prediction in multihop networks. Finally, Section VIII concludes and discusses future research.

II. IEEE 802.11 Protocol

The IEEE 802.11 standard provides two functions in the MAC sublayer: the distributed coordination function (DCF) and the point coordination function (PCF). Since PCF requires a Point Coordinator in the Access Point to poll other nodes to coordinate every packet transmission, its requires complex design of the access points and may also reduce the capacity of the network due to the overhead of polling. Because of this reason, in practice, PCF is not widely used. Hence, in this paper, we only examine bandwidth estimation for DCF and extensions of DCF.

A. DCF Mode

In IEEE 802.11 DCF mode, the transmission of each unicast packet invokes an RTS-CTS-DATA-ACK or DATA-ACK handshake between the sender and the receiver. A node desiring to transfer a data packet first invokes the carrier-sense mechanism to determine the busy/idle state of the medium based on the *Carrier-sensing Threshold* or an RTS or CTS packet indicating active communication in its neighborhood. If the medium is idle, the node defers a *DCF interframe space* (DIFS). If the medium stays idle during this DIFS period, the node may transmit its RTS packet. If the medium is busy, the node waits until the medium is determined to be idle for DIFS time units if the last detected frame was received correctly or *extended interframe space* (EIFS) time units if the last detected frame was not received correctly. After this DIFS or EIFS idle time, the node defers for an additional backoff period before transmitting an RTS. If the backoff timer is not yet set, the backoff period is generated as: $BackoffTime = Random() \times$

aSlotTime, where *Random()* is a pseudo random number uniformly distributed between 0 and *contention window (CW)* and *aSlotTime* is a very small time period (20 μs in the IEEE 802.11b standard). The backoff time is decremented by *aSlotTime* if the channel is idle during this period and stopped when a transmission is detected on the channel. The backoff timer is reactivated when the channel is sensed idle again for more than DIFS time units. The node transmits when the backoff timer reaches zero. For the first transmission attempt of each packet, CW is set to *minimum contention window* (CW^{min}) . After each failed transmission attempt, CW is doubled until it reaches *maximum contention window* (CW^{max}). At the end of every successful transmission, the CW value reverts to CW^{min} and a backoff procedure is performed immediately, even if no additional transmissions are currently queued.

The notations of the entire paper can be find at Appendix D.

B. Service Differentiation

In recent years, several methods [1], [12] have been proposed to provide service differentiation in IEEE 802.11 networks by adjusting the size of the backoff window to reflect the priority of packets. In these approaches, packets from different priorities are put into different queues in a node. Each queue independently observes and contends for the channel. Higher priority traffic has a statistically smaller average contention window size than lower priority traffic. Therefore, higher priority traffic has a greater chance to win the channel when competing with lower priority traffic. One method to differentiate average contention window sizes is to use different CW^{min}s (i.e., higher priority queues have smaller CW^{min}s).

Another method is to use different ratios to increase CW after a collision [1], [12]. The new value of CW is determined by $CW = PF \times CW + 1$, where PF is chosen according to the priority of the packet. Generally, lower priority traffic has a larger PF. However, this service differentiation method is unstable in its differentiation effects [1].

Due to the limitations of using different ratios to increase CW, in this paper, our model only focuses on supporting achievable bandwidth estimation under the first differentiation scheme where the priorities of flows are realized through different CW^{min}s. In Section III, we present our model that captures bandwidth allocations under the first differentiation scheme.

III. Bandwidth Allocation Model for Single-hop Networks

To effectively support achievable bandwidth precition, it is neccessary to model bandwidth allocation. Hence, in this section, we present our bandwidth allocation model for single-hop wireless networks.

Contrary to other analytical models that only examine networks under saturated conditions [3], [5], [6], [11], our model captures saturated, unsaturated and semi-saturated networks. Because of our model's ability to capture multiple network states, it can be used to estimate the achievable bandwidth of a flow in single-hop networks very accurately as shown in Section IV. In addition, this ability to capture multiple network states also ensures that our model can be used as a heuristic to predict achievable bandwidth in multihop networks and still outperform all existing methods (See Section VII). In the remainder of this section, we introduce our channel model and then investigate the relationship between bandwidth allocation and the states of nodes and networks.

A. Channel Model

Our model of channel includes two parts. We first show how we can map a single real node carrying multi-priority flows to multiple virtual nodes each carrying flows with a single priority. With this mapping, we then focus on modeling the backoff process at a (virtual or real) node that only carries single-priority flows so that we can understand how nodes compete for channel bandwidth.

1) Model of Multi-Priority Flows: Since our goal is to support multiple priorities of flows, each node is assumed to have Q queues, one for each priority class. While some advanced scheduling algorithm can be used to determine the channel access patterns of the traffic in each class, the interactions between scheduling and contention resolution algorithms is complex and out of the scope of this paper. Therefore, we assume a simple design similar to IEEE 802.11e [12], where there is no higher level scheduling algorithm and each queue accesses the channel independently as shown in Figure 1. Given this assumption, the channel access behavior of a node with multiple priority queues can be approximated by Q "virtual" nodes, each accessing the channel independently. Each "virtual" node only carries traffic of a specific class, as depicted in Figure 1. Therefore, in our model, we assume that one node (which may be a "virtual node") only transmits one class of traffic. The bandwidth allocation to a real node with multiple queues is approximately the sum of the bandwidth allocated to all of its "virtual nodes". This approximation omits the fact that virtual collisions between the queues of the same real node are internally resolved inside the real node and do not result in real collisions in the channel. Hence, no channel capacity is lost due to virtual collisions. By approximating queues using virtual nodes, we essentially omit the differences between virtual collisions and real collisions. This approximation, however, is reasonable since the number of queues in a real node is usually small so that the probability of virtual collisions is relatively small. Omiting the impact of virtual collisions does not significantly affect the accuracy of bandwidth prediction.

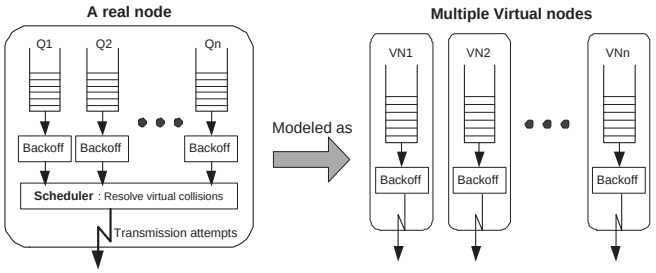


Fig. 1. Using virtual nodes to model multiple queues

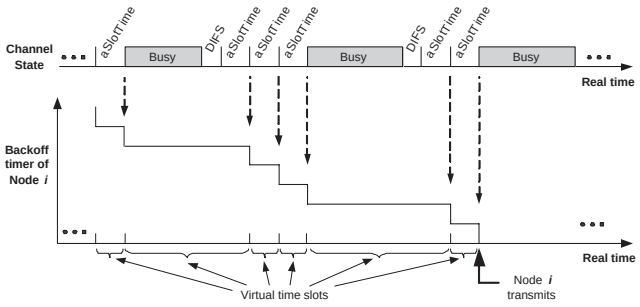


Fig. 2. Virtual time slots

2) Model of Backoff Process: Since the competition for bandwidth between nodes in IEEE 802.11 is determined by the backoff process at each node, to build our bandwidth allocation model, it is important to model this backoff process. For a fixed set $\mathcal{N} = \{1, 2 \ldots n\}$ of active nodes that are within contention range of each other, the backoff process at each node works as follows. When the channel turns from busy to idle, all nodes with backlogged packets start counting down their backoff timers one tick per $aSlotTime$ period until one node's backoff timer reaches 0. This node starts to transmit and all other nodes pause their backoff timers and wait for the channel to be idle again.

In Bianchi's model [5] of IEEE 802.11, it has been shown that this backoff process essentially divides real time into *virtual time slots*, where a node decrements its backoff timer once per virtual time slot. An example of such virtual time slots is shown in Figure 2, which depicts the channel state and Node i's backoff timer. There are two types of virtual time slots. When the channel is idle, a virtual time slot equals $aSlotTime$ (e.g., Node i's first virtual time slot). When the channel is busy, a virtual time slot includes a busy period, a DIFS period and an $aSlotTime$ period (e.g., Node i's second virtual time slot). Similar to real time slots, at most one packet can be sent in a virtual time slot. If multiple nodes attempt to send in the same virtual time slot, a collision happens.

Based on these virtual time slots, the backoff process at any Node i can be modeled as a discrete Markov process as shown in Figure 3. This Markove model enables us to formulate the competition for bandwidth between nodes and eventually calculate the bandwidth allocation between nodes. The Markove process is built using a method that is similar to Bianchi's model [5], except that we introduce service differentiation and unsaturated nodes in the modeling. In the Markov process, W_i represents the CW^{min} of Node i. λ_i is the probability of packet arrival at Node i during a virtual time slot, P_B is the channel busy probability and Q_i is the probability that Node i's queue is not empty when it finishes a successful packet transmission. State $\{s, b\}$ represents that the remaining backoff time at Node i is b virtual time slots and the contention window size at Node i is $2^s W_i$. A transition from $\{s, b+1\}$ to $\{s, b\}$ captures

the decrement of the backoff timer for every virtual time slot. At state $\{s, 0\}$, Node i transmits on the channel. With probability ϕ_i, the transmission fails and the new initial backoff time is uniformly chosen in the range $(0, \min(2^{s+1}W_i, 2^m W_i))$, where m is the number of collisions that are needed for the contention window size to reach CW^{max}. With probability $1 - \phi_i$, the transmission is successful and, depending on whether there are additional packets in the queue of Node i, Node i transits to either state III or state I. The transition to state III represents that when Node i finishes its current transmission, it has additional packets in its queue so that it will immediately start a backoff process for the next transmission. The transition to state I represents that when Node i finishes its current transmission, it has no more packets in its queue. In such cases, Node i should still perform a backoff process, which is captured in states $\{E, b\}$, where b is the remaining backoff time at Node i and E represents the empty state of Node i's queue. When the state of Node i reaches $\{E, 0\}$, since there are no packets in its queue, Node i does not transmit on the channel and remains in state $\{E, 0\}$ until a packet arrival. If the packet arrives during the busy period of the channel, Node i starts a backoff process before it transmits the packet (transition from state $\{E, 0\}$ to state III). Otherwise, Node i transmits immediately the packet (transition from state $\{E, 0\}$ to state $\{0, 0\}$).

Based on the Markov model of the backoff process, next, we can examine the different states and channel access behavior of nodes in the remainder of Section III.

B. States of Nodes

Independent of the state of the network, a node's bandwidth share is related to its traffic load and its contention window size. To understand this relationship, nodes in a wireless network are classified into two types: saturated nodes and unsaturated nodes. A *saturated node* always has backlogged packets ($Q_i = 1$), while an *unsaturated node* often has an empty queue ($Q_i < 1$). By examining the relationship between bandwidth allocation and node states, we show that the bandwidth share of a node depends on the states of all competing nodes in the network.

To analyze the relationship between bandwidth allocation and node states, let S_i be the fraction of channel bandwidth allocated to Node $i \in \mathcal{N}$ and P_i be the probability that Node i successfully transmits a packet in a virtual time slot. Subscript sat and $\overline{sat}$ are used to indicate saturated and unsaturated nodes respectively. For example, $S_{i,\overline{sat}}$ represents Node i's bandwidth when Node i is an unsaturated node. Since bandwidth prediction should be based on long term network state rather than transient network behavior, short term variations, such as the instantaneous bandwidth allocation to a node right after a

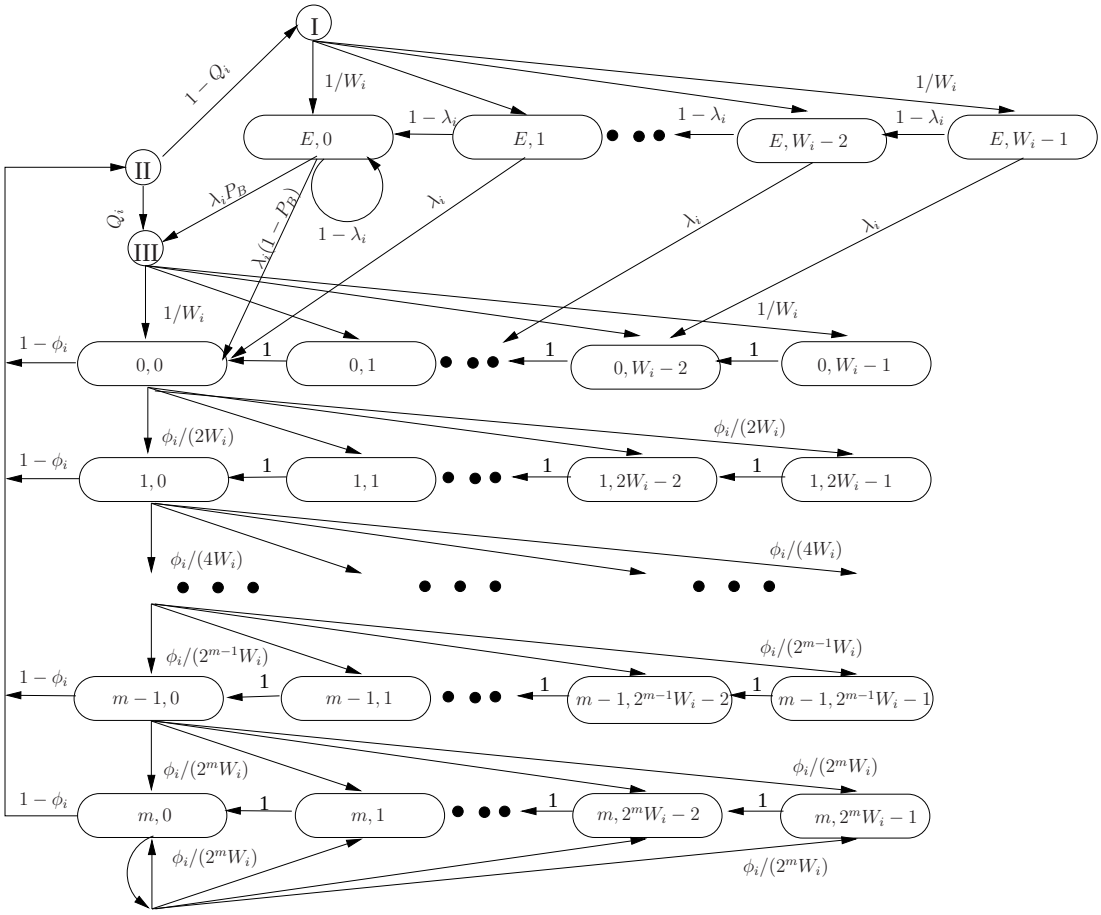


Fig. 3. Markov chain of the backoff process at Node i.

collision, should not be considered in achievable bandwidth prediction. Therefore, throughout this paper, our analysis focuses on the steady state of the network so that all variables in the paper represent average values. In addition, we assume that the traffic characteristics of flows are stable enough so that a moving average algorithm is able to extract the time average of the traffic characteristics such as the average duration of a successful data transmission (a full RTS/CTS/DATA/ACK handshake) at Node i, L_i, and the average packet arrival rate at Node i, R_i. We also assume that the queue length in each node is large enough to absorb the short-term variations in traffic arrival rate. Therefore, as long as the long-term average packet arrival rate extracted by the moving average is smaller than the packet transmission rate at a node, the packet loss caused by queue overflow can be omitted. Finally, we assume that traffic sources at each node are independent of each other.

The bandwidth allocated to Node i is related to P_i, which is determined by two probabilities: τ_i and ϕ_i. τ_i is the probability that Node i transmits in a randomly chosen virtual time slot and ϕ_i is the probability that a collision happens when Node i attempts to transmit a packet. In the following theorem, we show the relationship between S_i, P_i , τ_i and ϕ_i.

Theorem 1: For any node state,

1)
$$\frac{S_i}{\sum_{j=1}^{n} S_j} = \frac{P_i L_i}{\sum_{j=1}^{n} P_j L_j}, \tag{1}$$

$$P_i = \frac{\tau_i}{1-\tau_i} \prod_{j=1}^{n} (1-\tau_j), \tag{2}$$

$$\phi_i = 1 - \frac{\prod_{j=1}^{n}(1-\tau_j)}{1-\tau_i}. \tag{3}$$

2) Assume C is the maximum utilization of the channel, which is the maximum fraction of channel bandwidth that can be used for successful RTS/CTS/DATA/ACK handshakes. Then, S_i satisfies:
$$\sum_{i=1}^{n} S_i \leq C. \tag{4}$$

Proof: Assuming that there are x number of virtual time slots in a time unit, the expected number of packets that a Node i sends is xP_i and the expected amount of bandwidth used by Node i is $S_i = xP_iL_i$. Hence, $S_i / \sum_{i=1}^{n} S_j = xP_iL_i / (\sum_{i=1}^{n} xP_jL_j)$. Canceling out x results in Equation (1).

When multiple nodes try to transmit in the same virtual time slot, a collision happens and all transmissions fail. Therefore, the probability that a successful transmission of Node i happens in a virtual time slot equals the probability that Node i transmits in the slot and is the only node that transmits in that slot. Therefore,
$$P_i = \tau_i \prod_{j \neq i, j \in \mathcal{N}} (1-\tau_j) = \frac{\tau_i \prod_{j=1}^{n}(1-\tau_j)}{1-\tau_i}. \tag{5}$$

Similarly, the probability that when Node i transmits in a slot, the transmission collides with some other node's transmission can be expressed as:
$$\phi_i = 1 - \prod_{j \neq i, j \in \mathcal{N}} (1-\tau_j) = 1 - \frac{\prod_{j=1}^{n}(1-\tau_j)}{1-\tau_i}. \tag{6}$$

■

If the state of a node is saturated, besides Theorem 1, the following additional relationships among S_i, P_i, τ_i and ϕ_i hold.

Theorem 2: For a saturated Node i,

1) the probability that Node i transmits in a randomly chosen virtual time slot is:
$$\tau_{i,sat} = \frac{2(1-2\phi_i)}{(1-2\phi_i)(W_i+2) + \phi_i(W_i+1)(1-(2\phi_i)^{m_i})}. \tag{7}$$

2) The maximum bandwidth allocation of Node i equals $S_{i,sat}$ and

$$S_{i,sat} = \frac{P_{i,sat} L_i \sum_{j=1}^{n} S_j}{\sum_{j=1}^{n} P_j L_j}. \tag{8}$$

3) Node i is a saturated node if and only if its load is larger than its maximum bandwidth allocation:

$$S_{i,sat} < R_i L_i, \tag{9}$$

where R_i is the average packet arrival rate at node i.

Proof:

1) The proof for Equation (7) is similar to the one presented in Bianchi's model [5] and can be found in Appendix A.
2) Solving for P_i in Equation (1) results in Equation (8). Since there are always packets in Node i's queue, Node i is always competing for the channel. Therefore, $S_{i,sat}$ is the maximum bandwidth allocation that Node i can achieve.
3) If Node i is saturated (i.e., it always has packets in its queue), the average packet arrival rate at Node i must be larger than its maximum bandwidth allocation. Otherwise, the queue in Node i would become empty at some time. On the other hand, if the average packet arrival rate is larger than Node i's bandwidth allocation, according to queuing theory, eventually the queue length in Node $i \to \infty$. Therefore, Node i is saturated.

■

It is worth noting that the physical meaning of the expression $\frac{\sum_{j=1}^{n} S_j}{\sum_{j=1}^{n} P_j L_j}$ in Equation (8) is the reciprocal of the average duration of a virtual time slot. Therefore, $P_{i,sat}\frac{\sum_{j=1}^{n} S_j}{\sum_{j=1}^{n} P_j L_j}$ in Equation (8) essentially represents Node i's actual transmission rate in terms of packets per second on the channel. Hence, for unsaturated nodes, $P_{i,\overline{sat}}\frac{\sum_{j=1}^{n} S_j}{\sum_{j=1}^{n} P_j L_j}$ should equal the load on Node i, which is confirmed by Equation (12) of Theorem 3.

Theorem 3: For any unsaturated Node i,

$$S_{i,\overline{sat}} = R_i L_i, \tag{10}$$

$$S_{i,\overline{sat}} \leq S_{i,sat}, \tag{11}$$

$$P_{i,\overline{sat}} = \frac{R_i \sum_{j=1}^{n} P_j L_j}{\sum_{j=1}^{n} S_j}, \tag{12}$$

$$P_{i,\overline{sat}} < P_{i,sat}, \tag{13}$$

Proof:

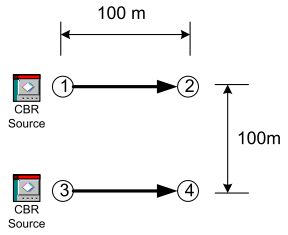


Fig. 4. A two-flow network that may be in different network states.

1) When Node i finishes transmitting, it often has no more packets in its queue to transmit. Based on a similar analysis to Theorem 2, this happens when the total amount of traffic that Node i needs to transmit is smaller than its maximum bandwidth $S_{i,sat}$ (Inequality (11)). In this case, the node's throughput is the same as its load (Equation (10)).
2) Combining Equation (1) and Equation (10) results in Equation (12).
3) Because an unsaturated node often has no packets to transmit during idle periods, it is obvious that the probability that it transmits in a virtual time slot is lower than the case when the node always has backlogged packets and is always trying to transmit (Inequality (13)).

■

As can be seen from Equations (8) and (10), the bandwidth allocated to a node depends on both its own state and the bandwidth allocations of the other nodes, which in turn is related to the states of the other nodes. Therefore, the bandwidth allocated to a node is related to the congestion level of the whole network.

C. States of Networks

Depending on the congestion level of the network, an IEEE 802.11 network can be in one of three states: saturated, unsaturated or semi-saturated. A network is in a *saturated state* when every node is saturated, which usually means that the network is overloaded. In an *unsaturated network*, no node is saturated, which indicates a lightly loaded network. A *semi-saturated* network is between the saturated state and the unsaturated state, where some of the nodes are saturated while other nodes are unsaturated.

To demonstrate that the bandwidth allocated to a node is related to the network state, a simple ns-2 [8] simulation is performed. In the 150-second simulation, there are two CBR flows, $1 \rightarrow 2$ and $3 \rightarrow 4$ (see Figure 4), with the same priority competing for a 2Mbps channel. The queue size at each node is 50 packets.

Figures 5 and 6 depict the queue length and the throughput of Nodes 1 and 3, respectively. From 5 to 50 seconds, the CBR sources in Nodes 1 and 3 each generate 50 512Byte packets per second. The queues

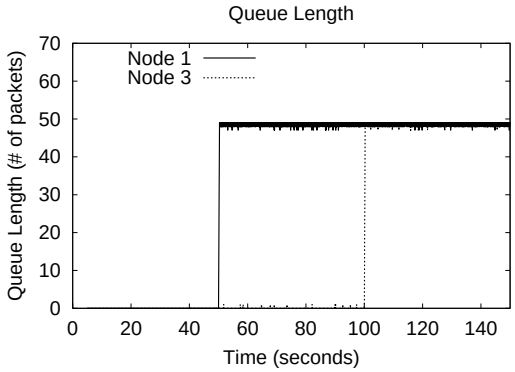


Fig. 5. Queue length of Nodes 1 and 3

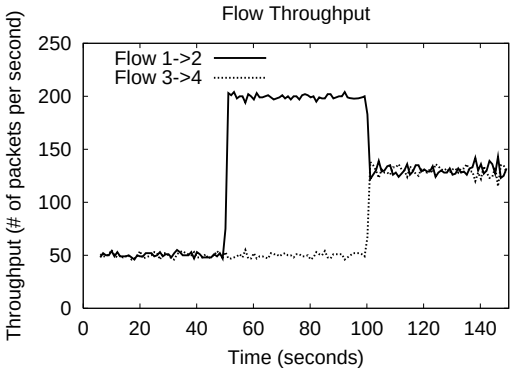


Fig. 6. Throughput of Nodes 1 and 3

in both nodes are often empty during this period, indicating an unsaturated network. Both flows achieve throughputs that match their packet generation rates. At 50 seconds, the CBR source in Node 1 increases its rate to 300 packets per second. The queue in Node 1 becomes full while the queue in Node 3 is still often empty, indicating a semi-saturated network. During this period, even though Node 1 tries to send more packets, it is not able to "push down" Node 3's bandwidth. At 100 seconds, the CBR source in Node 3 also increases its rate to 300 packets per second. Both queues in Nodes 1 and 3 become constantly full, indicating a saturated network. During this period, Nodes 1 and 3 share the channel bandwidth equally.

This example shows that the achievable bandwidth in the presence of competing nodes is dependent on the state of the network. In general, a practical network can be in any of the three states depending on the traffic load. Therefore, an effective achievable bandwidth prediction model must capture bandwidth allocation in all network states.

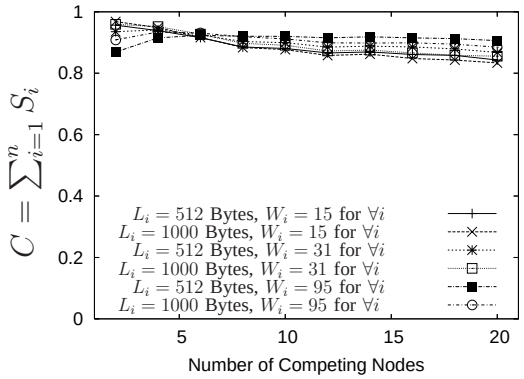
D. Bandwidth Allocation for Different Networks States

Due to the relationship between bandwidth allocation and network states, in this section, we analyze the bandwidth allocation for each of the three network states: saturated, unsaturated and semi-saturated. Our analysis reveals four very simple mathematical equations (Equations (38), (39), (40) and (41)), which can be used to calculate the bandwidth allocated to each node under all network states.

1) Saturated Network: In a saturated network with n transmitting nodes, every node always has packets to transmit and hence fills up the network bandwidth. Therefore,

$$\sum_{i=1}^{n} S_i = C, \tag{14}$$

where C is the maximum channel utilization, which is the maximum fraction of channel time that is able to be used for successful data transmission (successful RTS/CTS/DATA/ACK handshake). Although



C is related to the number of competing nodes, n, and the L_i and W_i of these nodes, C's value is not very sensitive to n, L_i and W_i and can be roughly treated as a constant under a reasonable number of competing flows. Figure III-D.1 shows the value of C under vastly different values of n, L_i and W_i using ns-2 [8] simulations. By choosing a good approximation value for C , such as 0.9, the approximation error is less than 7.5%.

Given the approximated constant C, if $\tau_{i,sat}$ of each node is known, it is easy to solve S_i for every node using Theorems 1, 2 and Equation (14) since in a saturated network the state of all nodes is saturated. However, although it is possible to calculate $\tau_{i,sat}$ and ϕ_i based on Equations (3) and (7), there is no closed form solution even in the simplest case where $\tau_{i,sat}$ and ϕ_i are the same for all nodes as shown in Bianchi's model [5]. The solution is especially hard to calculate for a MAC layer that supports multiple classes, since in such a network, $\tau_{i,sat}$ and ϕ_i differ from node to node due to the different W_is at nodes. Numerical analysis must be used to get the exact solution, which has high computational complexity and is not practical for use since the flows in a real network may change quickly due to the arrival and departure of flows and the mobility of nodes. Therefore, we use a simple model with low computational overhead to approximate the exact model. This approximation model does not require calculations of ϕ_i and $\tau_{i,sat}$.

The relationship between ϕ_i and $\tau_{i,sat}$ can be determined from Equation (3):

$$(1-\phi_i)(1-\tau_{i,sat}) = \prod_{j=1}^{n}(1-\tau_{j,sat}), \tag{15}$$

which indicates that for Nodes i and j, the following relationship holds:

$$(1-\phi_i)(1-\tau_{i,sat}) = (1-\phi_j)(1-\tau_{j,sat}). \tag{16}$$

Solving ϕ_i from Equation (16) results in:

$$\phi_i = \phi_j + \frac{\tau_{i,sat} - \tau_{j,sat}}{1 - \tau_{i,sat}}\phi_j \quad (17)$$

$$\approx \phi_j = \phi \text{ for some constant } \phi. \quad (18)$$

The approximation in Equation (18) is valid since the approximation error $\frac{\tau_{i,sat}-\tau_{j,sat}}{1-\tau_{i,sat}}$ is very small. This is because in [10] it has been proved that $\phi_i < 1/2$. Combining this with Equation (7) results in $\tau_{i,sat} < \frac{2}{(2+W_i)}$. In most real IEEE 802.11 networks, $W_i \gg 1$ (e.g., $W_i = 31$ in IEEE 802.11b) so that $\tau_{i,sat} \ll 1$. Hence, $|\frac{\tau_{i,sat}-\tau_{j,sat}}{1-\tau_{i,sat}}| < \frac{\max(\tau_{i,sat},\tau_{j,sat})}{1-\tau_{i,sat}} \ll 1$ (e.g., in IEEE 802.11b, $\frac{\tau_{i,sat}}{1-\tau_{i,sat}} < 6.45\%$).

To derive the relationship between $S_{i,sat}$ and $S_{j,sat}$, it is necessary to determine the relationship between $P_{i,sat}$ and $P_{j,sat}$. From Equation (2),

$$\frac{P_i}{P_j} = \frac{\tau_i(1-\tau_j)}{\tau_j(1-\tau_i)}. \quad (19)$$

Since $W_i \gg 1, W_j \gg 1$, combining Equations (7), (18) and (19),

$$\frac{P_{i,sat}}{P_{j,sat}} = \frac{(1-2\phi)W_j + \phi W_j(1-(2\phi)^{m_j})}{(1-2\phi)W_i + \phi W_i(1-(2\phi)^{m_i})}. \quad (20)$$

However, this relationship is still quite complex and depends on ϕ, which is difficult to calculate. Therefore, it is beneficial to find a simpler approximation. Note that if $m_i \approx m_j$, Equation (20) can be further approximated as:

$$\frac{P_{i,sat}}{P_{j,sat}} \approx \frac{W_j}{W_i}. \quad (21)$$

Combining this approximation with Equations (8) and (14), and considering that every node in the network is saturated, S_i in a saturated network can be solved as:

$$S_i = S_{i,sat} = \frac{P_{i,sat}L_i}{\sum_{j=1}^{n} P_{j,sat}L_j}\sum_{j=1}^{n} S_j \approx \frac{\frac{1}{W_i}L_i}{\sum_{j=1}^{n}\frac{1}{W_j}L_j}C. \quad (22)$$

Since the above approximation of S_i does not depend on ϕ_i and $\tau_{i,sat}$, numerical analysis is not needed.

2) Unsaturated Network: Since every node in an unsaturated network is unsaturated, from Equations (10) and (12),

$$P_i = P_{i,\overline{sat}} = \frac{R_i\sum_{j=1}^{n} P_{j,\overline{sat}}L_j}{\sum_{j=1}^{n} R_jL_j}, \quad (23)$$

$$S_i = S_{i,\overline{sat}} = R_iL_i. \quad (24)$$

3) Semi-saturated Network: In a semi-saturated network, where the set of saturated nodes is N_1, the set of unsaturated nodes is N_2, $N_1 \cup N_2 = \mathcal{N}$. Since the saturated nodes in the network always have packets for transmission and hence fill up the channel bandwidth,

$$\sum_{i=1}^{n} S_i \approx C. \tag{25}$$

In a semi-saturated network, there are both saturated and unsaturated nodes. Therefore, to solve the bandwidth allocation S_i of Node i, it is necessary to determine the state of Node i. Theorems 2 and 3 show that S_i is determined by $S_{i,sat}$ and Node i's load R_iL_i. If $S_{i,sat}$ is larger than R_iL_i, Node i is unsaturated and S_i equals R_iL_i. If $S_{i,sat}$ is smaller than R_iL_i, Node i is saturated and S_i becomes $S_{i,sat}$. Therefore, as long as $S_{i,sat}$ and R_iL_i are known, S_i can be easily determined as:

$$S_i = \min(R_iL_i, S_{i,sat}). \tag{26}$$

To solve $S_{i,sat}$, combining Equations (8) and (25) results in:

$$S_{i,sat} = \frac{P_{i,sat}L_i}{\sum_{j=1}^{n} P_jL_j} C = \frac{P_{i,sat}L_i}{\rho} C, \tag{27}$$

where $\rho = \sum_{j=1}^{n} P_jL_j$ is the average number of bits transmitted in a virtual time slot. Essentially, Equation (27) shows that $S_{i,sat}$ depends on $P_{i,sat}$ and the congestion level on the channel, which is captured by ρ.

Clearly, it is necessary to determine both $P_{i,sat}$ and ρ to solve $S_{i,sat}$. From Equations (2),

$$P_{i,sat} = \frac{\tau_{i,sat} \prod_{j=1}^{n}(1-\tau_j)}{1-\tau_{i,sat}}. \tag{28}$$

Therefore, ρ can be expressed as:

$$\begin{aligned}
\rho &= \sum_{i\in N_1} P_{i,sat}L_i + \sum_{i\in N_2} P_{i,\overline{sat}}L_i \\
&\text{(Using Equation (12))} \\
&= \sum_{i\in N_1} P_{i,sat}L_i + \sum_{i\in N_2} \frac{R_iL_i \sum_{j=1}^{n} P_jL_j}{C} \\
&\text{(Using the definition of } \rho\text{)} \\
&= \sum_{i\in N_1} P_{i,sat}L_i + \sum_{i\in N_2} \frac{R_iL_i\rho}{C}.
\end{aligned}$$

Solving for ρ results in:

$$\rho = \frac{\sum_{i\in N_1} P_{i,sat}L_i}{1-\sum_{i\in N_2}\frac{R_iL_i}{C}} \quad (29)$$

(Using Equation (28))

$$= \frac{\sum_{i\in N_1}\frac{\tau_{i,sat}}{1-\tau_{i,sat}}L_i\prod_{j=1}^{n}(1-\tau_j)}{1-\sum_{i\in N_2}\frac{R_iL_i}{C}}. \quad (30)$$

Equations (28) and (30) show that both $P_{i,sat}$ and ρ depend on τ, which has no easy solution. However, it is possible to extract the parts of ρ and $P_{i,sat}$ that do not depend on τ and use them to approximate $S_{i,sat}$.

First, we simplify the $\frac{\tau_{i,sat}}{1-\tau_{i,sat}}$ part in Equations (28) and (30). From Equation (7),

$$\frac{1-\tau_{i,sat}}{\tau_{i,sat}} = \frac{(1-2\phi_i)W_i+\phi_i(W_i+1)(1-(2\phi_i)^{m_i})}{2(1-2\phi_i)}.$$

Since $W_i \gg 1$ and $m_i \approx m_j = m$, the above equation can be approximated as:

$$\frac{1-\tau_{i,sat}}{\tau_{i,sat}} \approx \frac{1}{2}(1+\phi_i\frac{1-(2\phi_i)^m}{1-2\phi_i})W_i. \quad (31)$$

Similar to Equation (18) for saturated networks, for semi-saturated networks, the following relationship still holds:

$$\phi_i = \phi_j + \frac{\tau_i-\tau_j}{1-\tau_i}\phi_j \approx \phi_j = \phi \text{ for some constant } \phi. \quad (32)$$

This is because in semisaturated networks, for any Node i, $\tau_i \le \tau_{i,sat}$ (If Node i is unsaturated, $\tau_i < \tau_{i,sat}$. If Node i is saturated, $\tau_i = \tau_{i,sat}$.) Therefore, $|\frac{\tau_i-\tau_j}{1-\tau_i}| < \frac{\max(\tau_{i,sat},\tau_{j,sat})}{1-\tau_{i,sat}}$. Since $\frac{\max(\tau_{i,sat},\tau_{j,sat})}{1-\tau_{i,sat}} \ll 1$ as discussed in Section III-D.1, $|\frac{\tau_i-\tau_j}{1-\tau_i}| \ll 1$. For example, for two Nodes i and j with dramatically different contention window allocations of $W_i = 40$ and $W_j = 4000$, approximating ϕ_i to ϕ_j only gives an error $\frac{\tau_i-\tau_j}{1-\tau_i} < 5\%$. Therefore, Equation (32) is true. Using Equation (32), Equation (31) can be approximated as:

$$\frac{1-\tau_{i,sat}}{\tau_{i,sat}} \approx \frac{1}{2}(1+\phi\frac{1-(2\phi)^m}{1-2\phi})W_i = f(m,\phi)W_i, \quad (33)$$

where $f(m,\phi) = \frac{1}{2}(1+\phi\frac{1-(2\phi)^m}{1-2\phi})$. The impact of the approximation in Equation (32) to Equation (33) is not significant. Consider the example where we show that the approximation error in Equation (32) is less than 5% for two nodes with $W_i = 40$ and $W_j = 4000$. In this case, assuming that m is 5 (IEEE 802.11b configuration) and the collision probability is at its upper bound $\phi = 0.5$, the approximation error in Equation (33) is no larger than 11%. Since in unsaturated networks, ϕ is usually much smaller than

its upper bound 0.5, the approximation error in Equation (33) can be even smaller in reality. (e.g., for $\phi = 0.3$, approximation error is 4.3%.)

Applying the approximation in Equation (33) to Equation (30) results in:

$$\rho \approx \frac{\sum_{i\in N_1} \frac{L_i}{W_i} \prod_{j=1}^{n}(1-\tau_j)}{(1-\sum_{i\in N_2} \frac{R_i L_i}{C}) f(m,\phi)} \tag{34}$$

$$= \eta\alpha, \tag{35}$$

where $\eta = \frac{\sum_{i\in N_1} \frac{L_i}{W_i}}{1-\sum_{i\in N_2} \frac{R_i L_i}{C}}$ and $\alpha = \frac{\prod_{j=1}^{n}(1-\tau_j)}{f(m,\phi)}$. Applying the approximation in Equation (33) to Equation (28) results in:

$$P_{i,sat} = \frac{\prod_{j=1}^{n}(1-\tau_j)}{f(m,\phi)W_i} = \frac{\alpha}{W_i}. \tag{36}$$

Note that η is not dependent on τ_i or ϕ and so can be easily calculated, while α is dependent on τ_i and ϕ and so requires numerical analysis to be calculated.

Combining Equations (35) and (36) with Equation (27), we finally get $S_{i,sat}$ as follows:

$$S_{i,sat} = \frac{L_i C}{\eta W_i}. \tag{37}$$

Note that $S_{i,sat}$ does not depend on α. Therefore, the need for numerical analysis has been eliminated.

Combing Equations (37) and (26), S_i becomes:

$$S_i = \min(R_i L_i, S_{i,sat}) = \min(R_i L_i, \frac{L_i C}{\eta W_i}). \tag{38}$$

Therefore, to determine S_i, we only need to determine the value of η of the network. To do this, note that the larger the η, the smaller the $S_{i,sat}$. When η is so large so that $R_i L_i > S_{i,sat}$, Node i becomes saturated and $S_i = S_{i,sat}$. When Node i is at the edge of turning from unsaturated to saturated, $R_i L_i = S_{i,sat} = \frac{L_i C}{\eta W_i}$. Therefore, the threshold value of η at this turning point, η_i^*, can be expressed as:

$$\eta_i^* = \frac{C}{R_i W_i}. \tag{39}$$

Sorting the nodes according to their η_i^* in ascending order results in a sequence of nodes $(x_1, x_2, \ldots, x_n)$ where $\eta_{x_i}^* \leq \eta_{x_j}^*$ if $i < j$. If $\eta_{x_k}^* < \eta < \eta_{x_{k+1}}^*$, nodes $x_1, \ldots, x_k$ are saturated and nodes $x_{k+1}, \ldots, x_n$ are unsaturated. Therefore,

$$\eta = \eta(k) = \frac{\sum_{i=1}^{k} \frac{L_{x_i}}{W_{x_i}}}{1-\sum_{i=k+1}^{n} \frac{R_{x_i} L_{x_i}}{C}}, \tag{40}$$

$$\eta_{x_k}^* \leq \eta(k) < \eta_{x_{k+1}}^*. \tag{41}$$

Since the range of k is the number of competing neighboring nodes, which is generally not large, we can calculate the value of η corresponding to each value of k using Equation (40). The value of η that satisfies the inequality constraint (41) is the solution to η and determines the corresponding solution for k. It can be shown that the solution (η, k) is unique (See Appendix B). With the unique solution (η, k), the state of the nodes can be uniquely decided, where the saturated nodes are $N_1 = \{x_1, x_2, ..., x_k\}$ and the unsaturated nodes are $N_2 = \{x_{k+1}, x_{k+2}, ..., x_n\}$. Hence, the bandwidth allocation to every node can also be determined using Equation (38).

Note that Equations (38), (40) and (41) can also be used to express the bandwidth allocation in both saturated and unsaturated networks. For saturated networks, we can simply define $\eta^*_{x_{n+1}} = \infty$. Since all nodes are saturated, $\eta > \eta^*_j$ for any Node j in the network. Therefore, the solutions to Equations (40) and (41) are obtained at $k = n$, where $\eta^*_{x_{n+1}} > \eta(n) = \sum_{i=1}^{n} \frac{L_{x_i}}{W_{x_i}} > \eta^*_{x_n}$. In this case, N_2 is empty and it is easy to check that Equation (38) is reduced to Equation (22). For a non-saturated network, there is no valid solution for $k > 0$. In such a case, we can let $k = 0$ and Equation (38) becomes Equation (24).

Figure 7 shows an example of the $\eta(k)$ in a five node network in saturated, unsaturated and semi-saturated states, respectively. The points in the figure represent the values of η corresponding to k calculated using Equation (40). The inequality constraint (41) is represented by the shaded area. When a point for η is located in the shaded area, the point represents a valid solution for η. In Figure 7, the solution for a saturated network is achieved when $k = 5$, the solution for a semi-saturated network is achieved when $k = 2$, and the unsaturated network has no valid solution for $k > 0$. Any value in the negative zone of η essentially means that the k value is too small. When k is too small, it implies that too many nodes are regarded as unsaturated nodes and their bandwidth requirements exceed the network capacity, resulting in a negative denorminator in Equation (40). These negative values of η are not in the feasible zone since they are ruled out by Inequality (41) as shown in Figure 7.

Since with the four simple mathematical equations (Equations (38), (39), (40) and (41)), the bandwidth allocated to each node under all network states can be calculated, we next show how to use these equations to predict the achievable bandwidth of a new flow by calculating the bandwidth allocation of the node that carries this new flow.

IV. Prediction of Achievable Bandwidth in Single-hop Networks

To predict the achievable bandwidth of a new flow at Node k in a single-hop network, Node k must know the load of all competing nodes to calculate η, which determines its bandwidth share (see

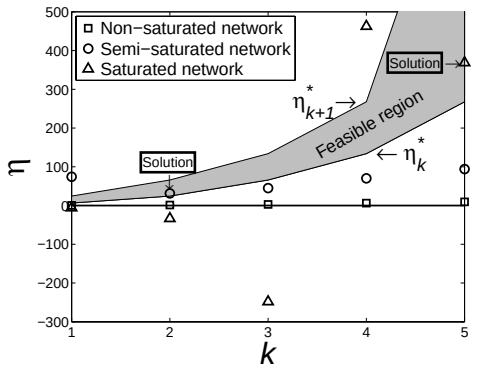


Fig. 7. Example of η, η^* and the corresponding solution

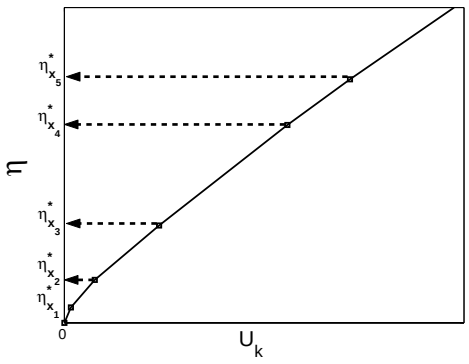


Fig. 8. Piece-wise linear relationship between η and U_k

Equations (38) and (40)). In a single-hop wireless network, such as a wireless LAN, load information can be exchanged between competing nodes through low-rate periodic broadcasting so that each node can make their own achievable bandwidth predictions. Although broadcasting neighbor information consumes network bandwidth and resources, the frequency of the broadcasts does not need to be very high. With a low broadcasting rate, the information exchanged between nodes represents the average of the traffic load rather than the burstiness of the traffic. Therefore, bandwidth estimation reflects long term achievable bandwidth, rather than instantaneously achievable throughput that may change quickly. This packet overhead can be further reduced by piggybacking load information onto control and data packets, adding minimal overhead to heavily loaded networks. Hence, in the remainder of the section, we assume that a node knows the traffic load information of its neighbors and focus on discussing the details of using our bandwidth allocation model to predict achievable bandwidth.

Note that Node k obtains its maximum bandwidth allocation if it is saturated after the new flow arrives. Hence, the achievable bandwidth of the new flow equals $S_{k,sat}$. The network state when Node k achieves $S_{k,sat}$ is either semi-saturated or saturated. Therefore, using Equation (40),

$$\eta = \frac{\frac{L_k}{W_k} + \sum_{i \in N_1} \frac{L_i}{W_i}}{1 - \sum_{i \in N_2} \frac{R_i L_i}{C}} = \frac{U_k + \sum_{i \in N_1} \frac{L_i}{W_i}}{1 - \sum_{i \in N_2} \frac{R_i L_i}{C}}, \tag{42}$$

where $U_k = \frac{L_k}{W_k}$ reflects the load that Node k adds on the network when it achieves $S_{k,sat}$. To build a mapping between η and U_k so that η can be estimated whenever a new flow arrives at Node k, it is necessary to first sort the nodes according to their saturation threshold η_i^*. Then, Equation (42) can be

rewritten as:

$$U_k = \begin{cases} \eta(1 - \sum_{i=k+1}^{n} \frac{R_{x_i} L_{x_i}}{C}) - \sum_{i=1}^{k} \frac{L_{x_i}}{W_{x_i}}, \text{ for } \eta^*_{x_k} \leq \eta < \eta^*_{x_{k+1}}, \\ \eta(1 - \sum_{i=1}^{n} \frac{R_{x_i} L_{x_i}}{C}), \text{ for } 0 \leq \eta < \eta^*_{x_1}, \\ \eta - \sum_{i=1}^{n} \frac{L_{x_i}}{W_{x_i}}, \text{ for } \eta^*_{x_n} \leq \eta. \end{cases} \tag{43}$$

Note that the above equation is a piece-wise linear function of U_k and η in the positive area of U_k. Using the collected load information of its neighbors, Node k is able to build such a piece-wise linear function. Given any U_k, Node k can immediately determine the corresponding η and use this η to find its bandwidth allocation $S_{k,sat}$ according to Equation (37). Hence, the achievable bandwidth for Node k can be predicted.

An example of the piece-wise linear function of U_k and η is shown in Figure 8, where Node k has five competing neighbors. The piece-wise function consists of five line segments, corresponding to the η_i^*s of the five competing neighbors. As U_k increases (i.e., through a smaller W_k or a larger L_k), the η in the network increases and passes the competing nodes' saturation threshold η_i^* one by one, indicating that the new flow pushes the competing nodes to their saturated state and gets an increasing amount of bandwidth. The pseudo code for building the piece-wise function and predicting achievable bandwidth is presented in Appendix C.

It is also worth noting that this piece-wise linear function also provides a method to support admission control since it essentially identifies the effects of Node k's traffic on the other nodes. Combined with a priority-based admission control policy, it can be decided at what U_k value Node k's traffic will push a higher priority node to a saturated state, so that admission decisions for Node k's traffic can be made. The use of our prediction model for admission control can be found in our work [17].

V. Other Bandwidth Prediction Approaches

In this section, we analyze the limitations of the three existing methods, the delay model, the free bandwidth model, and the saturation model, for estimating the achievable bandwidth of a node. The analysis shows that none of these three methods can capture the network under all network states, which is confirmed by our simulation results in Section VI.

A. Average Delay Model

In [2], [7], [9], [14], [15], the achievable bandwidth at Node i is predicted using the average packet transmission delay at the MAC layer, Δ_i, which is the period of time between when a packet is ready to be transmitted at the MAC layer and when the packet is successfully transmitted. The achievable bandwidth of Node i is approximated as $\frac{1}{\Delta_i}$, with some additional adjustment for different packet sizes. This heuristic seems appropriate since $1/\Delta_i$ represents the service rate of the network to the node. However, the analysis in this section shows that Δ_i is not independent of Node i's transmission rate. When Node i increases its transmission rate, its Δ_i increases. Therefore, $\Delta_i < \tilde{\Delta}_i$, where $\tilde{\Delta}_i$ is the packet transmission delay when Node i is saturated. Hence, using Δ_i under low transmission rates to estimate the achievable bandwidth, which is achieved when Node i is saturated, can be over-optimistic.

To show that Δ_i increases when Node i increases its transmission rate, note that Δ_i includes four parts: the time of Node i's backoff slots, the time consumed by successful transmissions from other nodes during Node i's backoff procedure, the time consumed by collisions and the time of Node i's own successful transmission. Therefore,

$$\Delta_i \approx B_i \times aSlotTime + B_i \sum_{j=1}^{n} P_j L_j + B_i P_c T_c + L_i, \tag{44}$$

where B_i is the average number of backoff slots before a transmission. T_c is the average duration of a collision, which usually equals a duration of RTS-CTS exchange since most of the collisions happen between RTS packets. P_c is the probability that multiple nodes transmit in a virtual time slot simultaneously and create a collision.

As Node i increases its transmission rate, the first three parts on the right side of the equal sign in Equation (44) increase due to the following three reasons. First, B_i increases as Node i becomes saturated. When Node i is not saturated, after it finishes a transmission and the backoff procedure following the transmission, the next packet may still not have arrived. When the next packet arrives and Node i sees an idle channel, Node i does not need to perform a backoff procedure before it starts a new transmission. In this case, $B_i = 0$. However, when Node i becomes saturated, it must always backoff before it starts a transmission. Therefore, B_i increases as Node i becomes saturated. Second, the probability that some other nodes successfully transmit their packets during Node i's backoff procedure also increases as Node

i becomes saturated. Assuming Node k is another non-saturated node in the network, from Equation (12),

$$P_k = \frac{R_k \sum_{j=1}^{n} P_j L_j}{(\sum_{j=1}^{n} S_j)} = \frac{R_k(\sum_{j=1, j\neq i}^{n} P_j L_j + P_i L_i)}{(\sum_{j=1}^{n} S_j)}. \tag{45}$$

As Node i becomes saturated, its P_i increases. Therefore, P_k increases as shown in Equation (45). Similarly, all other non-saturated nodes in the network also increase their probability of successfully transmitting a packet in a virtual time slot. Therefore, $\sum_{j=1}^{n} P_j L_j$ in Equation (44) increases, indicating more successful transmissions from other nodes during Node i's backoff procedure. Finally, as Node i increases its transmission rate, the network load increases, which essentially increases P_c, the probability that a collision happens in a virtual time slot. More formally, according to Equations (2) and (3), $\tau_k = \frac{P_k}{1-\phi_k}$. As P_k increases following the increase of Node i's load, τ_k increases as well. Therefore, P_c, which can be expressed as:

$$P_c = 1 - \prod_{j=1}^{n}(1-\tau_j) - \sum_{j=1}^{n} \tau_j \prod_{k=1, k\neq j}^{n} (1-\tau_k), \tag{46}$$

increases too since P_c's expression in Equation (46) is a monotonically increasing function regarding any τ_k. Therefore, $B_i P_c T_c$ increases. In conclusion, since as Node i increases its transmission rate, the first three parts on the right side of the equal sign in Equation (44) increases, $\tilde{\Delta}_i > \Delta$. Therefore, using packet delay to predict achievable bandwidth is over-optimistic when there are many unsaturated nodes in the network.

B. Free Bandwidth Model

In [4], the achievable bandwidth at a node is approximated as the free channel bandwidth. As we have pointed out, free bandwidth does not equal the maximum achievable bandwidth to a flow in a contention-based IEEE 802.11 network, since newly arriving flows may be able to "push down" the throughput of existing flows through contention. Therefore, using free bandwidth to predict the maximum achievable bandwidth to a node can be over-pessimistic, especially for new flows that have high priorities.

C. Saturation Model

This method predicts the achievable bandwidth using a model that only represents the bandwidth allocation in a saturated network [3], [5], [6], [11], [13]. However, since not all nodes are sending at their maximum achievable rate, the network is not necessarily saturated. Therefore, this prediction method can often be over-pessimistic, especially in a lightly loaded network.

TABLE I
W CONFIGURATIONS OF PRIORITY CLASSES

Priority	5	4	3	2	1	0
W	15	31	47	63	79	95

VI. EVALUATION

We evaluate the accuracy of the four prediction methods, our prediction model, the delay model, the free bandwidth model and the saturation model by comparing the bandwidth predictions produced by the prediction methods with the actual achievable throughput measured in NS2 simulations [8]. The metrics that are used for the evaluation include the mean (M) and the root-mean-square (RMS) of prediction errors defined as follows:

$$M = \frac{1}{n}\sum_{i=1}^{n}(Prediction_i - ActualThroughput_i), \quad (47)$$

$$RMS = \sqrt{\frac{\sum_{i=1}^{n}(Prediction_i - ActualThroughput_i)^2}{n}}, \quad (48)$$

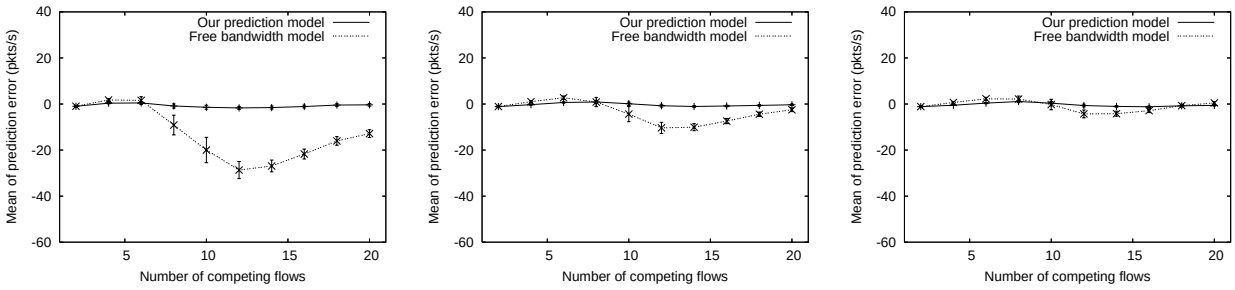
where n is the number of simulations. RMS is used to measure the prediction accuracy, where a small RMS indicates that the prediction is close to the actual achievable bandwidth. M shows the direction of the estimation error. A positive M indicates that the estimation tends to be larger than the actual achievable bandwidth, while a negative M indicates that the estimation tends to be smaller.

In the implementation of our prediction model and the saturation model, every node uses a moving average to collect its load information and periodically broadcasts the load information to its neighboring nodes. In the delay model implementation, a node periodically broadcasts a packet from each priority class to collect per-class packet delay information.

In all of the simulations, the channel transmission rate is 2Mbps, the transmission range is 250m and the carrier-sensing range is 550m. The class of traffic belongs to six priority levels. Table I shows the W configurations for the priorities. The CW^{max} for all classes is 1023. All of the simulations are performed in randomly generated $250m \times 250m$ networks and each simulation lasts for 100 seconds.

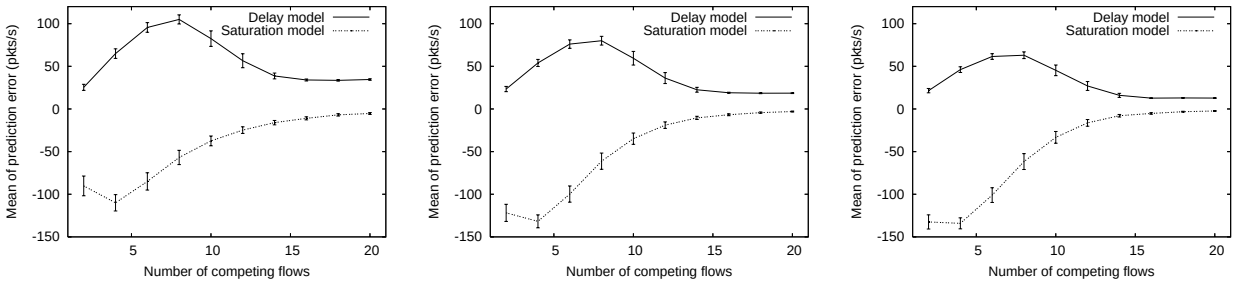
A. Simulations under low flow rate

In each of the first 1500 simulations, 1 to 19 one-hop flows start at the first 10 seconds of the simulation. Each flow carries a CBR source, that has a packet generation rate uniformly distributed in [1,50] packets/second and a traffic priority uniformly distributed between 0 and 5. At the 50th second of these simulations, a new flow that always has backlogged packets starts. The throughput of the new flow,



(a) New flow has priority 4 traffic (b) New flow has priority 3 traffic (c) New flow has priority 0 traffic

Fig. 9. Mean of prediction error (M) of our prediction model and the free bandwidth model. Competing flows' rate range is [1,50] pkts/sec.

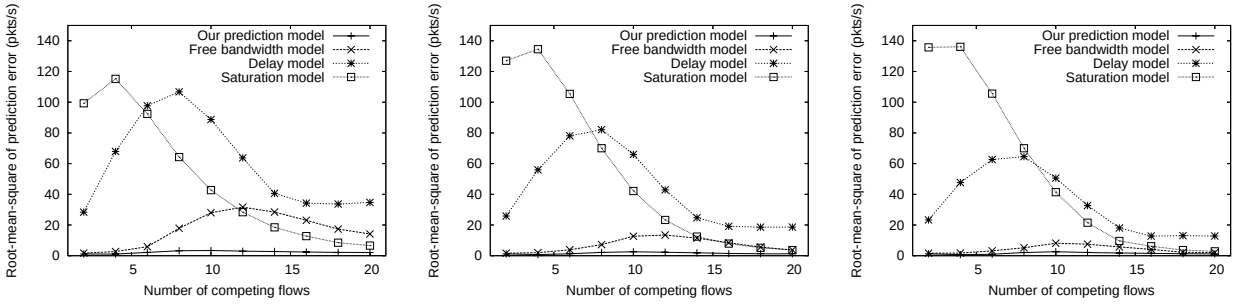


(a) New flow has priority 4 traffic (b) New flow has priority 3 traffic (c) New flow has priority 0 traffic

Fig. 10. Mean of prediction error (M) of the delay and saturation models. Competing flows' rate range is [1,50] pkts/sec.

which is essentially the new flow's achievable bandwidth, is compared with the achievable bandwidth predictions of the new flow obtained through our prediction model, the delay model, the free banwidth model and the saturation model. Due to space limitations, only the simulation results for the new flow with priorities 0, 3 or 4 are shown.

Figure 9 shows M (mean of prediction error) and M's confidence interval (95%) for both our prediction model and the free bandwidth model. Both M and the confidence interval of M are very small ($M < 1.7$ packets/second and the confidence interval is smaller than 0.9 packets/second), indicating consistent



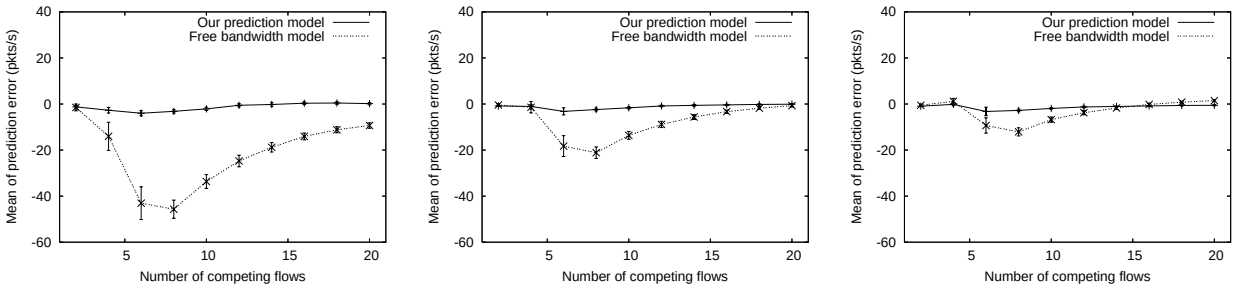
(a) New flow has priority 4 traffic (b) New flow has priority 3 traffic (c) New flow has priority 0 traffic

Fig. 11. Root-mean-square of prediction error (RMS). Competing flows' rate range is [1,50] pkts/sec.

accurate predictions. As expected, since the free bandwidth model does not consider that newly arriving flows may get more bandwidth than the idle bandwidth by reducing the throughput of existing flows, its prediction tends to be over-persimistic ($M < 0$). The free bandwidth model's large confidence interval also indicates that its inaccurate modeling results in large variations in its prediction error. Such inconsistency in prediction error makes it very difficult to accurately compensate for the prediction error. In addition, the free bandwidth model has larger prediction errors for higher priority flows. This is because higher priority flows tend to get more bandwidth from pushing existing flows, resulting in larger gaps between the idle bandwidth and the actual achievable bandwidth.

Figure 10 shows M and its confidence interval (95%) for both the delay and saturation models. (Note that Figure 10 uses a larger scale in the y axis than Figure 9.) Confirming our analysis in Section V-A, the delay model is severely over-optimistic. Figure 10 also shows that the over-estimation of the delay model is at its peak when the number of existing flows are around eight. This is because a major part of the delay model's estimation error comes from the fact that unsaturated nodes increase their transmission probabilities after a new flow starts(See Equations (44) and (45)), which means that fewer unsaturated nodes result in smaller estimation errors. The number of unsaturated nodes can be small when the network has many competing flows, since most of the nodes are saturated in such a case. The number of unsaturated nodes can also be small when there is only very few competing flows. Therefore, the estimation error of the delay model is the largest when the network has neither very few nor very many flows, resulting in the wavy shape of the estimation error in Figure 10. It is also worth noting that the estimation error of the delay model decreases as the priority of the new flow decreases. This can be explained by observing Equations (44) and (45). Essentially, a higher priority results in a smaller contention window size, which implies a higher transmission probability of the new flow (P_i in Equation (45)). This higher transmission probability causes larger changes in the transmission probability of other unsaturated nodes as shown in Equation (45), which essentially leads to a larger discrepency between the delay measurement before the new flow starts and the actual delay that the new flow experiences after it starts. Hence, the prediction error of the delay model is larger for flows with higher priorities.

For the saturation model in Figure 10, when the number of flows is small, the saturation model is too persimistic in estimating achievable bandwidth since the network is far from a saturated state. As the number of flows increases, the network becomes closer to a saturated state and the saturation model's accuracy improves. In addition, Figure 10 also shows that higher priority flows have a smaller prediction



(a) New flow has priority 4 traffic (b) New flow has priority 3 traffic (c) New flow has priority 0 traffic

Fig. 12. Prediction error (M) of our prediction model and the free bandwidth model. Competing flows' rate range is [1,100] pkts/sec.

error under the saturation model. This is because high priority flows are likely to push more lower priority flows to their saturation state, resulting in a network state that is closer to a saturated state.

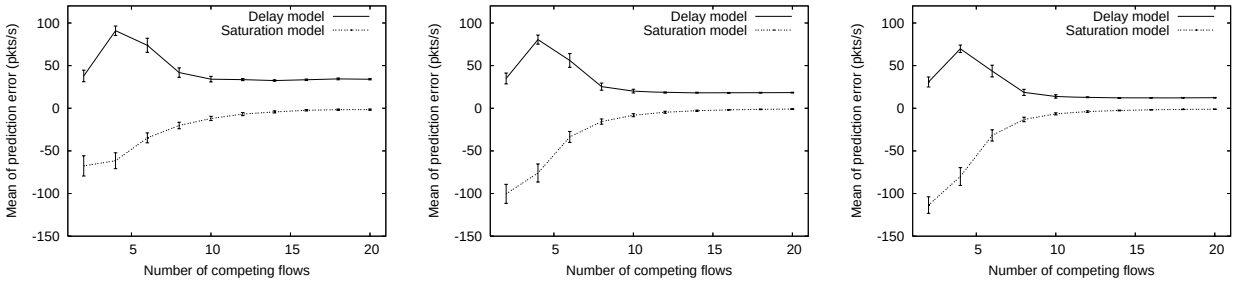
Figure 11 shows the RMS of all of the prediction models. The saturation model is only accurate in terms of RMS when the network is highly loaded, while the free model is only accurate when the network is lightly loaded. The delay model's accuracy is very poor when the network has a medium amount of load. Overall, our prediction model is the only one that provides uniformly accurate prediction for all network states.

B. Simulations under high flow rate

To understand whether the accuracy of the prediction models may be affected by the rate of flows, in the second set of 1500 simulations, we increase the range of transmission rates of flows to [1,100] 512Byte packets per second. The other settings of the simulations are the same as the simulations in Section VI-A. Figures 12 and 13 show M and M's confidence interval (95%) for all prediction models. Figure 14 depicts the RMS of these prediction models. All results in these simulations are very similar to the results in Section VI-A. The only difference is that the peaks of the prediction errors for the saturation, delay and free models appear with fewer flows than in the simultions in Section VI-A due to the increased average flow rates. Our prediction model is again very accurate compared to all other prediction methods.

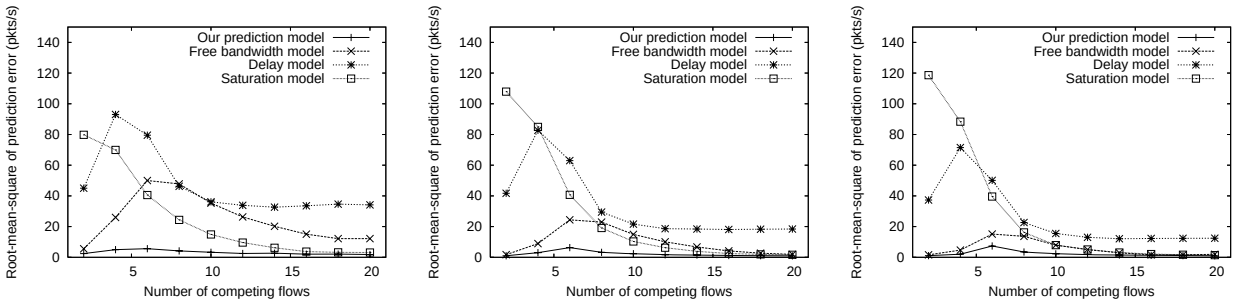
VII. Extending Our Prediction Model To Multihop Networks

Since our prediction model is mainly targeted at single-hop networks, the estimation accuracy of our prediction model degrades when it is extended to be used in multihop environments. The inaccruacy is mainly due to the fact that in multihop networks, each node has different sets of competing neighboring nodes, which may cause severe hidden terminal problems and unfair bandwidth allocations. Neither our



(a) New flow has priority 4 traffic (b) New flow has priority 3 traffic (c) New flow has priority 0 traffic

Fig. 13. Prediction error (M) of the delay and saturation model. Competing flows' rate range is [1,100] pkts/sec.



(a) New flow has priority 4 traffic (b) New flow has priority 3 traffic (c) New flow has priority 0 traffic

Fig. 14. Root-mean-square of prediction error (RMS). Competing flows' rate range is [1,100] pkts/sec.

prediction model nor the other existing prediction methods are able to capture such complex bandwidth allocations in multihop networks. However, despite the inaccuracy of our prediction model, in our simulations, we find that our prediction model still performs better than the other approaches in multihop networks due to our model's consideration of network states. In the rest of the section, we first discuss how to use our prediction model in multihop networks and then we show experiment results for using our prediction model in multihop networks.

A. Prediction of Achievable Bandwidth in Multihop Networks

To predict achievable bandwidth in multihop networks using our prediction model, we first introduce how to obtain neighbor load information in multihop networks and then discuss the prediction process.

1) Collecting Neighbor Load Information: As shown in Equations (38) and (40), to predict a Node i's bandwidth allocation using our prediction model, Node i must know the load on its competing neighbors to calculate η, which determines its bandwidth share. Since in a multihop environment, these competing neighbors may be located outside of Node i's transmission range but inside its carrier-sensing range, Node i must collect its multihop neighbors' load information. Therefore, in our prediction model, every node

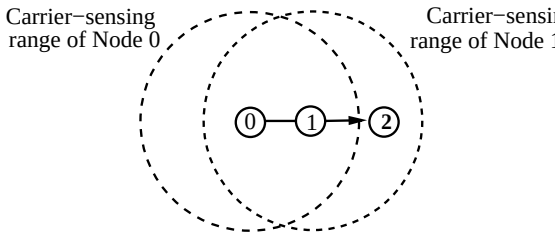


Fig. 15. Multihop Flows

periodically broadcasts its load information to its one-hop neighborhood, including packet arrival rate and average packet size. The broadcast message also carries the load information of its two-hop neighbors, which it has gathered through listening to other nodes' broadcasts. Using this method, every node learns the load on competing nodes in its three-hop neighborhood. We choose the three-hop range since the ratio between IEEE 802.11 network's carrier-sensing range and transmission range is between 2 and 3. The broadcast rate is set low to ensure that information exchanged between nodes is the average of the traffic load rather the burstiness of the traffic.

2) Achievable Bandwidth Prediction: For a multihop flow, the minimum achievable bandwidth among the nodes on the route is not equal to the achievable bandwidth of the flow. This is because the nodes on the route of the flow also contend with each other. For example, Figure 15 depicts a flow with rate R_f and route $0 \rightarrow 1 \rightarrow 2$. Since Nodes 0 and 1 are in each other's carrier-sensing range, only one node can transmit at a time. Therefore, Node 0 must share its bandwidth with Node 1 and both Nodes 0 and 1 experience a traffic load of $2R_f$ on the channel. (Node 2 does not transmit data since it is the sink of the flow.) As R_f increases, the first node among Node 0 and Node 1 that turns saturated becomes the bottleneck of the flow and determines the maximum throughput of the flow.

In general, if the bottleneck node is Node b, the achievable bandwidth R_f of a flow equals the saturation throughput of Node b when Node b is competing with other nodes on the route. Since Node b is the bottleneck, the other nodes on the route are unsaturated when the flow is sending at rate R_f. If there are α nodes on the route that are also in Node b's contention range, using Equation (40), the η at Node b when the flow achieves its sending rate R_f is:

$$\eta = \frac{\frac{L_f}{W_b} + \sum_{i \in N_1} \frac{L_i}{W_i}}{1 - \sum_{i \in N_2} \frac{R_i L_i}{C} - \alpha \frac{R_f L_f}{C}}, \tag{49}$$

where $\frac{L_f}{W_b}$ represents the saturated Node b's load on the network while $\alpha \frac{R_f L_f}{C}$ captures the load of the α unsaturated nodes on the route of the flow. Since Node b learns the identities of its contending neighbors through the periodic exchange of load information, α can easily be calculated given the route of the flow.

When the flow rate reaches R_f, Node b turns from unsaturated to saturated, $\eta = \eta_b^*$. Combining with Equation (39),

$$R_f = \frac{C}{\eta_b^* W_b} = \frac{C}{\eta W_b}. \tag{50}$$

By replacing R_f in Equation (49) using Equation (50) and then solving for η,

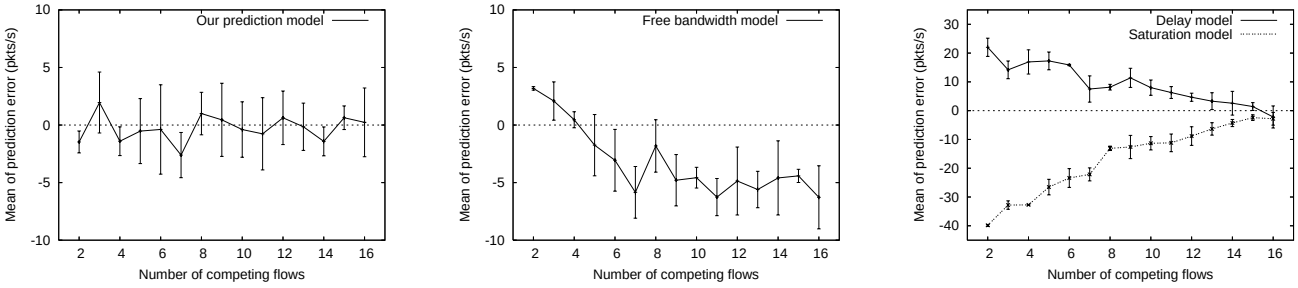
$$\eta = \frac{(\alpha+1)\frac{L_f}{W_b} + \sum_{i \in N_1} \frac{L_i}{W_i}}{1 - \sum_{i \in N_2} \frac{R_i L_i}{C}}. \tag{51}$$

Essentially, Equation (51) is the same as Equation (42) except that the new load $U_k = (\alpha+1)\frac{L_f}{W_b}$. Hence, the same piece-wise linear function described in Section IV can be used to calculate η and then be used to predict the maximum throughput of Node b (See Appendix C).

To obtain the achievable bandwidth for a flow, every node along the route of the flow can assume it is the bottleneck and predict its maximum throughput using Algorithm 1 in Appendix C. The smallest prediction equals the achievable bandwidth of the entire flow. Hence, using our bandwidth allocation model, we can estimate the the achievable bandwidth for a multihop flow in a multihop network.

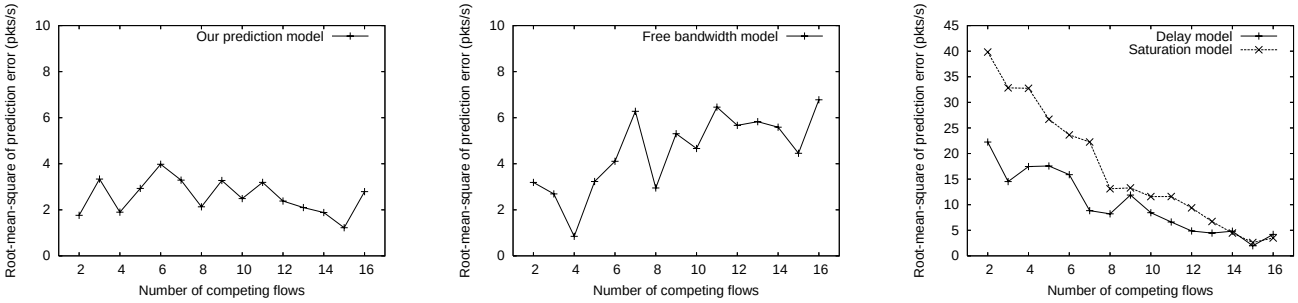
B. Multihop evaluation

To evaluate the performance of our prediction model in multihop networks, we generate 300 $1000m \times 1000m$ random topologies with 100 nodes. In each network, 2 to 16 randomly located active nodes start to send CBR traffic in the first 10 seconds of the simulation. The rate of each CBR flow is uniformly distributed in [1,50] 512Byte packets per second. At the 50th second, a new flow starts. The maximum throughput of the new flow is compared to the achievable bandwidth predictions generated by the four prediction models. Figure 16 shows the mean of the prediction error and its confidence interval (95% confidence) when the new flow has priority 1 and the hop count of the new flow is 5. Figure 17 depicts the root-mean-square of the prediction error. (Note that the scales of the y axis in Figure 16(a), (b) and (c) and Figure 17(a), (b) and (c) are different.) Figures 16 and 17 demonstrate that our analytical results in Section V arre still valid in multihop networks. The saturation model is over-pessimistic and the delay model is over-optimistic in lightly and medium loaded networks. The free bandwidth model's prediction becomes over-pesimisitic when the network load increases. Although the prediction accuracy of our model is degraded compared to the simulation results in single-hop networks, our model still consistently provides more accurate estimations than the other methods and the network load level does not impact the accuracy of our prediction model. We also examined the performance of our prediction



(a) Our prediction model (b) Free bandwidth model (c) Delay model and Saturation model

Fig. 16. Mean of prediction error for five-hop priority 1 new flow.



(a) Our prediction model (b) Free bandwidth model (c) Delay model and Saturation model

Fig. 17. Root-mean-square of prediction error for five-hop priority 1 new flow.

model under different priority classes and different contention window allocations for the priority classes. These results are very similar to Figures 16 and 17 and can be found in our technical report [18].

VIII. Conclusions and Future Work

In this paper, we have presented a simple analytical model to predict the achievable bandwidth of a new flow in a wireless network using IEEE 802.11. Our model is extremely flexible and covers networks with or without service differentiation. The strength of our model is that it captures achievable bandwidth estimation in the MAC layer for saturated, non-saturated or semi-saturated networks. Simulation results show that our model is very accurate in predicting achievable bandwidth for single-hop networks and performs much better than all existing approaches.

Since our model is still not very accurate in multihop networks, part of our future work is to investigate more accurate bandwidth allocation models for multihop wireless networks so that better predictions of achievable bandwidth can be obtained. In addition, due to the impacts of hidden terminals on the fairness of bandwidth allocation in multihop networks, we are also studying methods that can enable a node to

discover its hidden terminals and quantify the impacts of hidden terminals on its communications with other nodes. Finally, our current prediction model uses a conservative assumption that the traffic generation rates at the sources of existing flows do not decrease after the arrival of a new flow. However, for TCP flows, this assumption may be too conservative. Therefore, another topic of our future research about achievable bandwidth prediction is to identify the joint effects of transport layer congestion control and MAC layer bandwidth allocation.

REFERENCES

[1] Imad Aad and Claude Castelluccia. Differentiation Mechanisms for IEEE 802.11. In *IEEE INFOCOM*, 2001.

[2] Gahng-Seop Ahn, Andrew Campbell, Andras Veres, and Li-Hsiang Sun. SWAN: Service Differentiation in Stateless Wireless Ad Hoc Networks. In *IEEE INFOCOM*, 2002.

[3] Albert Banchs, Xavier Perez-Costa, and Daji Qiao. Providing Throughput Guarantees in IEEE 802.11e Wireless LANs. In *Proceeding of the 18th International Teletraffic Congress(ITC-18)*, 2003.

[4] Michael G. Barry, Andrew T. Campbell, and Andras Veres. Distributed Control Algorithms for Service Differentiation in Wireless Packet Networks. In *IEEE INFOCOM*, 2001.

[5] Giuseppe Bianchi. Performance analysis of the IEEE 802.11 distributed coordination function. *IEEE Journal on Selected Areas in Communications*, 18(3), 2000.

[6] Frederico Cali, Marco Conti, and Enrico Gregori. Dynamic Tuning of the IEEE 802.11 Protocol to Achieve a Theoretical Throughput Limit. *IEEE/ACM TRANSACTION ON NETWORKING*, 8(6):785–799, Dec. 2000.

[7] Kai Chen and Klara Nahrstedt. Exact: An explicit rate-based flow control framework in manet. Technical Report UIUCDCS-R-2002-2286/UILU-ENG-2002-1730, 1 2003.

[8] Kevin Fall and Kannan Varadhan. NS notes and documentation. In *The VINT Project, UC Berkely, LBL, USC/ISI, and Xerox PARC*, 1997.

[9] Manthos Kazantzidis, Mario Gerla, and Sung-Ju Lee. Permissible Throughput Network Feedback for Adaptive Multimedia in AODV MANETs. In *IEEE International Conference of Communications (ICC)*, 2001.

[10] Byung-Jae Kwak, Nah-Oak Song, and Leonard E. Miller. Analysis of the Stability and Performance of Exponential Backoff. In *IEEE Wireless Communications And Networking Conference(WCNC)*, 2003.

[11] Bo Li and Roberto Battiti. Performance Analysis of An Enhanced IEEE 802.11 Distributed Coordination Function Supporting Service Differentiation. In *International Workshop on Quality of Future Internet Service*, 2003.

[12] Stefan Mangold, Sunghyun Choi, Peter May, Ole Klein, Guido Hiertz, and Lothar Stibor. IEEE 802.11e Wireless LAN for Quality of Service. In *Proceedings of European Wireless*, 2002.

[13] P.Chatzimisios, V. Vitsas, and A. C. Boucouvalas. Throughput and Delay analysis of IEEE 802.11 protocol. In *Proceedings of the 5th IEEE International Workshop on Networked Appliances*, 2002.

[14] Samarth H. Shah, Kai Chen, and Klara Nahrstedt. Available Bandwidth Estimation in IEEE 802.11-based Wireless Networks. In *The 1st Bandwidth Estimation Workshop (BEst 2003)*, 2003.

[15] Samarth H. Shah, Kai Chen, and Klara Nahrstedt. Dynamic Bandwidth Management for Single-hop Ad Hoc Wireless Networks. In *Proceedings of IEEE International Conference on Pervasive Computing and Communications (PerCom)*, 2003.

[16] IEEE Computer Society. 802.11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications.

[17] Yaling Yang and Robin Kravets. Throughput Guarantees for Multi-priority Traffic in Ad Hoc Networks. In *International Conference on Mobile Ad-hoc and Sensor Systems (MASS)*, October 2004.

[18] Yaling Yang, Jun Wang, and Robin Kravets. Achievable bandwidth prediction in ieee 802.11 networks. Technical Report UIUCDCS-R-2005-2367, Department of Computer Science, University of Illinois at Urbana-Champaign, 2005.

[19] Jun Zhao, Zihua Guo, Qian Zhang, and Wenwu Zhu. Performance study of MAC for Service Differentiation in IEEE 802.11 WLAN. In *Procceedings of IEEE Globecom*, 2002.

APPENDIX

A. Expression of $\tau_{i,sat}$

Lemma 1: For a saturated Node i, the probability that it transmits in a randomly chosen virtual time slot is:

$$\tau_{i,sat} = \frac{2(1-2\phi_i)}{(1-2\phi_i)(W_i+2)+\phi_i(W_i+1)(1-(2\phi_i)^{m_i})}, \tag{52}$$

Proof: Denote $p_i\{j_1,k_1|j_0,k_0\}$ as Node i's transition probability from state $\{j_0,k_0\}$ to state $\{j_1,k_1\}$, which can be expressed as:

$$p_i\{j_1,k_1|j_0,k_0\} = \Pr\{s(t+1)=j_1, b(t+1)=k_1 | s(t)=j_0, b(t)=k_0\}. \tag{53}$$

Based on the Markov chain in Figure 3, for a saturated Node i, since $Q_i = 1$, the only non null one-step transition probabilities are:

$$\begin{cases} p_i\{j,k|j,k+1\} = 1 & k \in (0, 2^jW_i-2) \quad j \in (0,m_i) \\ p_i\{0,k|j,0\} = (1-\phi_i)/W_i & k \in (0, W_i-1) \quad j \in (0,m_i) \\ p_i\{j,k|j-1,0\} = \phi_i/(2^jW_i) & k \in (0, 2^jW_i-1) \quad j \in (1,m_i) \\ p_i\{m_i,k|m_i,0\} = \phi_i/(2^{m_i}W_i) & k \in (0, 2^{m_i}W_i-1). \end{cases} \tag{54}$$

Let $b_i(j,k) = \lim_{t\to\infty}\Pr\{s(t)=j, b(t)=k\}, j \in (0,m_i), k \in (2^jW_i-1)$ be the stationary distribution of the Markov chain. In steady state, we can derive the following relationships through chain regularities.

$$b_i(j,0) = \phi_i^j b_i(0,0), \quad 0<j<m_i, \tag{55}$$

$$b_i(m_i,0) = \frac{\phi_i^{m_i}}{1-\phi_i} b_i(0,0), \tag{56}$$

$$b_i(j,k) = \frac{2^jW_i-k}{2^jW_i} \begin{cases} (1-\phi_i)\sum_{l=0}^{m_i} b_i(l,0) & j=0 \\ \phi_i b_i(j-1,0) & 0<j<m_i \\ \phi_i[b_i(m_i-1,0)+b_i(m_i,0)] & j=m_i \end{cases} \tag{57}$$

$$\sum_{j=0}^{m_i}\sum_{k=0}^{2^jW_i-1} b_i(j,k) = 1 \tag{58}$$

From Equations (55) to (58), we have:

$$b_i(0,0) = \frac{2(1-2\phi_i)(1-\phi_i)}{(1-2\phi_i)(W_i+1)+\phi_i W_i(1-(2\phi_i)^{m_i})}. \tag{59}$$

Since Node i transmits when its backoff timer reaches 0, the probability $\tau_{i,sat}$ that Node i transmits in a randomly chosen virtual time slot can be expressed as:

$$\tau_{i,sat} = \sum_{j=0}^{m_i} b_i(j,0) = \frac{b_i(0,0)}{1-\phi_i} = \frac{2(1-2\phi_i)}{(1-2\phi_i)(W_i+2)+\phi_i(W_i+1)(1-(2\phi_i)^{m_i})}. \tag{60}$$

■

B. Uniqueness of η

The solution (η, k) to Equation (40) and Inequality (41) is unique. This can be proved as follows. If there are two solutions (η_1, k_1) and (η_2, k_2), with $k_1 < k_2$, then the following inequality holds:

$$\eta(k_1) \;<\; \eta^*_{x_{k_1+1}} \tag{61}$$

$$\eta(k_2) \;\geq\; \eta^*_{x_{k_2}}. \tag{62}$$

Combining Inequality (61) and (62) with Equations (39) and (40), we get:

$$\frac{\sum_{i=1}^{k_1}\frac{L_{x_i}}{W_{x_i}}}{1-\sum_{i=k_1+1}^{n}\frac{R_{x_i}L_{x_i}}{C}} \;<\; \frac{C}{R_{x_{k_1+1}}W_{x_{k_1+1}}} \tag{63}$$

$$\frac{\sum_{i=1}^{k_2}\frac{L_{x_i}}{W_{x_i}}}{1-\sum_{i=k_2+1}^{n}\frac{R_{x_i}L_{x_i}}{C}} \;\geq\; \frac{C}{R_{x_{k_2}}W_{x_{k_2}}} \tag{64}$$

According to Inequality (63), we have:

$$\sum_{i=1}^{k_1}\frac{L_{x_i}}{W_{x_i}} < \frac{1}{R_{x_{k_1+1}}W_{x_{k_1+1}}}(C-\sum_{i=k_1+1}^{n}R_{x_i}L_{x_i}) = \frac{1}{R_{x_{k_1+1}}W_{x_{k_1+1}}}(C-\sum_{i=k_2+1}^{n}R_{x_i}L_{x_i}-\sum_{i=k_1+1}^{k_2}R_{x_i}L_{x_i}) \tag{65}$$

According to Inequality (64), we have:

$$\sum_{i=1}^{k_1}\frac{L_{x_i}}{W_{x_i}} \geq \frac{1}{R_{x_{k_2}}W_{x_{k_2}}}(C-\sum_{i=k_2+1}^{n}R_{x_i}L_{x_i})-\sum_{i=k_1+1}^{k_2}\frac{L_{x_i}}{W_{x_i}} = \frac{1}{R_{x_{k_2}}W_{x_{k_2}}}(C-\sum_{i=k_2+1}^{n}R_{x_i}L_{x_i}-\sum_{i=k_1+1}^{k_2}R_{x_i}L_{x_i}\frac{R_{x_{k_2}}W_{x_{k_2}}}{R_{x_i}W_{x_i}}) \tag{66}$$

Note that $\eta^*_{x_i} < \eta^*_{x_j}$ if $i < j$. Therefore, based on Equation (39), we have:

$$\frac{1}{R_{x_i}W_{x_i}} < \frac{1}{R_{x_j}W_{x_j}}, \forall i < j. \tag{67}$$

Therefore, $\frac{R_{x_{k_2}}W_{x_{k_2}}}{R_{x_i}W_{x_i}} < 1, \forall i < k_2$. Hence Inequality (66) can be further derived to be:

$$\sum_{i=1}^{k_1}\frac{L_{x_i}}{W_{x_i}} \geq \frac{1}{R_{k_2}W_{k_2}}(C - \sum_{i=k_2+1}^{n} R_{x_i}L_{x_i} - \sum_{i=k_1+1}^{k_2} R_{x_i}L_{x_i}\frac{R_{x_{k_2}}W_{x_{k_2}}}{R_{x_i}W_{x_i}}) \tag{68}$$

$$\geq \frac{1}{R_{k_2}W_{k_2}}(C - \sum_{i=k_2+1}^{n} R_{x_i}L_{x_i} - \sum_{i=k_1+1}^{k_2} R_{x_i}L_{x_i}) \tag{69}$$

$$\geq \frac{1}{R_{k_1+1}W_{k_1+1}}(C - \sum_{i=k_2+1}^{n} R_{x_i}L_{x_i} - \sum_{i=k_1+1}^{k_2} R_{x_i}L_{x_i}) \tag{70}$$

Comparing Inequality (65) with (70), we get a paradox. Therefore, it is impossible that there are two solutions of (η, k). Hence, the solution of (η, k) is uniquely defined by Equation (40)) and Inequality (41).

C. Algorithm

BANDWIDTH_PREDICTION ($L_{new}, W_{new}, C, \alpha$)
  $U_{new} \leftarrow (\alpha+1)\frac{L_{new}}{W_{new}}$
  for $i = 1$ to n **do**
    if $U^*_{new,i} \leq U_{new} < U^*_{new,i+1}$ **then**
      $\eta \leftarrow \frac{X_i + U_{new}}{1 - Y_i}$; BREAK
    end if
  end for
  $S_{new} \leftarrow \frac{U_{new}}{(\alpha+1)\eta}C$;

INITIALIZATION ($\mathbf{R}, \mathbf{L}, \mathbf{W}, C$)
  for $i = 1$ to n **do**
    $\eta^*_i \leftarrow \frac{C}{W_i R_i}$
  end for
  SORT ($\mathbf{R}, \eta^*$); SORT ($\mathbf{L}, \eta^*$); SORT ($\mathbf{W}, \eta^*$)
  /* Building the mapping between η and U_{new} */
  $X_0 \leftarrow 0$
  $Y_0 \leftarrow \sum_{i=1}^{n} \frac{R_i L_i}{T}$
  for $i = 1$ to n **do**
    $X_i \leftarrow X_{i-1} + \frac{L_i}{W_i}$
    $Y_i \leftarrow Y_{i-1} - \frac{R_i L_i}{C}$
    $U^*_{new,i} \leftarrow \eta^*_i(1 - Y_i) - X_i$
  end for

SORT($array, index$)
  Sort $array$ in ascending order of $index$

Algorithm 1: The algorithm that predicts achievable bandwidth of a node

D. Notation

1) Q_i: the probability that Node i's queue is not empty when it finishes a successful packet transmission.
2) λ_i: the probability that a packet arrives at Node i during a virtual time slot.

3) P_B: the probability that the channel is busy.
4) $\mathcal{N}$: Set of transmitting nodes.
5) n: number of competing nodes.
6) CW: contention window size.
7) CW^{min} or W_i: minimum contention window size.
8) CW^{max}: maximum contention window size.
9) S_i: fraction of channel bandwidth allocated to Node i.
10) P_i: probability that Node i successfully transmits a packet in a virtual slot.
11) Subscript sat: saturated node.
12) Subscript $\overline{sat}$: unsaturated node.
13) R_i: the average packet arrival rate at Node i.
14) L_i: the average duration of a successful packet transmission at Node i, including a full RTS/CTS/DATA/ACK handshake.
15) τ_i: probability that Node i transmits in a randomly chosen virtual time slot.
16) ϕ_i: the probability of collision of Node i's transmission when Node i transmits in a virtual time slot.
17) C: the maximum utilization of the channel, which is the maximum fraction of channel bandwidth that can be used for successful RTS/CTS/DATA/ACK handshakes.
18) m: the number of collisions that are needed for the contention window size to reach CW^{max}.
19) N_1: saturated node set.
20) N_2: unsaturated node set.
21) ρ: the average number of bits transmitted in a virtual time slot.
22) η: the part of ρ that does not depend on to τ_i or ϕ.
23) α: the part of ρ that depends on to τ_i or ϕ.
24) η_i^*: the threshold value of η at when node i is at the edge of turning from unsaturated to saturated.
25) U_k: the load that Node k adds on the network when it achieves $S_{k,sat}$.
26) Δ_i: average packet transmission delay at the MAC layer.
27) $\tilde{\Delta}_i$: the packet transmission delay when Node i is saturated.