An Adaptive Strategy for Maximizing Throughput in MAC layer Wireless Multicast

Prasanna Chaporkar
Department of Electrical and Systems Engineering
University of Pennsylvania
prasanna@seas.upenn.edu

Anita Bhat
Department of Electrical and Systems Engineering
University of Pennsylvania
abhat@seas.upeen.edu

Saswati Sarkar
Department of Electrical and Systems Engineering
University of Pennsylvania
swati@ee.upeen.edu

ABSTRACT

Bandwidth efficiency of wireless multicast can be improved substantially by exploiting the fact that several receivers can be reached at the MAC layer by a single transmission. The multicast nature of the transmissions, however, introduces several design challenges, and systematic design approaches that have been used effectively in unicast and wireline multicast do not apply in wireless multicast. For example, a transmission policy that maximizes the stability region of the network need not maximize the network throughput. Therefore, the objective is to design a policy that decides when a sender should transmit in order to maximize the system throughput subject to maintaining the system stability. We present a sufficient condition that can be used to establish the throughput optimality of a stable transmission policy. We subsequently design an adaptive stable policy that allows a sender to decide when to transmit using simple computations based only on limited information about current transmissions in its neighborhood, and without using any information about the network statistics. The proposed policy attains the same throughput as the optimal offline stable policy that uses in its decision process past, present, and even future network states. We prove the throughput optimality of this policy using the sufficient condition and the large deviation results. We present a MAC protocol for acquiring the local information necessary for executing this policy, and implement it in *ns-2*. The performance evaluations demonstrate that the optimal strategy significantly outperforms the existing approaches in adhoc networks consisting of several multicast and unicast sessions.

Categories and Subject Descriptors

C.2 [**Computer-Communication Networks**]: Network Architecture and Design

[1]This research was supported in part by NSF grants ANI01-06984 and NCR02-38340.

General Terms

Algorithms, Design

Keywords

Wireless multicast, MAC layer scheduling, throughput optimal policy, stability

1. INTRODUCTION

Many current day wireless applications need *multicast* communication, e.g., sensor networks and military operations. Wireless communication is inherently broadcast in nature, i.e. all nodes in the transmission range of a sender can receive a transmission from the sender. Hence, at the MAC layer, it suffices to transmit each packet only once in order to reach all the intended receivers. *We focus on designing an optimal MAC strategy for wireless multicast that utilizes the broadcast nature of the wireless medium.*

Though the broadcast nature of wireless transmissions provides a possible approach to improve the efficiency of the multicast communication, it also imposes various difficulties. A multicast specific challenge is that some but *not all* the receivers may be ready to receive. For example, in Figure 1, when S_2 is transmitting to R_5, R_2 can not receive the transmission from S_1 as both the transmissions will collide at R_2. However, R_1, R_3 and R_4 can still receive the transmission. The readiness state of a receiver depends on the network load. The policy decision is whether S_1 should transmit or it should wait till all the receivers are ready.

A transmission policy that does not transmit until a sufficient number of receivers are ready may lead to unstable systems that have unbounded queue lengths at the senders. On the other hand, if the senders transmit when only a few receivers are ready, then the transmitted packet will be lost at the receivers that were not ready, which may result in low system throughput. Thus, there is a tradeoff between system stability and the throughput. The system clearly needs to be stable. *The challenge therefore is to design a MAC layer transmission strategy that maximizes the system throughput, while maintaining system stability.* Furthermore, the optimum transmission strategy should be such that each sender decides whether or not to transmit using (a) simple computations, (b) no information about system statistics, (c) limited control message exchange and (d) limited information about its neighbors.

We propose a transmission strategy that attains the above objectives (Section 3). First step in this direction has been to develop a model that captures the essential features of a wireless network

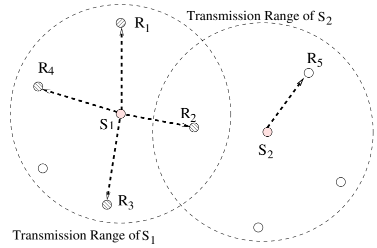


Figure 1: An example to demonstrate the advantages and the challenges associated with wireless multicast. The figure shows two senders S_1, S_2 and 5 receivers R_1 to R_5. R_1 to R_4 are S_1's receivers, and R_5 is S_2's receiver. Dashed circle indicates the communication range of a sender.

(Section 2). We subsequently use this model to obtain a sufficient condition to establish the throughput optimality of an arbitrary stable policy. Using the model, we show that the proposed policy maximizes throughput among all policies that stabilize the system [5]. The transmission decisions of the proposed policy are based on the queue length at a sender and the number of ready receivers. The first quantity is easily available at a sender. We propose a few MAC protocols that allow a sender to estimate the number of ready receivers (Section 4). We evaluate the performance of various multicast schemes using ns-simulations in a wireless network consisting of several multicast and unicast sessions (Section 5). Simulation results show that the proposed optimal policy provides significantly higher throughput than existing approaches. We present discussion on the model and the analytical results in Section 6.

The previous research in wireless multicast have lead to the development of transport and network layer protocols. End-to-end error recovery protocols address the issue of reliable loss recovery with minimum cost of information exchange among nodes, e.g., [4, 17]. Several multicast routing protocols have been proposed at the network layer, e.g., [8, 13, 16, 19, 23]. Protocols for energy efficient multicast routing have been proposed in [22, 29, 31]. Zhou *et al.* have investigated content based multicast in adhoc networks [32]. Nagy *et al.* have investigated multicast in cellular networks [15]. The proposed transport and network layer protocols can work with any underlying MAC layer strategy. Though efficiency of these higher layer schemes depends on the efficiency of MAC layer strategy, MAC layer multicast has not been adequately explored.

Singh *et al.* have proposed a MAC protocol for power aware broadcast [22]. Wang *et al.* have proposed a scheduling and power control protocol to minimize the transmission power at the sender nodes [30]. Jaikaeo *et al.* have studied multicast with directional antennas [11]. Kuri *et al.* have proposed a protocol for reliable packet delivery in wireless LANs [12]. The design of the protocol is based on many assumptions that hold in wireless LANs but not in ad-hoc networks. For ad-hoc networks, Tang *et al.* have proposed a *unicast based multicast* scheme that transmits a packet to each receiver separately in round robin fashion [26]. Currently popular multiple access standard IEEE 802.11 implements multicast by broadcasting a packet after disabling all control messages. Thus, second hop interference is ignored. Tang *et al.* have also proposed the *threshold*-1 *multicast* scheme where a sender transmits a packet whenever at least one receiver is ready to receive [24, 25]. The *unicast based multicast* policy does not exploit the broadcast nature of wireless medium, and its multiple transmissions of a packet wastes power and bandwidth. The *broadcast based multicast* and *threshold-1 multicast* policies cause packet loss at receivers because several receivers may not be ready at the time of transmission. The *broadcast based multicast* also causes packet collision due to second hop interference. We have proposed a throughput optimal multicast strategy in [6]. The parameters of this policy can only be computed with knowledge of the network load and statistics of the arrival process, which sender may not know. We now present an adaptive policy that maximizes the throughput without requiring any such information.

We believe that MAC layer multicast presents several research challenges and design issues that would be of interest to both theoreticians and practitioners. Considering all such issues is beyond the scope of a single paper. We, however, outline open problems in this area in Section 7. We hope that our results would lay a basis for developing a fully operational MAC layer multicast protocol, and stimulate further research to address the open problems. *Due to the space constraints proofs for all the results are presented in* [5].

2. SYSTEM MODEL

We consider a wireless network with several MAC layer multicast and unicast sessions. All the nodes in the network need not be in each other's transmission range. Each multicast session comprises of a sender and a set of receivers (multicast group). At the MAC layer all the receivers are within the sender's transmission range. We consider transmission of data traffic. Time is slotted. We assume that each packet can be transmitted in a single slot; this assumption can be relaxed easily.

A major design challenge in wireless multicast is that several existing approaches for optimizing system performance do not apply. Consider the objective of maximizing system throughput in a network with n senders generating packets at rates $\lambda_1, \dots, \lambda_n$ respectively. Consider only the policies that ensure correct reception of every packet by at least one receiver. Now, throughput is the sum of the number of packets received correctly per unit time over all the receivers. The stability region of a transmission policy is the set of arrival rates $\vec{\lambda} = (\lambda_1, \dots, \lambda_n)$ for which the senders have finite expected queue lengths. The stability region of the network (denoted as Λ) is the union of that of all transmission policies. In unicast and wireline multicast, a policy maximizes throughput if and only if its stability region equals Λ. The latter happens if there exists a lyapunov function that has a negative drift for the policy in Λ. This property has been used to show that back-pressure based policies maximize the stability region and hence the system throughput in packet radio and wireline multicast networks [27, 20]. This systematic approach cannot be used in wireless multicast as a policy that attains Λ need not maximize the throughput and vice-versa.

Example: Consider Figure 1. When S_1 transmits, R_1, R_2, R_3 receive the packet without any error; R_4 receives the packet only if S_2 is not transmitting simultaneously. When S_2 transmits, R_5 receives the packet without any error. Consider two transmission policies Δ_1 and Δ_2. Under Δ_1, each sender transmits whenever it has a packet. Under Δ_2, S_2 transmits whenever it has a packet, while S_1 transmits only when S_2 is not transmitting. We assume that S_1 knows S_2's transmission decisions, and in each slot a packet arrives at S_1 (S_2) with probability λ_1 (λ_2). Policy Δ_1's stability

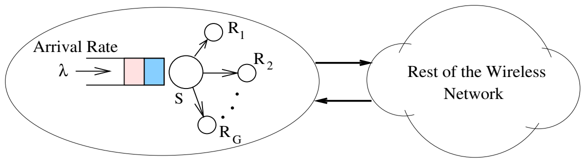


Figure 2: Figure shows a MAC layer multicast session and its interaction with the rest of the network. Node S is a sender, where packets arrive at the rate λ. Sender S transmits the packets to receivers R_1 to R_G.

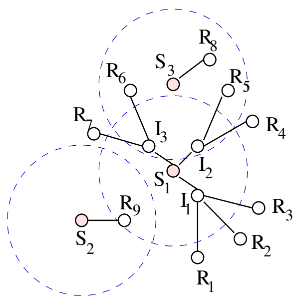


Figure 3: Figure shows a multicast session from S_1 to receivers R_1 to R_7 and two unicast sessions from S_2 to R_9 and S_3 to R_8 in a multi-hop wireless network. First we observe that a single network layer multicast session corresponds to many MAC layer multicast sessions, e.g., S_1 multicasts to intermediate nodes I_1 to I_3, I_1 multicasts to the receivers R_1 to R_3, etc. Consider a MAC layer multicast session from sender S_1 to MAC layer receivers I_1, I_2 and I_3. Observe that when R_9 is receiving data, S_1 is not ready to transmit, but all the receivers are ready to receive. Furthermore, when S_3 is transmitting to R_8, receivers I_2 and I_3 are not ready to receive, but S_1 is ready to transmit and I_1 is is ready to receive. Thus, readiness states of I_2 and I_3 are correlated.

region is $\Lambda_1 = \{\vec{\lambda} : 0 \leq \lambda_i < 1, i = 1, 2\}$. This is also the network's stability region as a sender can transmit only one packet in each slot. The network throughput under Δ_1 for arrival rates $\vec{\lambda} \in \Lambda_1$ is $\lambda_1(4 - \lambda_2) + \lambda_2$. Now, Δ_2's stability region is $\Lambda_2 = \{\vec{\lambda} : 0 \leq \lambda_2 < 1, 0 \leq \lambda_1 < 1 - \lambda_2\}$, which is a strict subset of the network's and Δ_1's stability region. The throughput under Δ_2 for arrival rates $\vec{\lambda} \in \Lambda_2$ is $4\lambda_1 + \lambda_2$. Thus, the throughput under Δ_2 is strictly higher than that under Δ_1 for $\vec{\lambda}$ in Λ_2, $\lambda_2 > 0$. Thus, unlike Δ_1, Δ_2 attains the stability region of the network, but for certain arrival rates its throughput is less than Δ_1's throughput.

The above observation has two consequences. First, we must maximize the throughput subject to stability. In other words, we must design a stable transmission policy that maximizes the throughput among all the stable policies. Second, the existing framework does not apply. Therefore, we need a new design technique to attain the objective. We develop a sufficient condition for a policy to attain the objective using an abstraction of a network which we describe next.

Figure 2 shows a single MAC layer multicast session, and represents its interaction with the rest of the network. Due to the broadcast nature of wireless medium, transmissions from other nodes in the network affect the performance of the multicast session and vice versa. The effect of the rest of the network on the multicast session is that the receivers are not always ready to receive. A receiver will not be ready when there are transmissions in its neighborhood or the transmission condition is poor, or when it is in a sleep mode. For example, in Figure 3 the receivers I_2 and I_3 will not be ready when S_3 is transmitting to R_8. Further, the readiness states of different receivers are correlated in the same slot. The correlation across slots is due to bursty channel errors. The impact of the session on the rest of the network is that the sender's transmission interferes with simultaneous transmissions in its neighborhood. The interference is controlled as follows. The sender does not transmit if any node in its neighborhood is receiving a packet. For example in Figure 3, S_2 does not transmit when S_1 is transmitting a packet to R_9. Also, the sender backs-off just after transmitting a packet so that other senders can use the shared medium. Thus, a sender is not ready when it backs off or a node in its neighborhood is receiving a packet. Thus, the effect of the session on the rest of the network is controlled by regulating the sender's readiness states. The readiness states of the receivers may be correlated with the readiness state of the sender.

We consider a single multicast session with G receivers, and model its interaction with the rest of the network by considering stochastic readiness states of the sender and the receivers. For example, in Figure 2 we only consider the sender S and the receivers R_1 to R_G, and assume that the readiness states are governed by a stationary and ergodic stochastic process. The readiness process in a slot is described by a $G+1$ dimensional vector $\vec{j} = [j_0\ j_1\ j_2\ \ldots\ j_G]$, where the component j_0 is 1 if the sender is ready and it is 0 otherwise. Further for all $l \geq 1$, j_l is 1 if the l^{th} receiver is ready and it is 0 otherwise.

The packet arrival process at the sender is an irreducible, aperiodic and time homogeneous Markov Chain (MC) of γ states. A state of the MC indicates the number of arrivals in a slot. Here γ denotes the maximum number of packets arriving in a slot, and λ denotes the expected number of arrivals in a slot under the MC's stationary distribution. Next we present some definitions that will be used in the rest of the paper.

DEFINITION 1. *A transmission policy is an algorithm at a sender node that decides when to transmit a packet. A necessary condition for a sender to transmit a packet is that it is ready to transmit, and it has a packet to transmit*

The class of transmission policies include *offline* strategies that use in the decision process a prior knowledge of packet arrivals and readiness states at all (*including future*) slots.

DEFINITION 2. *A reward for a packet is the number of receivers that receive the packet successfully.*

DEFINITION 3. *System throughput is the expected reward per unit time.*

DEFINITION 4. *The packet loss at a receiver is the fraction of transmitted packets that are either not received or received in error at the receiver. The system loss is the sum of the packet losses at all the receivers in the multicast group.*

DEFINITION 5. *A system is said to be stable if the mean queue length is bounded. Further, a transmission policy that stabilizes the system is called a stable policy.*

Note that for any stable policy the packet departure rate is equal to the arrival rate λ.

DEFINITION 6. *A stable transmission policy Δ is called ϵ-throughput optimal if no other stable transmission policy can achieve throughput more than ϵ plus the throughput under Δ.*

DEFINITION 7. *The busy slots are the slots in which the sender's queue is non-empty.*

DEFINITION 8. *A policy Δ belongs to the class of generalized threshold policies, if it sets threshold $T(t) \in \{0, \ldots, G+1\}$ in every busy slot t based on arbitrary rules and then transmits a packet only when the sender and $T(t)$ or more receivers are ready. The threshold may be selected based on past, present and future arrivals and readiness states.*

We note that for any transmission policy $\widehat{\Delta}$, there exists a generalized threshold policy that transmits in the same slots as $\widehat{\Delta}$. This can be seen as follows. Let $\widehat{\Delta}$, using certain rules, select slots t_1, $t_2, \ldots$ in which it transmits. Consider a generalized threshold policy Δ that computes slots t_1, $t_2, \ldots$ using the same rule as $\widehat{\Delta}$ and sets threshold 0 in these slots. In the remaining busy slots, Δ sets threshold $G+1$. Thus, $\widehat{\Delta}$ and Δ transmit in the same slots. Hence, it is sufficient to consider only generalized threshold policies.

In the following lemma, we provide a sufficient condition for a generalized threshold policy to be ϵ-throughput optimal. Let $s_T^{\Delta}(t)$ denote the number of busy slots in which threshold T is chosen till time t under a generalized threshold policy Δ. Note that it is not necessary to select a threshold when queue length is zero, as a packet cannot be transmitted in this case.

LEMMA 1. *For any $\epsilon > 0$, a stable generalized threshold policy Δ is ϵ-throughput optimal w.p. 1 if the following condition holds for some threshold $T \in \{0, \ldots, G\}$.*

$$\lim_{t\to\infty} \frac{s_T^{\Delta}(t) + s_{T+1}^{\Delta}(t)}{t} \geq 1 - \frac{\epsilon}{2G} \quad \textit{w.p. 1.} \tag{1}$$

Lemma 1 does not show how to design an ϵ-throughput optimal policy. Nevertheless it is a useful tool as it provides a sufficient condition to establish the ϵ-throughput optimality of a stable generalized threshold policy. The utility of the lemma is similar to that of a lyapunov function. Recall that a sufficient condition for a policy to be stable is the existence of a lyapunov function with negative drift. But this sufficient condition does not in general show how to design a stable policy. In the next section, we design an adaptive transmission policy that satisfies condition (1) and hence is ϵ-throughput optimal.

3. THROUGHPUT OPTIMAL TRANSMISSION POLICY ($\Delta_O(\Gamma)$)

We describe a parameterized transmission policy $\Delta_O(\Gamma)$, that we prove to be ϵ-throughput optimal. The policy selects a threshold value based on the queue length at the sender in each slot. A packet is transmitted if (a) the sender is ready to transmit, (b) the number of ready receivers is greater than or equal to the threshold and (c) the sender has a packet to transmit. The threshold values are selected as follows. Let Q denote the queue length at the sender and let Γ be some fixed positive integer. For $1 \leq T \leq G$, the threshold is T if $(G-T)\Gamma < Q \leq (G-T+1)\Gamma$ and threshold is 0 if $Q > G\Gamma$. Thus, the threshold value increases with decrease in queue length. The policy does not select a threshold when queue length is zero.

We show that $\Delta_O(\Gamma)$ is ϵ-throughput optimal under some additional assumptions on the readiness process. We assume that the readiness process is an irreducible, aperiodic and time homogeneous Markov Chain (MC) with arbitrary transition probabilities. The state of the MC is the $G+1$ dimensional vector $\vec{j} = [j_0\ j_1\ \ldots\ j_G]$ that represents the readiness state. Note that the MC has a finite number of states, since $j_l \in \{0,1\}$ for every $l \in \{0, \ldots, G\}$. Since we do not impose any restriction on the transition probabilities of the markov chain, the chain can capture the correlations of the sender's and the receiver's readiness states in the same and different time slots. Figure 3 shows how such correlations arise in practice. Let b_u^R denote the unique steady state probability that the sender is ready to transmit and u receivers are ready to receive. Let $n_u(t)$ denote the number of slots till time t in which the sender and u receivers are ready. Now, by stationarity and ergodicity of the readiness process

$$\lim_{t\to\infty} \frac{n_u(t)}{t} = b_u^R \text{ w.p. 1.} \tag{2}$$

In general, from ergodicity we cannot conclude anything about the rate of convergence of the empirical distribution $\lim_{t\to\infty} \frac{n_u(t)}{t}$ to the stationary distribution b_u^R. But for finite, aperiodic MC's, empirical distribution converges to the steady state distribution exponentially fast [3, 21]. We use this exponential convergence to prove the optimality of $\Delta_O(\Gamma)$. The optimality of $\Delta_O(\Gamma)$ holds for any stationary and ergodic stochastic process that has the exponential convergence property.

The throughput of policy $\Delta_O(\Gamma)$ is denoted as $\Omega^{\Delta_O(\Gamma)}$. Let the maximum throughput attained by a stable policy be Ω_{opt}. We prove the following result in the appendix.

THEOREM 1. *If the arrival rate λ is less than the steady state probability that the sender is ready ($\sum_{u=0}^{G} b_u^R$), then for any given $\epsilon > 0$ there exists Γ_O such that $\Delta_O(\Gamma)$ is ϵ-throughput optimal for every $\Gamma \geq \Gamma_O$. Formally, $\Omega_{opt} - \Omega^{\Delta_O(\Gamma)} \leq \epsilon$ w.p. 1. Further, no policy is stable if $\lambda > \sum_{u=0}^{G} b_u^R$.*

Note that the above result implies that any stable off-line policy that takes transmission decisions based on the knowledge of past, present and future arrivals and readiness states can not attain throughput more than $\Omega^{\Delta_O(\Gamma)}$. This holds even though $\Delta_O(\Gamma)$ takes transmission decisions based on only the current packet availability and the current number of ready receivers.

The intuition behind the result is as follows. Consider a policy that selects the same threshold in every slot. The expected reward is a monotonically increasing function of the threshold. Hence a throughput optimal policy should select the largest threshold that stabilizes the system. This threshold T_O must satisfy

$$\sum_{u=T_O+1}^{G} b_u^R < \lambda < \sum_{u=T_O}^{G} b_u^R. \tag{3}$$

The throughput can be further improved by appropriately randomizing between the threshold values T_O and T_O+1. The randomization should be such that the system remains stable. This is the basic idea behind the design of the static optimum policy [6]. Intuitively an adaptive optimum policy should select the thresholds T_O and T_O+1 most of the time. The difficulty, however, is that λ and b_u^R's are not known, and thus T_O cannot be computed.

Now, we explain why $\Delta_O(\Gamma)$ will select the thresholds T_O and T_O+1 most of the time. From (3), the rate at which slots with m or more ready receivers arrive is more than the packet arrival rate λ, for every $m \leq T_O$. On the other hand, for $m \geq T_O + 1$ the rate at which the samples with m or more ready receivers arrive is smaller than λ. This implies that for threshold values greater than or equal to $T_O + 1$, i.e., when $Q \leq (G - T_O)\Gamma$, the queue length process has a positive drift and as a result the queue length increases, and consequently the threshold decreases. On the other hand for threshold values less than or equal to T_O, i.e., when $Q > (G - T_O)\Gamma$, the queue length process has a negative drift and hence queue length decreases, and the threshold increases. Hence we observe that when Γ is large enough the thresholds T_O and $T_O + 1$ are selected most of the time.

Recall that a packet is lost at a receiver if the receiver is not ready at the time of transmission. The MAC protocol we propose may transmit a packet even when some of the receivers are not ready, and is therefore unreliable. But, wireless is an inherently unreliable medium. Thus, it is a standard practice to use a reliable transport layer strategy to retrieve the information lost at the MAC layer. Several existing MAC strategies for multicast in adhoc networks, like *broadcast based multicast* and *threshold-1 multicast* are unreliable as well. Fortunately, several reliable transport layer schemes have been proposed specifically for wireless multicast transmissions, which can be used in conjunction with any MAC layer strategy [4, 17]. But, the efficiency of these schemes is severely impaired when the packet loss at the MAC layer is high. Our focus is to minimize the packet loss subject to resource limitations in the network. Now, there would not be any loss if a packet is transmitted only when all the receivers are ready, but then as discussed before, the system may become unstable. Note that stability is essential as otherwise the queue lengths at the sender would be unbounded leading to unbounded delays. Thus our objective is to use a transmission policy that minimizes the packet loss among all stable policies. The next theorem shows that $\Delta_O(\Gamma)$ achieves this objective.

THEOREM 2. *If $\Delta_O(\Gamma)$ is ($\lambda\epsilon$)-throughput optimal, then no stable policy can achieve loss smaller than the loss under $\Delta_O(\Gamma)$ minus ϵ for any given $\epsilon > 0$.*

Assume that the system is stable. The system loss is $G - R$, where R is the average reward received by the policy per packet. Thus, we need to maximize the mean reward R in order to minimize the system loss. The throughput of a transmission policy is λR. Thus, a throughput optimal policy maximizes R and therefore minimizes the system loss subject to system stability. Since $\Delta_O(\Gamma)$ is throughput optimal, it minimizes the loss.

From Theorem 2, if the system loss for $\Delta_O(\Gamma)$ is more than that the system can tolerate, then the required loss constraint can not be guaranteed by any stable transmission policy. Since stability is essential, the resources available in this case are not enough to deliver the required QoS, and other measures such as admission control must be resorted to. This is beyond the scope of this paper. Henceforth we do not consider loss explicitly.

The value of Γ_O depends on the system parameters. However, since optimality of $\Delta_O(\Gamma)$ is guaranteed for all large Γ, only a rough estimate of Γ_O is necessary. Simulations show that the optimality is attained for modest values of Γ_O.

If the queue length is greater than $G\Gamma$, then $\Delta_O(\Gamma)$ has threshold 0, and can therefore transmit a packet even when no receiver is ready; but, in this case, the transmission is useless. So, we consider a policy $\widehat{\Delta}_O(\Gamma)$, that is similar to $\Delta_O(\Gamma)$, except that it transmits only when at least one receiver is ready. Now, $\widehat{\Delta}_O(\Gamma)$ selects threshold T if $(G - T)\Gamma < Q \leq (G - T + 1)\Gamma$, for $T > 1$, and threshold 1 if $Q > (G - 1)\Gamma$. Now, $\widehat{\Delta}_O(\Gamma)$'s and $\Delta_O(\Gamma)$'s transmission rules are the same once the thresholds are selected.

DEFINITION 9. *$\mathcal{C}$ is a class of policies that transmit a packet only when the sender and at least one receiver is ready, and the sender has a packet to transmit.*

Note that the broadcast policy is not in $\mathcal{C}$.

THEOREM 3. *If $\lambda < \sum_{u=1}^{G} b_u^R$, then for any given $\epsilon > 0$ there exists $\widehat{\Gamma}_O$ such that $\widehat{\Delta}_O(\Gamma)$ is ϵ-throughput optimal for every $\Gamma > \widehat{\Gamma}_O$. Furthermore, if $\lambda > \sum_{u=1}^{G} b_u^R$, then no policy in $\mathcal{C}$ is stable.*

The intuition for Theorem 3 is similar to that for Theorem 1.

The policy $\Delta_O(\Gamma)$ and $\widehat{\Delta}_O(\Gamma)$ provide computationally simple transmission rules. The sender only needs the number of ready receivers and need not know which particular receivers are ready. This simplifies the protocol design problem.

4. MAC LAYER PROTOCOL

The optimal decision rule is based on the sender's queue length, its readiness state, and the number of ready receivers. The sender is ready if it is not backing off, and none of its neighbors is receiving a packet. We explore various possible approaches to inform the sender about the number of ready receivers and transmissions in its neighborhood.

First we present the system challenges experienced by the existing IEEE 802.11 hand-shakes in case of multiple receivers. Recall that IEEE 802.11 uses RTS-CTS-DATA-ACK handshake for communication. The handshake works well if the sender transmits to only one receiver. If all the nodes in a multicast group send CTS simultaneously in response to a RTS from a sender, then these CTS messages will collide at the sender. Hence, the sender will not know whether the receivers are ready. If the sender knew that the colliding messages were CTS responses from the receivers, then the sender can infer the number of ready receivers by measuring the power. Power is additive and power information is not destroyed even during collision. But the power measurement does not solve the problem entirely because of the following reasons. The CTS message conveys the duration of the data transfer, so that the neighboring nodes can defer their transmissions. This is no longer possible if CTS collides and hence data packets collide with other transmissions in the neighborhood. Similar problems exist for ACK transmission.

We propose to use a busy-tone based scheme [28]. The available bandwidth is divided into two channels (a) a message channel and (b) a busy-tone channel. A sender initiates a data transfer by sending an RTS message on the message channel. After receiving RTS, all the receivers in the multicast group that are ready to receive send a busy-tone. This busy-tone acts as a CTS message. A sender estimates the number of ready receivers by measuring the power of the busy-tone signal. The total received power can be measured using standard circuits [18]. If this estimate is greater than a threshold value, the sender transmits the data packet. Receivers transmit the busy tone until the packet transmission is complete. Busy-tone does not interfere with the data transmission as both are on different frequency bands. Neighboring nodes defer their transmission till the

```
Procedure MAC_at_Sender()
begin
  /* CW refers to the current value of contention window */
  /* CW_min and CW_max refer to the minimum and the maximum values
  for CW respectively*/
  When a packet arrives from a higher layer, then do the following
  Enqueue the packet in the MAC layer buffer;
  queue_length = queue_length + 1;
  Compute threshold T based on the queue length as described in Section 3;
  if (queue_length = 1) then
    /*if the sender's queue was empty before the arrival*/
    CW = CW_min;
    Choose an integer I uniformly from {1, ..., CW};
    Set back-off timer for time duration I × slot_length;

  When the back-off timer expires do the following
  if (data or busy tone channel is not idle) then
    /* the sender is not ready */
    CW = min{CW + 1, CW_max};
    Choose an integer I uniformly from {1, ..., CW};
    Set back-off timer for time duration I × slot_length;
  else
    /* the sender is ready */
    Send RTS packet;
    Measure the power on busy tone channel;
    Estimate the number of ready receivers based on the measured busy
    tone power;
    if (The estimate < T) then
      CW = min{CW + 1, CW_max};
      Choose an integer I uniformly from {1, ..., CW};
      Set back-off timer for time duration I × slot_length;
    else
      Transmit data packet;
      queue_length = queue_length - 1;
      Compute threshold T based on the queue_length as described in
      Section 3;
      if (queue_length > 0) then
        CW = CW_min;
        Choose an integer I uniformly from {1, ..., CW};
        Set back-off timer for time duration I × slot_length;
end
```

Figure 4: Pseudo code describes the busy tone based MAC layer protocol at the sender.

```
Procedure MAC_at_Receiver()
begin
  When RTS intended for the receiver is received do the following
  if (Busy tone and data channels are idle) then
    Put up a busy tone and wait for data packet;
    Set RcvTimer for the duration equals to four times maximum propa-
    gation delay;
    if (Data packet begins to arrive before RcvTimer expires) then
      Receive the data packet and pass it to the higher layer;
      Stop the busy tone;
    else
      Stop the busy tone;
end
```

Figure 5: Pseudo code describes the busy tone based MAC layer protocol at the receiver.

busy-tone stops. If the decision is not to transmit, then the sender transmits a release signal and backs-off for a random interval. The back-off intervals are independent and uniformly distributed. The receivers stop emitting the busy tone once they receive the release signal. Refer to Figures 4 and 5 for pseudo code.

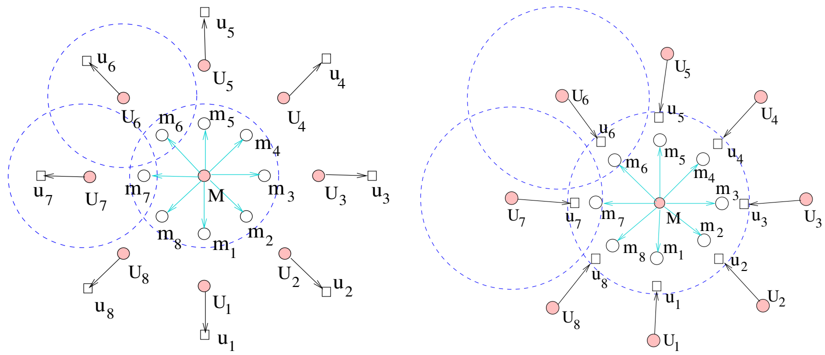
The sender can infer the number of ready receivers from the received busy-tone power only if the received power levels are the same for all the receivers. This is not possible without any power regulation as the receivers are at different distances from the sender ("near-far effect"). If all the receivers know their distances from the sender, then they can regulate the power so that the busy tones from all the receivers reach the sender at the same power level. The receivers can estimate their distances from the power level of the received RTS, assuming that all the senders transmit at the same power.

By providing unambiguous and instantaneous feedback, the busy tone scheme solves the problems originating from CTS and ACK collisions in IEEE 802.11 based multicast schemes. The limitations of the busy tone scheme are the following. (1) The scheme is not totally compatible with IEEE 802.11, though it works on the same philosophy. Hence the implementation of this scheme will need protocol changes. But some other recent papers have advocated busy tone based schemes as well [9, 14, 22]. (2) Accurate power control and power measurement are difficult in practice. Hence, the sender's estimate of the number of ready receivers may not be accurate. Using simulations, we have verified that the proposed optimal policy is robust, i.e., its throughput is close to optimal even in presence of the estimation errors (Figure 9). Further, future extensions for IEEE 802.11 are likely to support power control and power measurement functionalities [1]. We have also proposed an augmentation of IEEE 802.11 protocol to implement $\Delta_O(\Gamma)$ [7]. The MAC protocol in [7] does not need any power measurement and control functionalities. (3) The receivers' power consumption increase due to the transmission of busy tone throughout the data transmission. (4) The receivers need to transmit a busy tone signal while receiving the data. Thus, two radios are required at the receivers. Most of the current wireless appliances have a single radio and hence continuous transmission of the busy tone while receiving data may not be feasible. The problems (3) and (4) can be addressed with an additional busy tone channel [9]. On one busy tone channel, receivers can send a short burst of signal indicating CTS message, and on the other busy tone channel receivers can send a short burst indicating ACK. All the nodes in the neighborhood will not initiate any communication till they receive ACK busy tone for every CTS busy tone they received.

5. SIMULATION RESULTS AND DISCUSSION

We have proved that $\Delta_O(\Gamma)$ is ϵ-throughput optimal, when the readiness states are Markovian. We examine the application layer throughput in a wireless network with several unicast and multicast sessions when every sender uses $\Delta_O(\Gamma)$, and the readiness states are generated due to packet transmissions. The simulation results substantiate the claim that $\Delta_O(\Gamma)$ attains significantly higher throughput than the other existing policies.

Using *ns-2*, we simulate the performance of Δ_O and the other existing multicast policies like broadcast based multicast, unicast based multicast and threshold-1 multicast. In [2], we describe how



(a) Topology 1 (b) Topology 2

Figure 6: We evaluate the the performance of various multicast strategies in Topologies 1 and 2. Each topology has a multicast session with sender M and 8 receivers m_1 to m_8 and 8 unicast sessions. Unicast session i has sender U_i and receiver u_i, $1 \leq i \leq 8$. The difference between the topologies 1 and 2 is that the unicast receivers u_i's are not in the transmission range of M in topology 1, while they are in the transmission range of M in topology 2.

we implement these policies in *ns-2*. We have not compared the performance of $\Delta_O(\Gamma)$ with that of the MAC policies proposed for wireless LANs, e.g. [12], because they can only be used in the scenario where a single node is in the transmission range of all other nodes in the network. This assumption does not hold in adhoc networks.

5.1 Simulation Scenario

We use UDP at the transport layer. We do not use TCP, as the interaction between TCP and wireless MAC is not well understood and hence the topic of research even for unicast sessions [10]. We measure the throughput of a receiver as the number of packets it receives successfully per unit time, and the throughput of a session as the sum of the throughputs of its receivers. We consider a time interval of 5000 seconds. The choice of the channel capacity C scales the throughput of all the policies by a factor of C. We select $C = 1$ Mbps. The RTS packet has 44 bytes. The length of a slot is 20 μs and the maximum propagation delay is 2 μs. We use the busy tone based approach described in Section 4 (Figure 4 and 5) to implement $\Delta_O(\Gamma)$ and threshold-1 multicast.

We consider sample topologies shown in Figure 6. In both the topologies, we assume that the unicast sender U_i generates packets at rate λ_U and the multicast sender M generates packets at rate λ_M. The packet arrival process is Poisson. The packets arriving at M have length 552 bytes, while those arriving at U_i have length 53 bytes. For larger packet sizes at unicast sessions the results differ only in magnitude, but the trends remain the same. Further, stability region of the multicast session is small when the packet size for unicast sessions is large. We, however, study the impact of high network load on the multicast session by increasing λ_U (Figures 10 and 11).

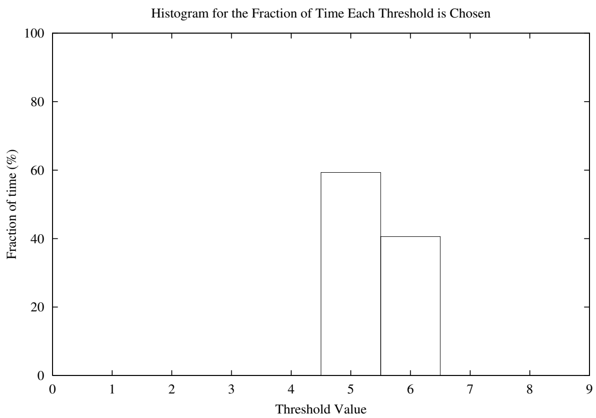
Now we discuss how the packet transmissions generate readiness states in topologies 1 and 2. In topologies 1 and 2, for every $i \in \{1, \dots, 8\}$, U_i is ready when it is not backing off and m_i is not generating busy tone. Also, m_i is ready when U_i is not transmitting a packet to u_i. In topology 1, M is not ready when it backs off, while u_i is always ready. In topology 2, M is not ready and when some u_i is generating busy tone. Further, none of the unicast receivers are ready when M is transmitting.

5.2 Discussion on Simulation Results

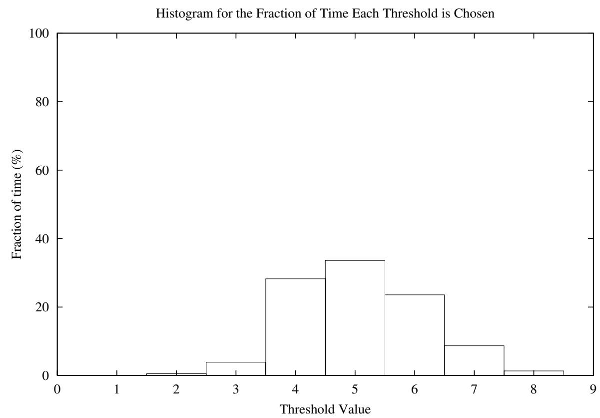
We have proved that any stable policy that selects any two consecutive thresholds T and $T + 1$ $(0 \leq T \leq G)$ most of the time maximizes throughput as long as the readiness states are ergodic (Lemma 1). The readiness states are likely to be stationary and ergodic even when they are generated by packet transmissions. We needed the additional assumption that the readiness states are Markovian to show that $\Delta_O(\Gamma)$ chooses two consecutive thresholds most of the time. Figure 7(a) shows that $\Delta_O(\Gamma)$ selects two consecutive thresholds most of the time even when the readiness states are generated by packet transmissions. This validates the optimality result. Now, Theorem 1 guarantees that $\Delta_O(\Gamma)$ would select two consecutive thresholds most of the time only when Γ is sufficiently large. The observation in Figure 7(b) that $\Delta_O(\Gamma)$ frequently selects more than three thresholds for low values of Γ further validates the theorem. Also, Figure 8 shows that as Γ increases the throughput of the multicast session under $\Delta_O(\Gamma)$ converges to optimum value.

Now, we compare the throughput of the multicast session in topologies 1 and 2 under $\Delta_O(\Gamma)$ with that under broadcast based multicast, threshold-1 multicast and unicast based multicast policies. We observe that Δ_O achieves much higher throughput than the other existing policies in both the topologies (Figures 10 and 11). We next explain the performance difference.

The broadcast based multicast scheme does not exchange any hand shake messages, and thus causes frequent data packet collisions. Thus, the reward per packet is low resulting in low system throughput. Threshold-1 policy exchanges hand-shake messages and avoids the data packet collisions. As a result, this policy provides much better throughput than broadcast based multicast policy. The limitation of this scheme is that the threshold is always 1, and hence the policy may transmit even when only a few receivers are ready. The policy $\Delta_O(\Gamma)$ outperforms these policies as it prevents data packet collisions by exchanging hand-shake massages,



(a) $\Gamma = 250$



(b) $\Gamma = 20$

Figure 7: Figures (a) and (b) plot the fraction of time each threshold is chosen by $\Delta_O(\Gamma)$ in topology 1 for Γ equal to 250 and 20, respectively. Here, λ_M = 50 packets/sec and λ_U = 500 packets/sec.

and it also prevents transmission when only a few receivers are ready by choosing an appropriate threshold value. The unicast based multicast policy uses separate transmissions to reach different members of the multicast group even when they can be reached using a single transmission. Hence, the total number of packets transmitted under this policy is much smaller than that under other policies. This results in low throughput.

From Figures 10(b) and 11(b), we observe that the throughput gain of $\Delta_O(\Gamma)$ over the unicast based multicast policy increases with increase in λ_M. This is because the stability region of the unicast policy is small, and hence the throughput of the policy saturates even for small values of λ_M (Figures 10(a) and 11(a)). On the other hand, $\Delta_O(\Gamma)$ has large stability region. Now, $\Delta_O(\Gamma)$ transmits λ_M packets per second in its stability region, and this number number increases with increase in λ_M. As λ_M increases the threshold chosen by $\Delta_O(\Gamma)$ may decrease in order to maintain the system stability. Hence the reward per packet under $\Delta_O(\Gamma)$ may reduce. But this decrease is compensated by the increase in the number of packets transmitted per unit time, which also explains the non-linear increase in throughput for $\Delta_O(\Gamma)$ as λ_M increases (Figures 10(a) and 11(a)).

Now, the throughput gain of $\Delta_O(\Gamma)$ over threshold-1 and broadcast based multicast policies decreases with the increase in λ_M. This is because as λ_M increases the thresholds chosen by $\Delta_O(\Gamma)$ decreases, becomes closer to 1. Thus the difference between the throughput of $\Delta_O(\Gamma)$ and threshold-1 and broadcast based multicast policy decreases. Hence the gain also decreases.

From Figures 10(e) and 11(e), we observe that as λ_U increases the throughput gain of $\Delta_O(\Gamma)$ over all the existing policies increases. This is because for small values of λ_U, the fraction of slots in which the unicast senders transmit a packet is small. Hence the multicast receivers are ready most of the time resulting in a large reward per transmission under all the policies. Thus, the choice of threshold does not affect the throughput. But for higher values of λ_U, large number of multicast receivers are ready only in a small fraction of slots. Unlike other policies $\Delta_O(\Gamma)$ transmits only in these slots due to an appropriate selection of threshold. Hence the $\Delta_O(\Gamma)$'s throughput gain increases with increase in λ_U.

We also study the throughput of $\Delta_O(\Gamma)$ in topology 1, when M makes errors in estimating the number of ready receivers. We consider binomial errors and plot $\Delta_O(\Gamma)$'s throughput for various values of the variance v in Figure 9. Simulation results show that the throughput of multicast session under $\Delta_O(\Gamma)$ decreases only marginally when the estimate is erroneous. The difference between $\Delta_O(\Gamma)$'s throughput without estimation errors and with estimation errors increases as v increases. Furthermore, the decreased throughput is still significantly higher than that of the other existing policies (Figure 9).

Figures 10(c) and 11(c) show that in both the topologies the throughput of unicast sessions is similar to or higher than that under any other the policy. Thus, $\Delta_O(\Gamma)$ increases the throughput of the multicast session by sending more packets when the unicast sessions are not transmitting and not by decreasing the throughput of the unicast sessions.

Now, we explain why the difference between the throughputs under threshold-1 policy and $\Delta_O(\Gamma)$ is small for all values of λ_M in topology 2 (Figure 11(a)). Recall that in topology 2, M and m_i are not ready when u_i is generating a busy tone. Thus, when M is ready, all the multicast receivers are also ready. Hence, the reward per packet should be 8 irrespective of the choice of threshold. In other words, $\Delta_O(\Gamma)$ and threshold-1 multicast policy should achieve the same throughput. The throughput values differ by a small amount because of the collision of RTS messages from U_i and M at m_i. In this case, m_i will not put a busy tone, and it will not receive the data packet even if M decides to transmit. Since $\Delta_O(\Gamma)$ chooses an appropriate threshold value, it may not transmit when m_i does not put up a busy tone, and thus unlike threshold-1 multicast policy which always has threshold 1, avoids packet loss at m_i. We note that this is an extreme case, and in general $\Delta_O(\Gamma)$ provides a significant throughput gain over threshold-1 multicast, e.g. Figure 10(b).

6. DISCUSSION

We have designed a policy $\Delta_O(\Gamma)$ which is throughput optimal constrained to stability. The policy $\Delta_O(\Gamma)$ is adaptive, and takes transmission decisions based only on the queue length at the sender and the receiver readiness states in the current slot. Hence, $\Delta_O(\Gamma)$ can be implemented in distributed settings. The optimality of $\Delta_O(\Gamma)$ also holds under the following more general scenarios than that considered here.

- The analytical framework can be generalized to allow three or more readiness states for each receiver. Optimality of $\Delta_O(\Gamma)$ holds even when the readiness process is an irreducible, aperiodic, time-homogeneous discrete time markov chain with L states where the lth state is $[j_{l,1}, \ldots, j_{l,G}]$, $j_{l,i}$ is the probability of error-free reception of a packet at the ith receiver in the lth state. Recall that earlier $j_{l,i} \in \{0, 1\}$. Thus, the optimality of $\Delta_O(\Gamma)$ holds under a more general fading model.

- We have considered packets of unit lengths. The optimality of $\Delta_O(\Gamma)$ holds for iid packet lengths.

- Lemma 1 holds for any stationary and ergodic receiver readiness process (not necessarily markovian). Also, Theorems 1 and 3 hold for any stationary and ergodic readiness process for which the empirical distribution converges to the stationary distribution at exponential rate.

Now, let us examine the features that are not captured by the analytical model proposed here. In the analytical model, we assume that the receiver readiness process cannot be controlled, and then the throughput for a multicast session is maximized constrained to stability under the given readiness process. This approach is justified if the senders do not coordinate with each other explicitly to decide transmission schedule. The following example shows that by appropriately coordinating the transmissions, it is possible to control the receiver readiness process and thereby achieve better throughput.

Example: Consider topology 1 in Figure 6(a). Let arrival rate at a unicast sender U_i be $1/2 - \delta$ for every $i \in \{1, \ldots, 8\}$ and for small $\delta > 0$. Also, let arrival rate at the multicast sender M be $1/2 - \delta$. Note that simultaneous transmissions from all the unicast senders can be received correctly at their respective receivers. In this scenario, the throughput is maximized if U_i's coordinate among each other so as to transmit only in odd slots, i.e., in slots $\{1, 3, \ldots, \}$ and M transmits only in even slots, i.e., in slots $\{2, 4, \ldots, \}$. Here, M's throughput is $G(1/2 - \delta)$. Observe that M's throughput may be less than $G(1/2 - \delta)$ if U_i's do not coordinate, e.g., if U_i transmits in slots $i + 8k$ for each $k = 0, 1, \ldots$ and $i = 1, 2, \ldots, 8$, then in most of the slots only $G - 1$ (and not G) recievers will be ready. Thus, M's throughput will be $(G - 1)(1/2 - \delta)$ approximately.

Designing a coordination among different senders typically leads to centralized scheduling schemes. For example, Tassiulas *et al.* have proposed a centralized algorithm for determining the optimal coordination for maximizing throughput in unicast wireless networks [27]. Since we are interested in a distributed implementation, we do not consider explicit coordination among senders.

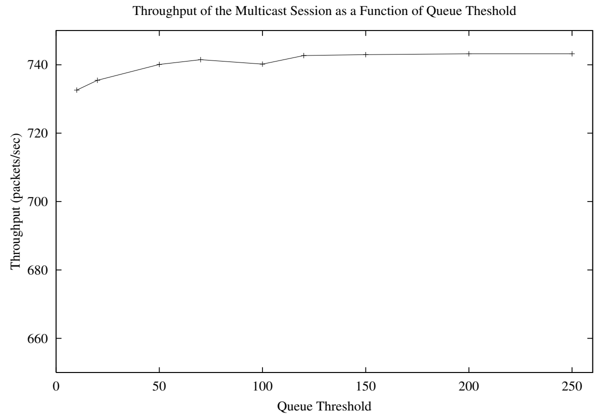


Figure 8: We plot the throughput of the multicast session under $\Delta_O(\Gamma)$ as a function of Γ in topology 1. Here, $\lambda_M = 100$ packets/sec and $\lambda_U = 300$ packets/sec. The parameter Γ is referred as *queue threshold* in the figure.

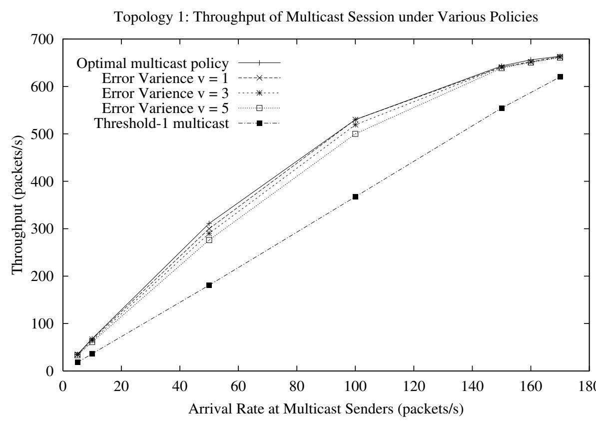


Figure 9: We plot the performance of $\Delta_O(\Gamma)$ when the multicast sender M makes random error E_t in estimating the number of ready receivers at time t. We assume that $|E_t|$ is binomial random variable with mean 0 and variance v, and E_t is positive with probability 1/2.

7. CONCLUSION

Maximizing the performance metrics in wireless multicast presents challenges that are not encountered in wireless unicast or wireline multicast networks. For example, a transmission policy that maximizes the stability region of the network need not maximize the network throughput. The objective therefore is to maximize throughput subject to attaining system stability. We consider a single multicast session, and model its interactions with the rest of the network using stochastic readiness states. We present a sufficient condition that can be used to establish the throughput optimality of a stable transmission policy. We subsequently design an adaptive stable policy that allows a sender to decide when to transmit using simple computations based only on its local information such as its queue length, its readiness state and the number of ready receivers, and without using any information about the network statistics. The proposed policy attains the same throughput as the optimal offline stable policy that uses in its decision process past, present, and even future arrivals and readiness states. We prove the throughput optimality of the proposed policy using the sufficient condition and the large deviation results. We present a MAC protocol for acquiring the local information necessary for executing this policy, and implement it in *ns-2*. Simulations demonstrate that the optimal strategy significantly outperforms the existing approaches in adhoc networks consisting of several multicast and unicast sessions.

We hope that the performance improvement obtained by the proposed policy and the intuition gained in its design would stimulate further research in this area. Some of the open problems are: (a) coordinating the transmissions of different nodes so as to maximize the performance metrics like throughput, (b) the interaction between the proposed MAC strategy and the reliable transport layer multicast protocols and (c) optimizing the performance in presence of mobility, dynamic group membership changes, security concerns, etc. We plan to address these problems in future.

8. REFERENCES

[1] IEEE 802.11 WG, draft supplement to standard for telecommunications and information exchange between systems - Part 11: Medium Access Control (MAC) and physical layer (PHY) specifications: Spectrum and transmit power management extensions in the 5Ghz band in Europe, March 2003.

[2] A. Bhat. *NS-2* simulation and comparison of various wireless mac multicast techniques and protocols. Master's thesis, University of Pennsylvania, 2004.

[3] J. A. Bucklew. *Large Deviation Techniques in Decision, Simulation, and Estimation*. Wiley-Interscience, 1990.

[4] R. Chandra, V. Ramasubramaniam, and P. Birman. Anonymous gossip: Improving multicast reliability in mobile ad-hoc networks. In *21st International Conf on Distr Computing Systems*, 2001.

[5] P. Chaporkar, A. Bhat, and S. Sarkar. An adaptive strategy for maximizing throughput in MAC layer wireless multicast. Technical report, University of Pennsylvania, 2004.

[6] P. Chaporkar and S. Sarkar. Stochastic control techniques for throughput optimal wireless multicast. In *Control and Decision Conference (CDC), 2003*, Maui, Hawaii, Dec 2003.

[7] P. Chaporkar and S. Sarkar. On-line optimal wireless multicast. In *WiOpt 2004*, Univ. of Cambridge, UK, Mar 2004.

[8] C. C. Chiang, M. Gerla, and L. Zhang. Forwarding group multicast protocol (FGMP) for multihop, mobile wireless networks. *Baltzer Cluster Computing*, 1(2):187–196, 1998.

[9] J. Deng and Z. Haas. Dual busy tone multiple access (DBTMA): A new medium access control for packet radio networks. In *IEEE ICUPC*, 1998.

[10] Z. Fu, P. Zerfos, H. Luo, S. Lu, L. Zhang, and M. Gerla. The impact of multihop wireless channel on TCP throughput and loss. In *INFOCOM'03*, San Francisco, California, 2003.

[11] C. Jaikaeo and C. Shen. Multicast communication in ad hoc networks with directional antennas. In *ICCCN 2003*, Dallas, Texas, 2003.

[12] J. Kuri and S. Kasera. Reliable multicast in multi-access wireless LANs. In *INFOCOM'99*, 1999.

[13] S. J. Lee, M. Gerla, and C. C. Chiang. On-demand multicast routing protocol. In *Proceedings of IEEE WCNC'99*, 1999.

[14] J. P. Monks, V. Bharghavan, and W. W. Hwu. A power controlled multiple access protocol for wireless packet networks. In *Proceedings of INFOCOM'01*, 2001.

[15] M. Nagy and S. Singh. Multicast scheduling algorithms in mobile networks. *Cluster Computing*, 1998.

[16] T. Ozaki, J. B. Kim, and T. Suda. Bandwidth-efficient multicast routing for multihop, ad-hoc wireless networks. In *INFOCOM'01*, 2001.

[17] E. Pagani and G. Rossi. Reliable broadcast in mobile multihop packet networks. In *MOBICOM'97*, pages 34–42, 1997.

[18] J. G. Proakis. *Wireless Communication*. McGraw-Hill, 2000.

[19] E. M. Royer and C. E. Perkins. Multicast operation of the ad-hoc on-demand distance vector routing protocol. In *Proceedings of ACM/IEEE MOBICOM'99*, 1999.

[20] S. Sarkar and L. Tassiulas. A framework for routing and congestion control for multicast information flows. *IEEE Trans. on Information Theory*, 48(10):2690–2708, Oct 2002.

[21] A. Shwartz and A. Weiss. *Large Deviations for Performance Analysis: Queues, communications, and computing*. CRC Press, 1995.

[22] S. Singh, C. S. Raghavendra, and J. Stepanek. Power efficient broadcasting in mobile ad hoc networks. In *PIMRC'99*, 1999.

[23] P. Sinha, R. Shivkumar, and V. Bharghavan. MCEDAR: Multicast core-extraction distributed ad hoc routing. In *Proceedings of IEEE WCNC'99*, 1999.

[24] K. Tang and M. Gerla. MAC layer broadcast support in 802.11 wireless networks. In *Proceedings of IEEE MILCOM 2000*, 2000.

[25] K. Tang and M. Gerla. Random access MAC for efficient broadcast support in ad hoc networks. In *Proceedings of IEEE WCNC 2000*, 2000.

[26] K. Tang and M. Gerla. MAC reliable broadcast in ad hoc networks. In *Proceedings of IEEE MILCOM'01*, 2001.

[27] L. Tassiulas and A. Ephremides. Stability properties of constrained queueing systems and scheduling for maximum throughput in multihop radio networks. *IEEE Trans. on Information Theory*, 37(12):1936–1949, Dec 1992.

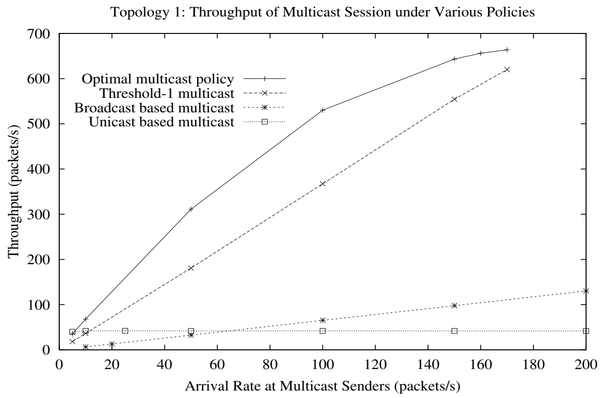
[28] F. A. Tobagi and L. Kleinrock. Packet switching in radio channels: Part II- the hidden terminal problem in carrier sense multiple-access and the busy-tone solution. *IEEE Trans. on Communications*, 23(12), 1975.

[29] P.-J. Wan, G. Calinescu, X. Li, and O. Frieder. Minimum-energy broadcast routing in static ad hoc wireless networks. In *INFOCOM'01*, 2001.

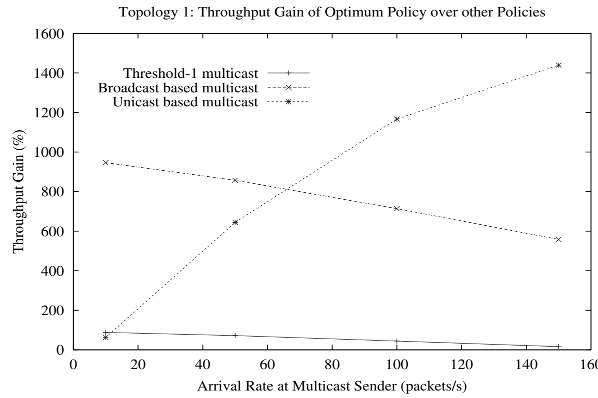
[30] K. Wang, V. Srinivasan, C. Chiasserini, and R. Rao. Joint scheduling and power control for multicasting in wireless ad hoc networks. In *IEEE VTC Fall 2003*, Orlando, FL, 2003.

[31] J. Wieselthier, G. Nguyen, and A. Ephermides. On the construction of energy-efficient broadcast and multicast trees in wireless networks. In *INFOCOM'00*, 2000.

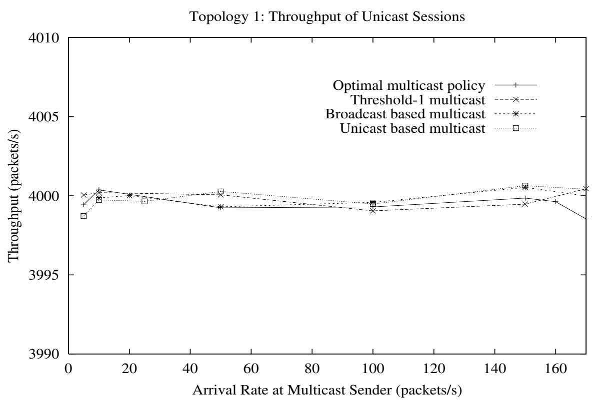
[32] H. Zhou and S. Singh. Content-based multicast for mobile ad hoc networks. In *Proc. MobiHoc 2000*, August 2000.



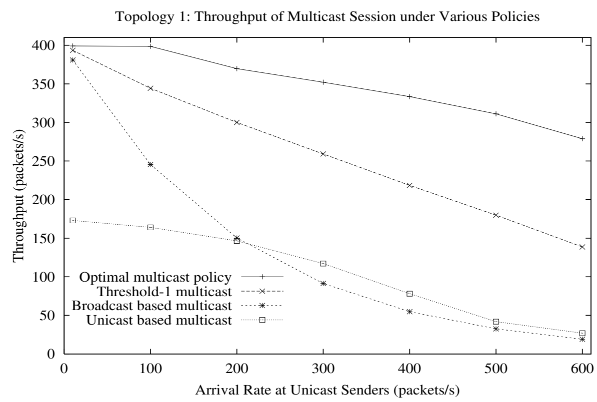
(a) Throughput of Multicast Session Vs. λ_M



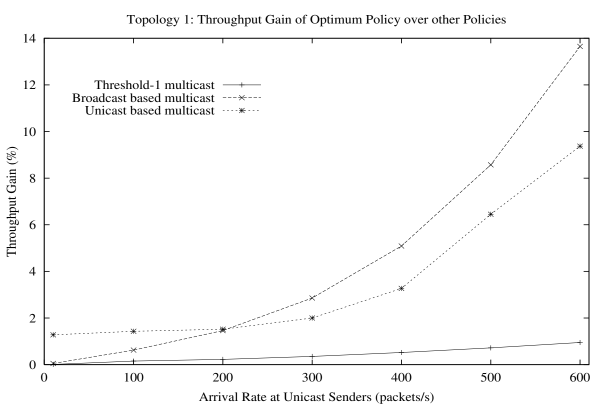
(b) Throughput Gain Vs. λ_M



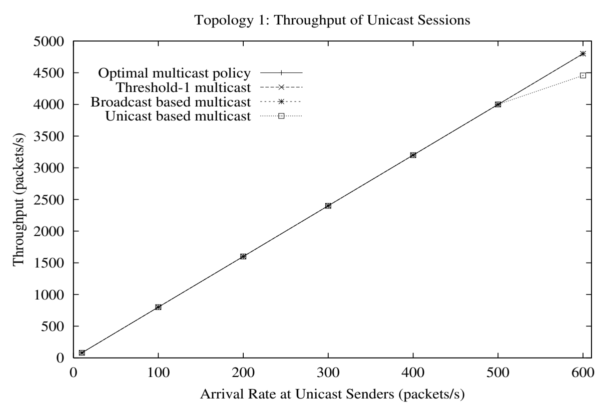
(c) Throughput of Unicast Sessions Vs. λ_M



(d) Throughput of Multicast Session Vs. λ_U

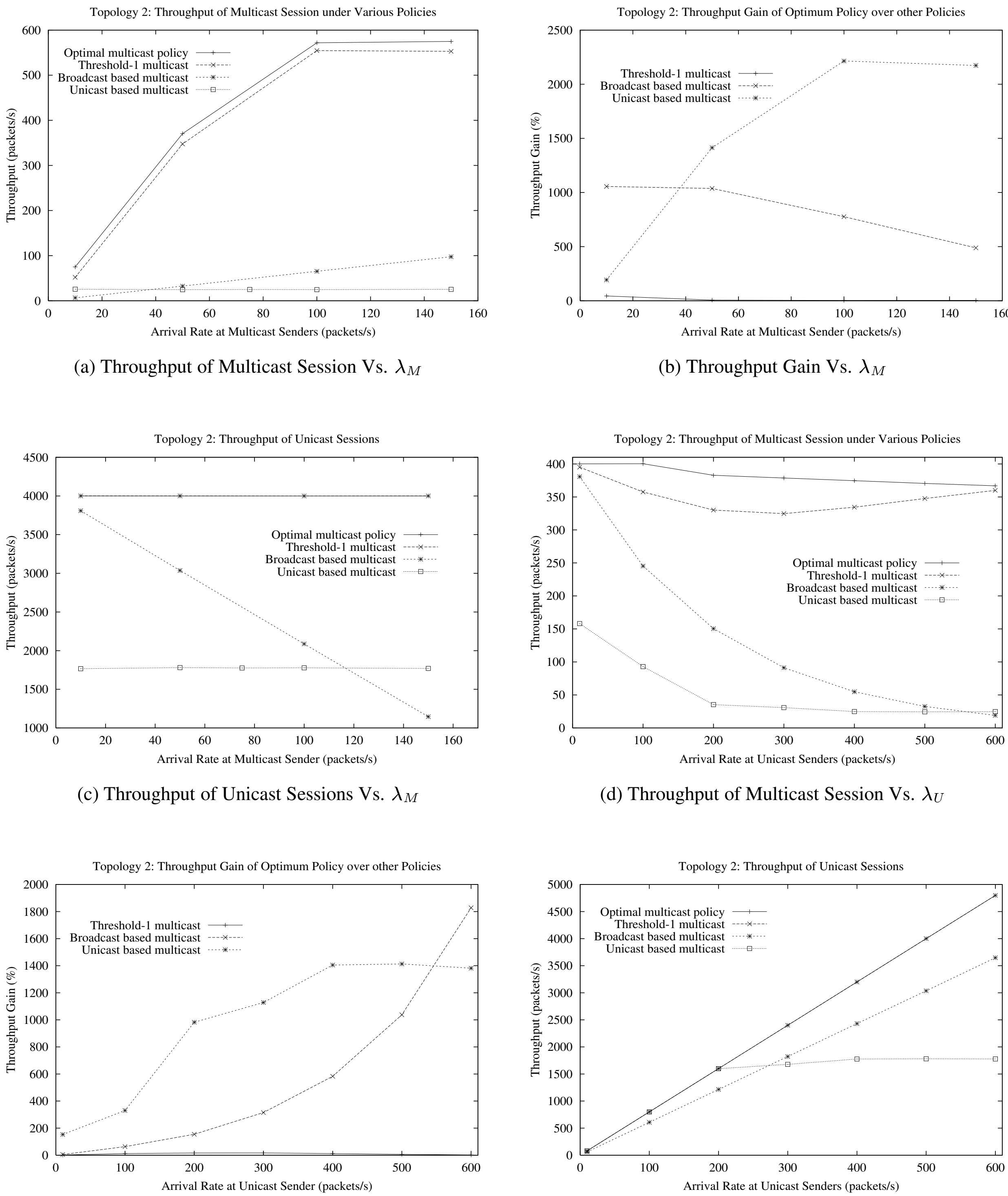


(e) Throughput Gain Vs. λ_U



(f) Throughput of Unicast Sessions Vs. λ_U

Figure 10: The figure plots the performance of various transmission strategies in topology 1. Figures (a) and (c) plot the throughputs of the multicast and unicast sessions, respectively, under various multicast policies as a function of λ_M. Figure (b) plots the throughput gain of $\Delta_O(\Gamma)$ over other polices as a function of λ_M for the multicast session. The throughput gain of $\Delta_O(\Gamma)$ over Δ is computed as $\frac{100\times(\Omega^{\Delta_O(\Gamma)}-\Omega^{\Delta})}{\Omega^{\Delta}}\%$. Here, $\lambda_U = 500$ packets/sec. Figures (d) and (f) plot the throughputs of the multicast and unicast sessions, respectively, as a function of λ_U. Figure (e) plots the throughput gain of $\Delta_O(\Gamma)$ over other policies as a function of λ_U for the multicast session. Here, $\lambda_M = 50$ packets/sec. In the figures, $\Delta_O(\Gamma)$ is referred as *Optimal multicast policy*.



(a) Throughput of Multicast Session Vs. λ_M

(b) Throughput Gain Vs. λ_M

(c) Throughput of Unicast Sessions Vs. λ_M

(d) Throughput of Multicast Session Vs. λ_U

(e) Throughput Gain Vs. λ_U

(f) Throughput of Unicast Sessions Vs. λ_U

Figure 11: The figure shows the performance of various transmission strategies in topology 2. The performance metrics and the parameter values are the same as described in Figure 10.