University of Pennsylvania
ScholarlyCommons

Departmental Papers (ESE)

Department of Electrical & Systems Engineering

August 2006

Dynamic Quorum Policy for Maximizing Throughput in Limited Information Multiparty MAC

Prasanna Chaporkar
University of Pennsylvania

Saswati Sarkar
University of Pennsylvania, swati@seas.upenn.edu

Rahul Shetty
University of Pennsylvania

Follow this and additional works at: http://repository.upenn.edu/ese_papers

Recommended Citation

Prasanna Chaporkar, Saswati Sarkar, and Rahul Shetty, "Dynamic Quorum Policy for Maximizing Throughput in Limited Information Multiparty MAC", . August 2006.

Copyright 2006 IEEE. Reprinted from *IEEE ACM Transactions on Networking*, Volume 14, Issue 4, August 2006, pages 835-848.

This material is posted here with permission of the IEEE. Such permission of the IEEE does not in any way imply IEEE endorsement of any of the University of Pennsylvania's products or services. Internal or personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution must be obtained from the IEEE by writing to pubs-permissions@ieee.org. By choosing to view this document, you agree to all provisions of the copyright laws protecting it.

This paper is posted at ScholarlyCommons. http://repository.upenn.edu/ese_papers/205
For more information, please contact libraryrepository@pobox.upenn.edu.

Dynamic Quorum Policy for Maximizing Throughput in Limited Information Multiparty MAC

Abstract
In multiparty MAC, a sender needs to transmit each packet to a set of receivers within its transmission range. Bandwidth efficiency of wireless multiparty MAC can be improved substantially by exploiting the fact that several receivers can be reached at the MAC layer by a single transmission. Multiparty communication, however, requires new design paradigms since systematic design techniques that have been used effectively in unicast and wireline multicast do not apply. 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 maximizes the system throughput subject to maintaining stability. We present a sufficient condition that can be used to establish the throughput optimality of a stable transmission policy. We subsequently design a distributed adaptive stable policy that allows a sender to decide when to transmit using simple computations. The computations are based only on limited information about current transmissions in the sender's neighborhood. Even though the proposed policy does not use any network statistics, it attains the same throughput as an 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 policy significantly outperforms the existing multiparty schemes in ad hoc networks.

Keywords
MAC layer scheduling, stability, throughput optimal policy, wireless multicast

Comments
Copyright 2006 IEEE. Reprinted from *IEEE ACM Transactions on Networking*, Volume 14, Issue 4, August 2006, pages 835-848.

This material is posted here with permission of the IEEE. Such permission of the IEEE does not in any way imply IEEE endorsement of any of the University of Pennsylvania's products or services. Internal or personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution must be obtained from the IEEE by writing to pubs-permissions@ieee.org. By choosing to view this document, you agree to all provisions of the copyright laws protecting it.

This journal article is available at ScholarlyCommons: http://repository.upenn.edu/ese_papers/205

Dynamic Quorum Policy for Maximizing Throughput in Limited Information Multiparty MAC

Prasanna Chaporkar, Saswati Sarkar, *Member, IEEE*, and Rahul Shetty

***Abstract*—In multiparty MAC, a sender needs to transmit each packet to a set of receivers within its transmission range. Bandwidth efficiency of wireless multiparty MAC can be improved substantially by exploiting the fact that several receivers can be reached at the MAC layer by a single transmission. Multiparty communication, however, requires new design paradigms since systematic design techniques that have been used effectively in unicast and wireline multicast do not apply. 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 maximizes the system throughput subject to maintaining stability. We present a sufficient condition that can be used to establish the throughput optimality of a stable transmission policy. We subsequently design a distributed adaptive stable policy that allows a sender to decide when to transmit using simple computations. The computations are based only on limited information about current transmissions in the sender's neighborhood. Even though the proposed policy does not use any network statistics, it attains the same throughput as an 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 policy significantly outperforms the existing multiparty schemes in ad hoc networks.**

***Index Terms*—MAC layer scheduling, stability, throughput optimal policy, wireless multicast.**

I. Introduction

In multiparty MAC, a sender needs to transmit each packet to a set of receivers within its transmission range. Multiparty MAC forms the basis of a growing and diverse class of network utilities—this motivates the design of intelligent resource allocation policies for multiparty MAC. We first present examples of such utilities.

Enhancing Reliability Using Multi-Path Diversity: Wireless communication is known to be unreliable. Several packets are dropped between the source and the destination, some inadvertently, e.g., due to channel errors during deep fades, extended periods of congestion, some deliberately by misbehaving nodes [1]. Even in one-to-one communication, regulated multiple transmissions of each packet through different paths [2], [3] enhance reliability.[1]

Communicating Routing Updates: Routing updates are communicated through limited flooding [4]–[7].

Anycast: Anycasting is the transmission of a message (e.g., query) such that it reaches at least one node (e.g., server) in a predetermined set [8]. It is used in database query, sensor networks and disaster recovery operations.

Multicast: Multicasting is the transmission of a message such that it reaches multiple nodes [9], [10]. It is used in group communication applications like distance learning and teleconferencing. We distinguish between multiparty MAC and multicast as follows. Unlike in multiparty MAC, in multicast the destinations need not be in the sender's transmission range; multicast is thus an end-to-end communication. Multicast is one of many utilities that can use multiparty MAC.

Since wireless communication is inherently broadcast, in multiparty MAC, a sender needs to transmit each packet only once in order to reach all its receivers. Multiparty communication is likely to benefit significantly from appropriate utilization of this "free-delivery" property. But, the broadcast property leads to several well-known transmission challenges (e.g., the hidden terminal problem), which adversely affect multiparty MAC. We focus on exploiting the advantages and mitigating the disadvantages of the broadcast property so as to design an optimal multiparty MAC.

A multiparty specific challenge is that some but *not all* the receivers may be ready to receive. For example, in Fig. 1, when S_2 is transmitting to R_5, R_2 cannot 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. Thus, S_1 needs to

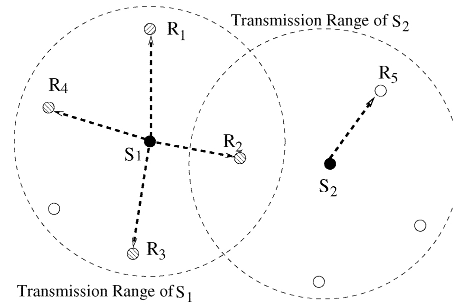

Fig. 1. An example to demonstrate the advantages and the challenges associated with wireless multiparty MAC. The figure shows two senders S_1, S_2 and five receivers R_1 to R_5. R_1 to R_4 are S_1's receivers, and R_5 is S_2's receiver. Dashed circle indicates the communication range of a sender.

[1]Paths may be link-disjoint, node-disjoint, and braided or partially-disjoint.

Manuscript received June 18, 2004; revised March 31, 2005; approved by IEEE/ACM Transactions on Networking Editor E. Knightly. This work was supported by the National Science Foundation under Grants ANI-0106984, NCR-0238340, and CNS-0435306. Parts of this paper were presented at WiOpt'04, University of Cambridge, U.K., March 2004, and MOBIHOC'04, Tokyo, Japan, May 2004.

P. Chaporkar is with INRIA, Paris, France (e-mail: Prasanna.Chaporkar@ens.fr).

S. Sarkar is with the Department of Electrical and Systems Engineering, University of Pennsylvania, Philadelphia, PA 19104 USA (e-mail: swati@.ee.upenn.edu).

R. Shetty is with GTL Limited, India (e-mail: shettre@hotmail.com).

Digital Object Identifier 10.1109/TNET.2006.880175

1063-6692/$20.00 © 2006 IEEE

decide whether it should transmit even when R_2 is not ready, or it should wait until all the receivers are ready.

A transmission policy decides whether a sender should transmit at any given time. A policy that does not allow transmission 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 trade-off between system stability and the throughput. The system clearly needs to be stable. *The challenge therefore is to design a multiparty MAC that maximizes the system throughput, while maintaining system stability.*

In Section III, we describe our system model and obtain a sufficient condition to establish the throughput optimality of an arbitrary stable policy. In Section IV, we propose a quorum based transmission policy that defers transmission at each sender until it has a quorum, i.e., a sufficient number of its receivers are ready to receive. The quorum-policy is suitable for distributed implementation in resource constrained ad hoc networks, as it uses 1) simple computations; 2) no information about system statistics; 3) limited control message exchange; and 4) limited information about its neighbors. We prove that the quorum-policy maximizes throughput among all policies that stabilize the system (see the Appendix). A sender's optimal quorum value depends only on its queue length and its transmission decisions depend on the number of its ready receivers. The first quantity is easily available at a sender. In Section V, we propose a MAC protocol that allow a sender to estimate the second quantity. In Section VI, we evaluate the performance of various multiparty schemes using *ns*-simulations in a wireless network consisting of several multicast and unicast sessions. Simulation results show that the optimal policy provides significantly higher throughput than existing approaches.

II. Literature Review

We now briefly review previous multiparty MAC schemes. Singh *et al.* have proposed a MAC protocol for power aware broadcast [11]. Wang *et al.* have proposed a scheduling and power control protocol to minimize the transmission powers [12]. Jaikaeo *et al.* have studied multiparty communication using directional antennas [13]. Kuri *et al.* have proposed a protocol for reliable packet delivery in wireless LANs [14]. This protocol is based on assumptions that hold in wireless LANs but not in ad hoc networks. For ad hoc networks, Tang *et al.* have proposed Broadcast Medium Window (BMW) protocol, which is a *unicast based multiparty* MAC that transmits a packet to each receiver separately in round robin fashion [15]. IEEE 802.11 implements multiparty communication by broadcasting a packet after disabling all control messages – we refer to this as *broadcast based multiparty*. Thus, second hop interference is ignored. Tang *et al.* have proposed a Broadcast Support Multiple Access (BSMA) protocol, which is a *quorum*-1 *multiparty* scheme [16], [17]. In this scheme, a sender transmits a packet whenever at least one receiver is ready to receive. Sun *et al.* have proposed Batch Mode Multicast MAC (BMMM) protocol, which also implements quorum-1 multiparty scheme [18]. The unicast based multiparty policy does not exploit the broadcast nature of wireless medium, and its multiple transmissions of a packet waste power and bandwidth. The broadcast based multiparty and quorum-1 multiparty policies cause packet loss at receivers because several receivers may not be ready at the time of transmission. The broadcast based multiparty also causes packet collision due to second hop interference. Thus, the performance trade-offs have not been adequately explored for multiparty MAC. Furthermore, there is no analytical performance guarantee for any of these schemes. Several interesting protocols have been proposed at the transport and network layers for the utilities that would use multiparty MAC, e.g., multicast [10], [11], [19], [20], transmitting routing updates [4]–[7], etc. These higher layer protocols can work with any underlying MAC, and their performances depend on the efficiency of the MAC which is the focus of our research.

III. System Model

We consider a wireless network with several MAC layer multiparty and one-to-one (unicast) sessions. Each multiparty MAC session comprises of a sender and a set of receivers (*party*) that are in the sender's transmission range. However, all nodes need not be in each other's transmission range. We consider transmission of data traffic. Time is slotted. We assume that each packet can be transmitted in a single slot.

A. Wireless Multiparty MAC Requires New Design Paradigms

A major design challenge in wireless multiparty MAC 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 $\lambda_1, \ldots, \lambda_n$ respectively. Consider only the transmission policies that ensure that each packet is received correctly by at least one designated 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 policy is the set of arrival rates $\vec{\lambda} = (\lambda_1, \ldots, \lambda_n)$ 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. Lyapunov function is a positive real valued function of queue lengths [21]. The Lyapunov function is said to have a negative drift, if its expected value decreases for large queue lengths. Then, the queue lengths are not likely to become large, and hence the system is stable. Thus, existence of a Lyapunov function with negative drift for every $\vec{\lambda} \in \Lambda$ under a certain policy is sufficient to prove that the policy maximizes the stability region. This in turn would prove that such a policy maximizes the throughput in packet radio and wireline multicast networks [22], [23]. This systematic approach cannot be used in wireless multiparty MAC as a policy that attains Λ need not maximize the throughput and vice versa.

Example: Consider Fig. 1. When S_1 transmits, R_1, R_3, R_4 receive the packet without any error; R_2 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 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, when $\lambda_2 > 0$, the throughput under Δ_2 is strictly higher than that under Δ_1 for $\vec{\lambda}$ in Λ_2. 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 new design techniques to attain the objective.

B. How Much Should Nodes Coordinate in Multiparty MAC?

The optimum policy and the maximum throughput depends on how much each node knows about the network. We describe three broad categories of coordination levels.

1) Full Coordination: Nodes coordinate with each other so that each node knows the queue lengths at all other nodes in the network. Thus, each node decides when to transmit based on the knowledge of every other node's transmission decisions. This is equivalent to having a centralized scheduler that knows the state of the entire network, decides the transmissions and informs the nodes accordingly. For example, in unicast packet radio networks, Tassiulas *et al.* have presented an optimum scheduling policy under full coordination [22].

2) Active Information Exchange: Each node decides when to transmit based only on the transmissions in its neighborhood, and the readiness states of its receivers. It learns the former by sensing the channel, and the latter by limited message exchange with its receivers. A node does not know anything else about the network, and does not coordinate its transmissions with those of any other node. The unicast IEEE 802.11 belongs in this framework. In IEEE 802.11, each node decides when to transmit based on channel sensing and RTS-CTS exchange with its receiver.

3) Passive Observation: Each node decides when to transmit based only on the transmissions in its neighborhood, which it learns by sensing the channel. Nodes do not exchange any control message. Randomized MAC protocols like ALOHA and CSMA belong in this framework.

An optimum full coordination scheme will have the maximum throughput, but is not likely to be deployed given its need for huge control message exchange and/or centralized coordination. On the other extreme, passive observation based schemes will be simple to implement, but will have low throughput due to excessive collisions. The active information exchange case provides a nice trade-off between the two extremes both in terms of throughput and control overhead, and are therefore most likely to be deployed. For example, IEEE 802.11 is one of the most popular unicast MAC schemes. We design an optimal dynamic multiparty MAC based on active information exchange. The proposed policy is distributed, adaptive, computationally simple, and can be implemented using a simple modification of IEEE 802.11.

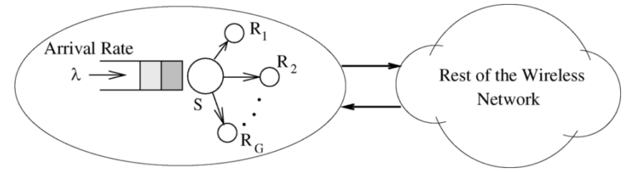

Fig. 2. MAC layer multiparty session and its interaction with the rest of the network. Node S is a sender, where packets arrive at the rate λ. Sender S transmits the packets to receivers R_1 to R_G.

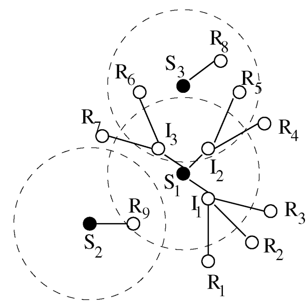

Fig. 3. Multicast session from S_1 to receivers R_1 to R_7 and two unicast sessions from S_2 to R_9 and S_3 to R_8 in a multi-hop wireless network. First we observe that a single network layer multicast session corresponds to many MAC layer multiparty sessions, e.g., using multiparty communication S_1 transmits to intermediate nodes I_1 to I_3, I_1 transmits to the receivers R_1 to R_3, etc. Consider a MAC layer multiparty 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 S_1's 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.

C. Mathematical Framework and System Objectives for Dynamic Multiparty MAC Based on Active Information Exchange

Fig. 2 represents the interaction between a MAC layer multiparty session, and 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 multiparty session and vice versa. The effect of the rest of the network on the multiparty 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 Fig. 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. This interference is controlled as follows. The sender does not transmit if any node in its neighborhood is receiving a packet. For example, in Fig. 3, S_1 does not transmit when S_2 is transmitting 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 that of the sender.

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

In each slot, the sender decides whether to transmit with the goal of maximizing the throughput subject to attaining stability. We determine the sender's optimal strategy based on its (a) readiness state (which it determines by sensing the channel); (b) queue length; and (c) observation of its receivers' readiness states. We adopted this model because the senders do not coordinate their transmissions, and thus from the perspective of a sender the network is a stochastic disturbance which is partially observable but not controllable. Each sender finds the network only partially observable as it knows only the readiness states of its receivers. Different fairness goals can be attained and inter-session interaction can be controlled by selecting appropriate backoff intervals (e.g., as in [24]) which in turn regulates each sender's and receiver's readiness states. *We allow an arbitrary ergodic readiness process so as to incorporate any desired inter-session interaction.* We focus on maximizing the throughput subject to stability for any given readiness process. This requires us to address several open research problems that are specific to multiparty MAC.

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. Each packet can be transmitted in a single slot. 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.

This class includes *offline* policies that decide transmissions based on knowledge of packet arrivals and readiness vectors in all past, present and future slot.

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.

Definition 5: A system is *stable* if the sender's 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: *Quorum* is the minimum number of multiparty receivers that have to be ready for a sender to transmit.

Definition 9: A policy Δ belongs to the class of *generalized quorum policies*, if it sets quorum $T(t) \in \{0, \ldots, G+1\}$ 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 quorum may be selected based on past, present and future arrivals and readiness states.

For any transmission policy $\widehat{\Delta}$, there exists a generalized quorum 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 quorum policy Δ that computes slots $t_1, t_2, \ldots$ using the same rule as $\widehat{\Delta}$ and sets quorum 0 in these slots. In the remaining busy slots, Δ sets quorum $G+1$. Thus, $\widehat{\Delta}$ and Δ transmit in the same slots. Hence, it is sufficient to consider only generalized quorum policies.

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

Theorem 1: For any $\epsilon > 0$, a stable generalized quorum policy Δ is ϵ-throughput optimal with probability (w.p.) 1 if the following condition holds for some $T \in \{0, \ldots, G\}$:

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

We now motivate the above result. The number of packets served per unit time under any stable policy is equal to the arrival rate λ. A stable policy Δ_1 can achieve throughput higher than that of Δ only by attaining a higher reward for infinitely many packets. Now, Δ selects quorum values T and $T+1$ except for $\epsilon/2G$ fraction of slots. We refer to these slots as *type-1* slots, and we refer to the remaining slots as *type-2* slots. Thus, Δ transmits in every type-1 slot that has $T+1$ or more ready receivers. Each of the remaining packets transmitted in type-1 slots achieves reward T. Thus, Δ_1 can achieve a higher reward infinitely often only by transmitting packets in type-2 slots. Now, even if all type-2 slots have G ready receivers, then the improvement in the throughput is at most $\epsilon/2$ as the fraction of type-2 slots is at most $\epsilon/2G$. Thus, Δ is $\epsilon/2$-throughput optimal and hence ϵ-throughput optimal.

Theorem 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 quorum policy. The utility of the theorem 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. We next design an adaptive transmission policy that satisfies condition (1) and hence is ϵ-throughput optimal.

IV. Throughput Optimal Transmission Policy ($\Delta_O(\Gamma)$)

We describe a parametrized quorum-policy $\Delta_O(\Gamma)$ that we prove to be ϵ-throughput optimal. The policy selects a quorum 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 quorum; and (c) the sender has a packet to transmit. In other words, the sender does not transmit unless it has a "quorum" i.e., unless the number of ready receivers exceeds or equals the selected quorum value. The quorum values are selected as follows. Let Q denote the queue length at the sender and let parameter Γ be some fixed positive integer. For $1 \leq T \leq G$, the quorum is T if $(G-T)\Gamma < Q \leq (G-T+1)\Gamma$ and quorum is 0 if $Q > G\Gamma$. Thus, the quorum value increases with decrease in queue length. The policy does not select a quorum when queue length is zero.

A. Analytical Performance Guarantees for $\Delta_O(\Gamma)$

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 receivers' readiness states in the same and different time slots. Fig. 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 until time t in which the sender and u receivers are ready. Now, by ergodicity of the readiness process

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

In general, from ergodicity we cannot conclude anything about the rate of convergence of the empirical distribution $\lim_{t\to\infty} n_u(t)/t$ to the steady state distribution b_u^R. But for finite, aperiodic MC's, empirical distribution converges to the steady state distribution exponentially fast [25]. We use this exponential convergence to prove the optimality of $\Delta_O(\Gamma)$. *The optimality of* $\Delta_O(\Gamma)$ *holds for any ergodic process that has the exponential convergence property.*

Next, we formally state the optimality result. Let the throughput of policy $\Delta_O(\Gamma)$ be $\Omega^{\Delta_O(\Gamma)}$, and let the maximum throughput attained by a stable policy be Ω_{opt}.

Theorem 2: 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 $\lambda > \sum_{u=0}^{G} b_u^R$.

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 cannot attain throughput more than $\Omega^{\Delta_O(\Gamma)}$. *This holds even though* $\Delta_O(\Gamma)$ *takes transmission decisions based only on 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 quorum in every slot. The expected reward is a monotonically increasing function of the quorum. Hence, a throughput optimal policy should select the largest quorum T_O that stabilizes the system, i.e.,

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

The throughput can be further improved by appropriately randomizing between the quorum values T_O and T_O+1. The randomization should be such that the system remains stable. Intuitively, the optimum policy should select the quorums T_O and T_O+1 most of the time. *The difficulty is that the sender does not know* λ *and* b_u^R*'s, and thus cannot compute* T_O. *But,* $\Delta_O(\Gamma)$ *selects the quorums* T_O *and* T_O+1 *most of the time, even though it does not know* T_O. This can be explained as follows. From (3), the rate at which slots with m or more ready receivers arrive is more than the packet arrival rate λ, for every $m \leq T_O$. But, for $m \geq T_O+1$ the rate at which the slots with m or more ready receivers arrive is smaller than λ. Thus, for quorum 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. Hence, the queue length increases, and consequently the quorum decreases. However, for quorum 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 the queue length decreases, and the quorum increases. Hence, when Γ is large enough the quorums 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. Now, $\Delta_O(\Gamma)$ 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 multiparty communication in ad hoc networks, like broadcast based multiparty and quorum-1 multiparty are unreliable as well. Fortunately, several reliable transport layer schemes have been proposed for wireless multicast transmissions, which can be used in conjunction with any multiparty MAC strategy [19], [20]. 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 3: 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 $\epsilon > 0$.

We describe the intuition behind this result. In a stable system, the throughput of a transmission policy is λR, where R is the policy's average reward per packet. Thus, $\Delta_O(\Gamma)$ maximizes R since it maximizes the throughput. Now, since the system loss under any policy is $G - R$, $\Delta_O(\Gamma)$ minimizes the system loss as well. Refer to [26] for the formal proof.

From Theorem 3, if the system loss for $\Delta_O(\Gamma)$ is more than that the system can tolerate, then the required loss constraint cannot be guaranteed by any stable 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 and rate control must be resorted to. This is beyond the scope of this paper. Henceforth, we do not consider loss explicitly.

B. Properties of $\Delta_O(\Gamma)$

1) In each slot, $\Delta_O(\Gamma)$ takes transmission decisions at each sender based only on local information: (a) sender's current queue length, (b) sender's and receivers' current readiness states. Hence, $\Delta_O(\Gamma)$ is distributed and dynamic.
2) Under $\Delta_O(\Gamma)$, each sender need not know which particular receivers are ready. The number of ready receivers turns out to be a sufficient statistic for throughput optimality. This simplifies the protocol design problem.
3) $\Delta_O(\Gamma)$ is computationally simple.
4) $\Delta_O(\Gamma)$'s optimality is guaranteed for all $\Gamma \geq \Gamma_O$. Now, Γ_O depends on the system parameters. But, only a rough estimate of Γ_O (e.g., an upper bound on Γ_O), is necessary for appropriately selecting Γ. Furthermore, simulations show that $\Delta_O(\Gamma)$'s performance is similar for different values of Γ. Once Γ is selected, $\Delta_O(\Gamma)$ does not require any statistical or topological information.

The first three properties of $\Delta_O(\Gamma)$ follow from its description, and the last property follows from Theorem 2.

V. Dynamic Multiparty MAC Protocols

The optimal decision rule at each sender is based on its queue length, 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 present a protocol to inform each sender about the number of ready receivers and transmissions in its neighborhood. In unicast, IEEE 802.11 uses RTS-CTS-DATA-ACK handshake for this purpose. The difficulty in multiparty communication is that if all the receivers of a sender send CTS simultaneously in response to the sender's RTS, then these CTS messages will collide. Hence, the sender will not know whether the receivers are ready, and cannot decide whether to transmit. Also, other nodes will not know whether to defer their transmissions.

We now propose a sequential CTS (SCTS) transmission scheme so as to prevent the collision of the CTS's. Each sender allots a unique sequence number $i \in \{0, \ldots, G-1\}$ to each of its receivers. When a sender wishes to transmit a packet, it first sends an RTS addressed to its party. A ready receiver with sequence number i sends a CTS after $ic + (i+1)s$ time units after it receives the RTS. The quantity c is the time required to transmit a CTS and s is one Short Inter Frame Space (SIFS) duration. The sender transmits the packet if the number of CTS responses it received is greater than or equal to the quorum determined by the decision rule (e.g., $\Delta_O(\Gamma)$ or $\hat{\Delta}_O(\Gamma)$).

Each RTS has a duration/ID field with value $P + Gc + (G+1)s$, where P is the duration of the packet. The ith receiver's CTS message has a duration/ID field with value $P + (G - i - 1)c + (G - i)s$. The nodes in the neighborhood of the sender and receivers set their Network Allocation Vector (NAV) equal to the maximum of the current NAV value and the value of duration/ID field in RTS (CTS) message, and does not transmit a packet in the NAV duration.

In the multiparty case, depending on the number of CTS messages received, the sender may not transmit a packet even after an RTS-CTS exchange. In this case, it transmits a release message. If a receiver receives a release message or does not receive data within a certain interval of transmitting CTS message, then it transmits a release message. A node that receives a release message either from the sender or a receiver resets its NAV to the previous value. After deciding not to transmit a packet, or after completing a packet transmission, each sender backs off for an i.i.d. (uniform) random interval.

The SCTS scheme has several advantages. It requires only a minor modification of IEEE 802.11. Thus, it can co-exist with IEEE 802.11, i.e., unicast and multiparty senders can implement IEEE 802.11 and SCTS respectively. Moreover, this scheme does not use direct sequence (DS) capture like BSMA [16], [17]. It has been shown that DS capture cannot be used in general topologies [18]. The control message exchange in SCTS is similar to that in BMMM [18]. The difference between SCTS and BMMM is that SCTS broadcasts the RTS whereas BMMM separately transmits RTS to each receiver. Also, unlike BMMM, SCTS requires release messages.

VI. Simulation Results and Discussion

We have proved that $\Delta_O(\Gamma)$ is ϵ-throughput optimal, when the readiness states are Markovian, time is slotted and information about readiness states is instantaneously available at the sender. We compare the application layer throughputs in a wireless network with several unicast and multiparty sessions for the following different multiparty MACs when the assumptions made in analysis do not hold: each unicast sender uses IEEE 802.11b and each multiparty sender uses 1) $\Delta_O(\Gamma)$; 2) broadcast based multiparty; 3) unicast based multiparty; 4) quorum-1 multiparty; and 5) full-quorum multiparty. We also investigate the impact of control overhead on the performance of these policies. Now, the readiness states are generated due to packet transmissions, time is continuous and the sender learns readiness states by exchanging control messages. The simulation results demonstrate that $\Delta_O(\Gamma)$ attains significantly higher throughput than the other existing policies. We implement these policies in *ns-2* [26].

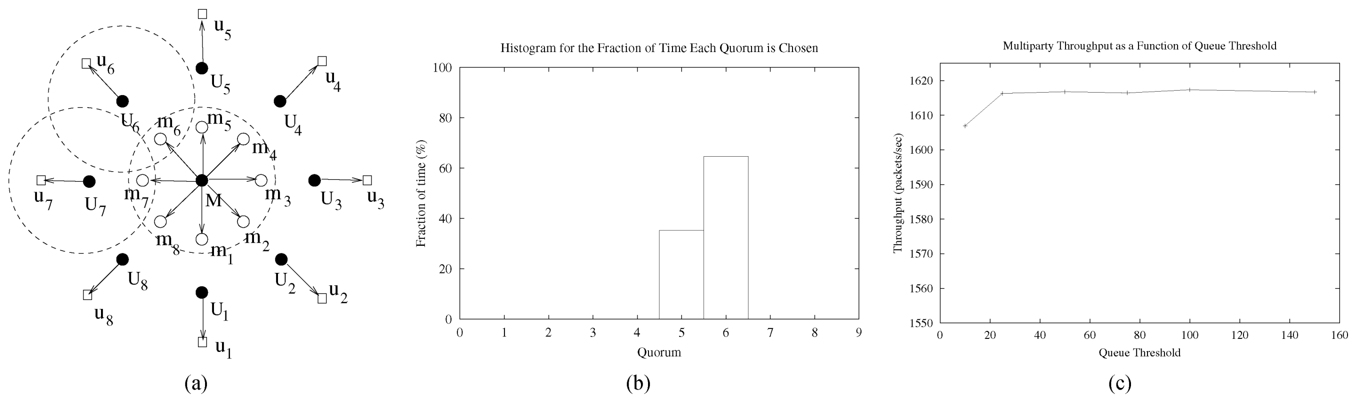

Fig. 4. We evaluate the the performance of various multicast strategies in the topology shown in (a). The topology has a multicast session with sender M and eight receivers m_1 to m_8 and eight unicast sessions. Unicast session i has sender U_i and receiver u_i, $1 \leq i \leq 8$. Here $\lambda_M = 200$ packets/s and $\lambda_U = 500$ packets/s. (b) plots the fraction of time each quorum is chosen by $\Delta_O(\Gamma)$ for $\Gamma = 75$. (c) shows the throughput of the multiparty session under $\Delta_O(\Gamma)$ as a function of Γ. In each part, Γ is referred to as *queue threshold*.

A. 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 is a topic of research even for unicast sessions. We measure a receiver's throughput as the number of packets it receives successfully per unit time, and a session's throughput as the sum of its receivers' throughputs. We use the SCTS protocol to implement $\Delta_O(\Gamma)$, quorum-1 and full-quorum multiparty policies. We consider a time interval of 2000 seconds and collect the relevant data only in the last 1500 seconds. Each channel has capacity $C = 11$ Mbps. The RTS packet has 44 bytes. The CTS and release packets have 38 bytes. Multiparty senders in unicast based multiparty MAC and unicast senders send ACK packets of size 38 bytes. The maximum propagation delay is 2 μs, and SIFS duration is 10 μs.

We present the simulation results for a topology shown in Fig. 4(a). In Fig. 4(a), each unicast sender U_i generates packets at rate λ_U and the multiparty sender M generates packets at rate λ_M. The packet arrival processes are Poisson. The packets arriving at U_i have length 100 bytes. Size of the packet header is 52 bytes. The trends remain same for larger packet sizes for unicast sessions; the results differ only in magnitude.

Now we discuss how the packet transmissions generate readiness states in the topology. For every $i \in \{1, \ldots, 8\}$, U_i is ready when it is not backing off and m_i has not reserved the channel by transmitting a CTS. Also, m_i is ready when U_i is not transmitting a packet to u_i. The multiparty sender M is not ready when it backs off, while u_i is always ready.

B. Discussion on Simulation Results

We have proved that any stable policy that selects any two consecutive quorums T and $T+1$ most of the time, maximizes the throughput as long as $n_u(t)/t$ converges. Our simulations demonstrate that even when the readiness states are generated by packet transmissions, $\Delta_O(\Gamma)$ selects two consecutive quorums most of the time [Fig. 4(b)], and $n_u(t)/t$ converges [26]. This validates the optimality result. In addition, Fig. 4(c) shows that as Γ increases, the throughput of the multiparty session under $\Delta_O(\Gamma)$ converges to the optimum value. We note that optimality is achieved even for small values of Γ.

We observe that $\Delta_O(\Gamma)$ achieves substantial throughput gain over other existing policies [Fig. 5(a) and (c)]. We next explain the trend. The broadcast based multiparty scheme does not exchange any control messages, and thus causes frequent data packet collisions. Thus, the reward per packet is low resulting in low throughput. Quorum-1 policy exchanges control messages and avoids data packet collisions. As a result, this policy provides much better throughput than broadcast based multiparty policy. The limitation of this scheme is that the quorum is always 1, and hence the policy may transmit even when only a few receivers are ready. The full-quorum policy achieves optimum throughput for small load, but saturates quickly ($\lambda_M = 50$ packets/s). Thus, though the reward per packet is high, the number of packets transmitted is much smaller resulting in low throughput and unstable system. The policy $\Delta_O(\Gamma)$ outperforms these policies as it prevents data packet collisions by exchanging control messages. Also, by selecting an appropriate quorum value, $\Delta_O(\Gamma)$ prevents transmission when only a few receivers are ready, transmits fast enough so as to attain stability, and therefore obtains the best possible reward per packet constrained to stability. The unicast based multiparty policy uses separate transmissions to reach different receivers even when they can be reached using a single transmission. Hence, the total number of packets delivered under this policy is much smaller than that under other policies. This results in low throughput.

Fig. 5(b) shows that when the multicast sender uses $\Delta_O(\Gamma)$, the throughput of unicast sessions is similar to that under any other policy for the multicast sender. 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.

We now evaluate the control overhead of different policies. The overhead decreases the throughput as transmission of the control packets increases packet transmission times, and increases the energy consumption due to transmission of additional control packets. The detrimental effect of overhead on the throughput and energy consumption increases with

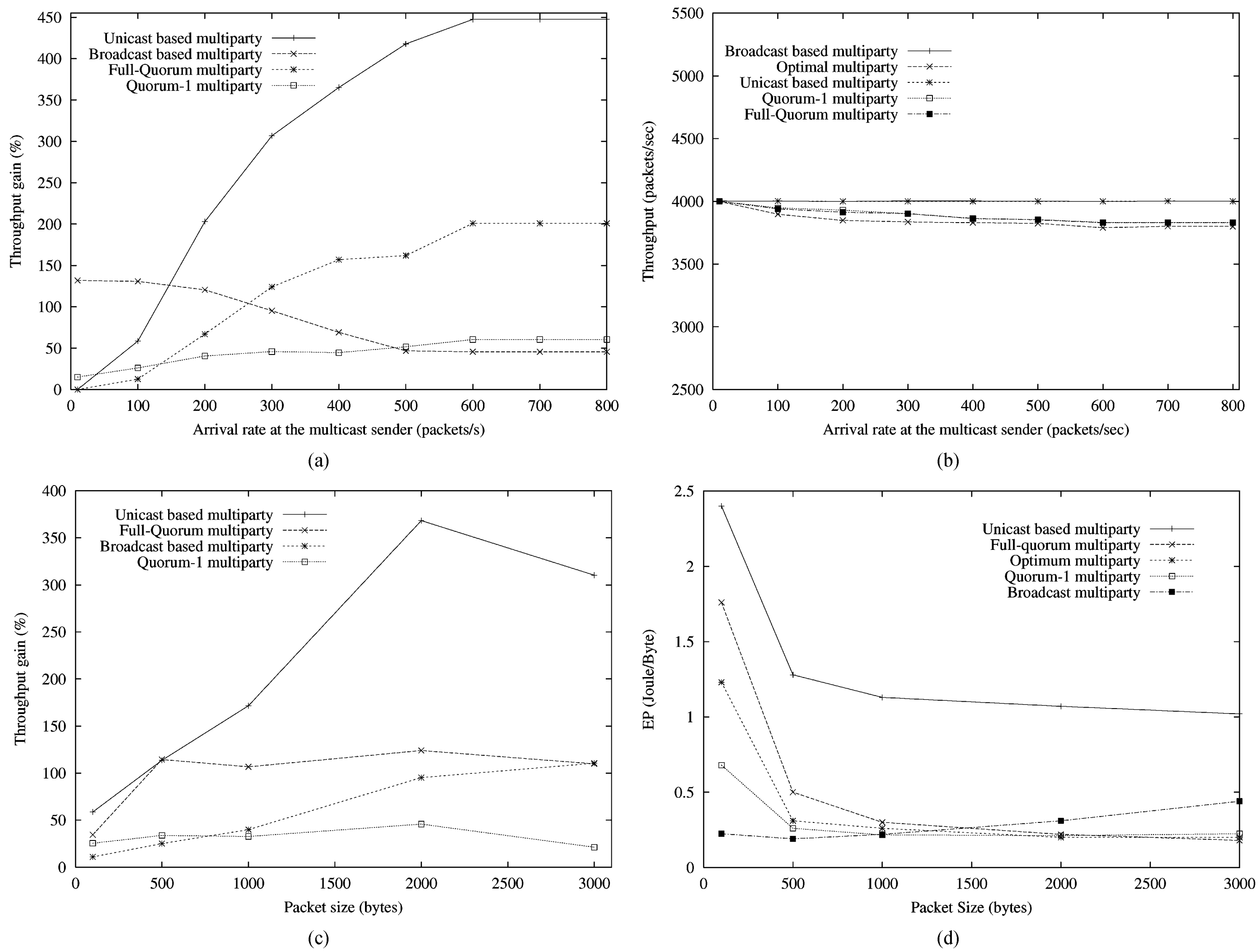

Fig. 5. We evaluate the performance of various multicast strategies in the topology shown in Fig. 4(a). In each part, $\lambda_U = 500$ packets/s, and $\Delta_O(\Gamma)$ is referred to as *Optimal multiparty policy*. (a) plots the throughput gain of $\Delta_O(\Gamma)$ over other polices as a function of λ_M for the multiparty session. The throughput gain of $\Delta_O(\Gamma)$ over Δ is computed as $(100 \times (\Omega^{\Delta_O(\Gamma)} - \Omega^{\Delta}))/\Omega^{\Delta}\%$. In (b) we plot the throughput of unicast sessions under various policies. In (a) and (b), P_M is 2000 bytes. In (c) and (d), we plot the throughput gain of $\Delta_O(\Gamma)$ over other polices and energy consumed per payload byte (EP) under various policies, respectively, as a function of P_M. Here, $\lambda_M = 300$ packets/s. EP is computed as $(P_M + O)/(g + P_M)$ Joules/byte.

increase in the ratio between the overhead and payload, which in turn depends on the packet sizes. We therefore investigate the impact of overhead on the performance of different policies by evaluating their throughput [Fig. 5(c)] and energy consumption [Fig. 5(d)] for different packet sizes.

Fig. 5(c) shows that even for small packet sizes, e.g., 500 bytes, $\Delta_O(\Gamma)$ achieves significant throughput gain over other policies. Thus, the high reward $\Delta_O(\Gamma)$ achieves per packet more than compensates for a larger net packet transmission time, which happens due to additional overhead. Furthermore, if overhead consumes large time, then the service rate under $\Delta_O(\Gamma)$ decreases and hence queue length at M increases. This also lowers the quorum. Thus, $\Delta_O(\Gamma)$ queries the system fewer times to achieve the quorum, which reduces the overhead. Thus, $\Delta_O(\Gamma)$ implicitly considers the control overhead in its decision process, and thereby achieves higher throughput. As expected, the increase in throughput diminishes as packet size decreases.

Now, we evaluate the energy consumption of various policies. Let M spend α Joules for transmitting each byte. Then, the total energy consumed in transmitting a packet is equal to $(P_M+O)\alpha$ Joules, where P_M is the packet size and O is the total overhead that includes packet headers, RTS, CTS and Release messages. A policy Δ delivers $g \times P_M$ payload bytes in a transmission, where g is the reward. The *energy consumed per payload byte* (EP) is the ratio of total energy spent and the total payload bytes delivered per packet ($\alpha(P_M + O)/(g \times P_M)$ Joules/byte). Without loss of generality, we assume $\alpha = 1$. Fig. 5(d) shows that for moderate packet sizes ($\geq$ 500 bytes), the EP of $\Delta_O(\Gamma)$ is comparable to that of other policies. For small packet size (100 bytes) $\Delta_O(\Gamma)$'s EP is significantly higher than that of quorum-1 multiparty and broadcast based multiparty. Now, we explain this trend. $\Delta_O(\Gamma)$ achieves significantly higher g, but has higher O. Note that EP equals $(1/g)(1 + O/P_M)$ Joules/byte. When packet size is moderate, $1/g$ dominates, and hence EP of $\Delta_O(\Gamma)$ is comparable to that of other policies. For smaller packet size, O/P_M dominates, and hence the EP of $\Delta_O(\Gamma)$ is larger than that of some other policies. The energy overhead under unicast based multicast is much higher than that under other policies as it transmits a packet separately to each receiver.

Summarizing, the simulations demonstrate that the gain in throughput attained by $\Delta_O(\Gamma)$ over other approaches more than compensates for its use of additional control overhead. Now, $\Delta_O(\Gamma)$ can be implemented in other ways so as to further mitigate the impact of overhead, yet retain its advantages. For example, consider the *burst sequential CTS protocol* (BSCTS) which can be used for delay tolerant traffic. BSCTS differs from SCTS in that in BSCTS the sender contends for channel access only when it has D packets ($D > 1$), where D is a constant. The sender transmits D packets in a single frame if it has a quorum. Thus, the frame size in BSCTS is D times that in SCTS, where D can be chosen so that the size of an RTS/CTS packet is significantly less than this frame size. This decreases the ratio between the control overhead and payload to any desired value, and reduces the EP of BSCTS to values lower than that of all other policies. For example, even when $P_M = 100$ bytes, if $D = 20$, the EP of BSCTS is 0.14 which is smaller than that of all the remaining policies. Note that BSCTS retains the throughput optimality of $\Delta_O(\Gamma)$[26]. Furthermore, if $\Delta_O(\Gamma)$ is stable, then BSCTS does not alter the fraction of time $\Delta_O(\Gamma)$ occupies the channel, and hence does not affect the throughput of other sessions. Detailed investigations of BSCTS and other protocols that implement $\Delta_O(\Gamma)$ constitute interesting topics for future research.

VII. Conclusion

Maximizing the performance in wireless multiparty MAC 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 goal therefore is to maximize throughput subject to attaining stability. We consider a scenario where each sender decides its transmissions based on the transmissions and readiness states in its neighborhood, and does not coordinate its decisions with its neighbors. We present a sufficient condition that can be used to establish the throughput optimality of a stable transmission policy. We subsequently design a distributed, adaptive stable quorum-policy that allows a sender to decide when to transmit using simple computations based only on its local information. The quorum-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 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 ad hoc 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 open problems are: 1) maximizing performance in full-coordination and passive observation cases; 2) studying the multiparty MAC's interaction with higher layer protocols for utilities like multicast, anycast, transmitting routing updates, attaining reliability through multipath diversity etc.; 3) maximizing the performance in presence of mobility, dynamic group membership changes, security concerns, etc.; and 4) designing a protocol that implements $\Delta_O(\Gamma)$ with the minimum possible control overhead.

Appendix

First we present some definitions.

Definition 10: A single quorum transmission policy (T) (denoted by Δ_T) is a generalized quorum policy for which $T(t) = T$ for every busy slot.

Definition 11: A single quorum policy (T) in a system with finite buffer capacity B will be denoted by $\Delta_{T,B}$.

We note that the policies $\Delta_O(\Gamma)$, Δ_T and $\Delta_{T,B}$ belong to the class of generalized quorum policies. Without loss of generality, we assume that a generalized quorum policy chooses quorum $G+1$ when the sender's queue is empty. In this case, the choice of quorum does not affect the transmission decision as the sender cannot transmit a packet anyway.

Now, we consider a process observed by the sender under an arbitrary transmission policy Δ as a three-dimensional process $Y_n^{\Delta} = (k, \vec{j}, a)$, where k is the queue length at the sender, $\vec{j}$ is the readiness state and a is the arrival process state in the nth time slot. Since the readiness and the arrival processes are Markovian, the process $\{Y_n^{\Delta} : n \geq 1\}$ is a Discrete-Time Markov Chain (DTMC), if $\Delta \in \{\Delta_O(\Gamma), \Delta_T, \Delta_{T,B}\}$. Furthermore, the system is stable under $\Delta_O(\Gamma)$, Δ_T and $\Delta_{T,B}$ if and only if the DTMC is positive recurrent. Thus, the stability implies existence of a unique stationary distribution. Let us denote by $\tilde{\pi}_k$, π_k and $\widehat{\pi}_k$ the steady state probability that the queue length at the sender is k under policies $\Delta_O(\Gamma)$, Δ_T and $\Delta_{T,B}$, respectively.

A. Proof of Theorem

Proof: We consider a generalized quorum policy Δ that satisfies (1). Our aim is to show that for an arbitrary policy Δ_1

$$\Omega^{\Delta_1} - \Omega^{\Delta} \leq \epsilon \qquad \text{w.p. 1.} \tag{4}$$

For every stable policy Δ_1, $\lim_{t\to\infty} z^{\Delta_1}(t)/t = \lambda$ w.p. 1, where $z^{\Delta_1}(t)$ denote the number of packets transmitted under policy Δ_1 until time t.

$$\text{Thus,} \quad \lim_{t\to\infty} \frac{z^{\Delta}(t)}{t} = \lim_{t\to\infty} \frac{z^{\Delta_1}(t)}{t} = \lambda \qquad \text{w.p. 1.} \tag{5}$$

Let $S^{\Delta}(t)$ denote the number of slots until time t in which the quorum under Δ is T or $T+1$ and let $\bar{S}^{\Delta}(t)$ denote the remaining slots until time t. Hence, for every $t \geq 1$

$$S^{\Delta}(t) = s_T^{\Delta}(t) + s_{T+1}^{\Delta}(t) \tag{6}$$

$$S^{\Delta}(t) + \bar{S}^{\Delta}(t) = t. \tag{7}$$

$$\text{From (1) and (7),} \quad \lim_{t\to\infty} \frac{\bar{S}^{\Delta}(t)}{t} \leq \frac{\epsilon}{2G}. \tag{8}$$

Further, let $n_u(t)$ be the number of slots until time t in which the sender and u receivers were ready. Ergodicity of the readiness process implies for both Δ and Δ_1

$$\lim_{t\to\infty} \frac{n_u(t)}{t} = b_u^R \qquad \text{w.p. 1.} \tag{9}$$

Let $S_u^{\Delta}(t)$ denote the number of slots until time t in which the sender and u receivers were ready, and the quorum under Δ was T or $T+1$.

$$\text{Let} \quad \bar{S}_u^{\Delta}(t) = n_u(t) - S_u^{\Delta}(t) \quad \text{for every } t \geq 1. \tag{10}$$

$$\text{Now} \quad \sum_{u=0}^{G} \bar{S}_u^{\Delta}(t) \leq \bar{S}^{\Delta}(t) \quad \text{for every } t \geq 1. \tag{11}$$

Finally, let $\hat{z}^{\Delta}(t)$ denote the number of packets that departed until time t when the quorum under Δ is T or $T+1$.

$$\text{For every } t \geq 1, \quad \hat{z}^{\Delta}(t) \geq z^{\Delta}(t) - \bar{S}^{\Delta}(t). \tag{12}$$

Furthermore, $\hat{z}^{\Delta}(t)$ is equal to the $\sum_{u=T+1}^{G} S_u^{\Delta}(t)$ plus some of the $S_T^{\Delta}(t)$ slots. This is because Δ transmits in every busy slot in which the sender and at least $T+1$ receivers are ready, and the quorum is T or $T+1$.

Therefore, the throughput Ω^{Δ} or the reward received per unit time under Δ satisfies the following relation:

$$\Omega^{\Delta} \geq \lim_{t\to\infty} \frac{\sum_{u=T+1}^{G} u S_u^{\Delta}(t)}{t} + \lim_{t\to\infty} \frac{T \max\left\{0, \hat{z}^{\Delta}(t) - \sum_{u=T+1}^{G} S_u^{\Delta}(t)\right\}}{t}. \tag{13}$$

$$\begin{aligned}\text{Now} \quad \lim_{t\to\infty} \frac{\sum_{u=T+1}^{G} u S_u^{\Delta}(t)}{t} &= \lim_{t\to\infty} \frac{\sum_{u=T+1}^{G} u(n_u(t) - \bar{S}_u^{\Delta}(t))}{t} \quad \text{(from (10))}\\ &\geq \lim_{t\to\infty} \frac{\sum_{u=T+1}^{G} u n_u(t)}{t} - \lim_{t\to\infty} \frac{G\bar{S}^{\Delta}(t)}{t} \quad \text{(from (11))}\\ &\geq \lim_{t\to\infty} \frac{\sum_{u=T+1}^{G} u n_u(t)}{t} - \frac{\epsilon}{2} \quad \text{(from (8))}.\end{aligned} \tag{14}$$

Now, we consider the second term in (13).

$$\begin{aligned}&\lim_{t\to\infty} \frac{T \max\left\{0, \hat{z}^{\Delta}(t) - \sum_{u=T+1}^{G} S_u^{\Delta}(t)\right\}}{t}\\ &\geq \lim_{t\to\infty} \frac{T \max\left\{0, \hat{z}^{\Delta}(t) - \sum_{u=T+1}^{G} n_u(t)\right\}}{t}\\ &\geq \lim_{t\to\infty} \frac{T \max\left\{0, z^{\Delta}(t) - \bar{S}^{\Delta}(t) - \sum_{u=T+1}^{G} n_u(t)\right\}}{t}\\ &\text{(from (12))}\\ &\geq \lim_{t\to\infty} \frac{T \max\left\{0, z^{\Delta}(t) - \sum_{u=T+1}^{G} n_u(t)\right\}}{t} - \frac{\epsilon}{2}.\end{aligned} \tag{15}$$

We note that the throughput of any stable policy Δ_1 is bounded as

$$\Omega^{\Delta_1} \leq \lim_{t\to\infty} \frac{\sum_{u=T+1}^{G} u n_u(t)}{t} + \lim_{t\to\infty} \frac{T \max\left\{0, z^{\Delta_1}(t) - \sum_{u=T+1}^{G} n_u(t)\right\}}{t} \tag{16}$$

From relations (5), (9), (14) and (15), (4) follows. ■

B. Proof for ϵ -Throughput Optimality of $\Delta_O(\Gamma)$ Theorem 2

We use the following results to prove Theorem 2.

Lemma 1: If $\lambda < \sum_{u=0}^{G} b_u^R$, then $\Delta_O(\Gamma)$ is stable for every $\Gamma < \infty$.

Lemma 2: The policy Δ_T is stable if

$$\sum_{u=T}^{G} b_u^R > \lambda. \tag{17}$$

Lemma 3: Consider any given $\epsilon > 0$ and a quorum T that satisfies (17). Then, for Δ_T, there exists a value $\Gamma_1(\epsilon, T)$ such that for every $\Gamma \geq \Gamma_1(\epsilon, T)$

$$\sum_{k=\Gamma+1}^{\infty} \pi_k \leq \frac{\epsilon}{4G}. \tag{18}$$

Lemma 4: Consider any given $\epsilon > 0$, a quorum T such that

$$\sum_{u=T}^{G} b_u^R < \lambda \tag{19}$$

and buffer capacity $B = (G-T+1)\Gamma$. Then, for $\Delta_{B,T}$ there exists a value $\Gamma_2(\epsilon, T)$ such that for every $\Gamma \geq \Gamma_2(\epsilon, T)$

$$\sum_{k=0}^{B-\Gamma} \hat{\pi}_k \leq \frac{\epsilon}{4G}. \tag{20}$$

Lemma 5: Let P_{T_-} denote the steady state probability that the queue length Q at the sender is greater than $(G-T+1)\Gamma$ under $\Delta_O(\Gamma)$. If T satisfies (17), then

$$P_{T_-} \leq \sum_{k=\Gamma+1}^{\infty} \pi_k. \tag{21}$$

Lemma 6: Consider buffer capacity $B = (G-T+1)\Gamma$ and let P_{T_+} denote the steady state probability that the queue length Q at the sender is less than or equal to $(G-T)\Gamma$ under $\Delta_O(\Gamma)$. Then

$$P_{T_+} \leq \sum_{k=0}^{(G-T)\Gamma} \hat{\pi}_k. \tag{22}$$

Results in Lemmas 1 and 2 are intuitive. We only present the intuition here. We observe that for any quorum T that satisfies (17), the rate at which the slots with ready sender, and T or more ready receivers arrive is higher than packet arrival rate. Hence, the expected busy period length is finite. Thus, the stability follows. Refer to [26] for the formal proofs.

1) Proof of Theorem 2:

Proof: In view of Theorem 1 and Lemma 1, it suffices to show that there exists a quorum T_O such that (1) is satisfied. Let

$$T_O = \arg\max_{0 \leq T \leq G} \left\{ \sum_{u=T}^{G} b_u^R > \lambda \right\}. \tag{23}$$

If $\lambda < \sum_{u=0}^{G} b_u^R$, then $0 \leq T_O \leq G$ exists.

First, let $0 < T_O < G$.

Now, we have shown earlier that the process $\{Y_n^{\Delta_O(\Gamma)} : n \geq 1\}$ is a DTMC. By Lemma 1, we know that the DTMC is ergodic. Hence,

$$\lim_{t\to\infty} \frac{s_{T_O}^{\Delta_O(\Gamma)}(t) + s_{T_O+1}^{\Delta_O(\Gamma)}(t)}{t} = 1 - P_{(T_O+1)_+} - P_{T_O-} \quad \text{w.p. 1.} \tag{24}$$

Now, we fix $\Gamma \geq \max\{\Gamma_1(\epsilon, T_O), \Gamma_2(\epsilon, T_O+1)\}$. Since T_O satisfies (17) and T_O+1 satisfies (19), from Lemmas 3 and 4

$$\sum_{u=\Gamma+1}^{\infty} \pi_k \leq \frac{\epsilon}{4G} \qquad \text{and} \tag{25}$$

$$\sum_{u=0}^{(G-T_O-1)\Gamma} \widehat{\pi}_k \leq \frac{\epsilon}{4G}. \tag{26}$$

Thus, from Lemmas 5 and 6 the result follows.

Now, if $T_O = G$, then $P_{(T_O+1)_+} = 0$. This is because quorum chosen by $\Delta_O(\Gamma)$ is always less than or equal to $G+1$. Hence,

$$\lim_{t\to\infty} \frac{s_{T_O}^{\Delta_O(\Gamma)}(t) + s_{T_O+1}^{\Delta_O(\Gamma)}(t)}{t} = 1 - P_{T_O-} \quad \text{w.p. 1 (from (24)).}$$

Further, since T_O satisfies (17), relation (25) holds by Lemma 3. Thus, the result follows from Lemma 5.

Now, if $T_O = 0$, then $P_{T_O-} = 0$. This is because quorum chosen by $\Delta_O(\Gamma)$ is always greater than or equal to 0. Hence,

$$\lim_{t\to\infty} \frac{s_{T_O}^{\Delta_O(\Gamma)}(t) + s_{T_O+1}^{\Delta_O(\Gamma)}(t)}{t} = 1 - P_{(T_O+1)_+} \quad \text{w.p. 1} \quad \text{(from (24)).}$$

Further, since $T_O + 1$ satisfies (19), relation (26) holds by Lemma 4. Thus, the result follows from Lemma 4.

Now, we show that if $\lambda > \sum_{u=0}^{G} b_u^R$, then no policy can stabilize the system. Let $\delta = \lambda - \sum_{u=0}^{G} b_u^R$. Observe that $\delta > 0$. We show that the queue length under arbitrary policy Δ becomes unbounded in this case w.p.1. Let $A(t)$ and $z^{\Delta}(t)$ denote the number of arrivals and departures under Δ until time t, then the queue length at time t is $A(t) - z^{\Delta}(t)$. Hence, it suffices to show that

$$\lim_{t\to\infty} \frac{A(t) - z^{\Delta}(t)}{t} \geq \delta \quad \text{w.p. 1.} \tag{27}$$

Note that since the sender can transmit only when it is ready, the total number of departures under any policy is bounded above by the total number slots in which the sender was ready. From the above observation, and ergodicity of the arrival and the readiness processes

$$\lim_{t\to\infty} \frac{A(t)}{t} = \lambda \quad \text{w.p. 1, and} \tag{28}$$

$$\lim_{t\to\infty} \frac{z^{\Delta}(t)}{t} \leq \sum_{u=0}^{G} b_u^R = \lambda - \delta \quad \text{w.p. 1.} \tag{29}$$

Relation (27) follows from (28) and (29). ■

2) Proof for Lemma 3:

Proof: From Lemma 2, we know that if T satisfies (17), then there exists a unique stationary distribution π_k under policy Δ_T. We note that $\sum_{k=0}^{\infty} \pi_k = 1$. Hence, $\lim_{u\to\infty} \sum_{k=u}^{\infty} \pi_k = 0$. Thus, the result follows. ■

3) Proof for Lemma 4:

Proof: We prove this Lemma in three steps. 1) We obtain an upper bound on the expected time required for reaching state $(B, \vec{j})$ for any readiness state $\vec{j}$ starting from empty buffer. 2) We obtain a lower bound on the expected time required to reach queue length $B - \Gamma$ starting from the full buffer given that the queue length $B - \Gamma$ is reached before the system returns to the full buffer state. 3) Using these bounds, we prove (20).

If T satisfies (19), then there exists a $\delta > 0$ such that $\lambda = \sum_{u=T}^{G} b_u^R + \delta$.

Part (a): Let $\widehat{Z}$ denote a r.v. indicating the time required to reach full buffer state starting from queue length zero. We obtain a bound on $\mathbf{E}[\widehat{Z}]$ independent of the initial arrival and readiness state. Let $\widehat{B} = B/\gamma$.

Let $A(t)$ denote the arrivals in the system until time t and let $\widehat{A}(t)$ denote the arrivals admitted in the system. Recall that the arrivals are dropped if the buffer is full. Hence, $\widehat{A}(t) \leq A(t)$. Also, from ergodicity of the arrival process

$$\lim_{t\to\infty} \frac{A(t)}{t} = \lambda \;\; \text{w.p. 1.} \tag{30}$$

Furthermore, let $z^{\Delta_{B,T}}(t)$ denote the number of departures until time t, and let $D^{\Delta_{B,T}}(t)$ denote the number of slots in which the sender and at least T receivers are ready until time t. We note that the policy $\Delta_{B,T}$ transmits a packet in every slot with ready sender and T or more ready receivers, except when the queue is empty. Hence, $z^{\Delta_{B,T}}(t) \leq D^{\Delta_{B,T}}(t)$. Also, from ergodicity of the readiness process

$$\lim_{t\to\infty} \frac{D^{\Delta_{B,T}}(t)}{t} = \sum_{u=T}^{G} b_u^R = \lambda - \delta \;\; \text{w.p. 1.} \tag{31}$$

$$\text{Now for every } k \leq \widehat{B}, \; \mathbf{P}\{\widehat{Z} \geq k\} = 1 \tag{32}$$

since to fill the buffer at least B packets have to arrive and at most γ packets can arrive in any slot. Without loss of generality, we assume that the queue length is zero at time zero. Thus, for any $k \geq 1$

$$\mathbf{P}\{\widehat{Z} \geq \widehat{B} + k\} = \mathbf{P}\left\{\bigcap_{u=0}^{k} \{\widehat{A}(\widehat{B}+u) - z^{\Delta_{B,T}}(\widehat{B}+u) < B\}\right\}. \tag{33}$$

Further, for every $u \leq k$

$$\left\{\omega : \widehat{Z}(\omega) \geq \widehat{B} + k\right\} \subseteq \left\{\omega : \widehat{A}(\widehat{B}+u, \omega) = A(\widehat{B}+u, \omega)\right\}.$$

Hence, from (33) it follows that

$$\begin{aligned} &\mathbf{P}\{\widehat{Z} \geq \widehat{B} + k\} \\ &\leq \mathbf{P}\left\{\bigcap_{u=1}^{k} \{A(\widehat{B}+u) - z^{\Delta_{B,T}}(\widehat{B}+u) < B\}\right\} \\ &\leq \mathbf{P}\left\{A(\widehat{B}+k) - z^{\Delta_{B,T}}(\widehat{B}+k) < B\right\}. \end{aligned} \tag{34}$$

Now, the event $\{A(\widehat{B}+u) - z^{\Delta_{B,T}}(\widehat{B}+u) < B\}$ implies the event $\{A(\widehat{B}+u) - D^{\Delta_{B,T}}(\widehat{B}+u) < B\}$. Let $t = \widehat{B}+k$. Hence, relation (34) becomes

$$\begin{aligned}\mathbf{P}\{\widehat{Z} \geq t\} &\leq \mathbf{P}\left\{A(t) - D^{\Delta_{B,T}}(t) < B\right\}\\ &= \mathbf{P}\left\{\frac{A(t)}{t} - \frac{D^{\Delta_{B,T}}(t)}{t} < \frac{B}{t}\right\}\\ &\leq 1 - \mathbf{P}\left\{\left\{\frac{A(t)}{t} - \lambda \geq \frac{B}{t} - \frac{\delta}{2}\right\}\bigcap\right.\\ &\qquad \left.\left\{\frac{D^{\Delta_{B,T}}(t)}{t} - (\lambda-\delta) < \frac{\delta}{2}\right\}\right\}\\ &\leq \mathbf{P}\left\{\frac{A(t)}{t} - \lambda < \frac{B}{t} - \frac{\delta}{2}\right\}\\ &\quad + \mathbf{P}\left\{\frac{D^{\Delta_{B,T}}(t)}{t} - (\lambda-\delta) \geq \frac{\delta}{2}\right\}.\end{aligned}$$

Now, since $t = \widehat{B}+k$,

$$\text{for every} \quad k > \left(\frac{4}{\delta} - 1\right)B, \quad \frac{B}{t} < \frac{\delta}{4}. \tag{35}$$

Hence, for every k that satisfies (35),

$$\begin{aligned}&\mathbf{P}\{\widehat{Z} \geq t\}\\ &\leq \mathbf{P}\left\{\frac{A(t)}{t} - \lambda < \frac{-\delta}{4}\right\} + \mathbf{P}\left\{\frac{D^{\Delta_{B,T}}(t)}{t} - (\lambda-\delta) \geq \frac{\delta}{2}\right\}\\ &\leq 2e^{-(\widehat{B}+k)C}\end{aligned} \tag{36}$$

where C is a constant independent of the initial arrival and receiver readiness state. The upper bound (36) follows from (30), (31), substituting $t = \widehat{B}+k$, and the large deviation bounds for the finite, ergodic Markov chains [25]. Let $\widehat{k} = (4/\delta - 1)B$. Now,

$$\begin{aligned}\mathbf{E}[\widehat{Z}] &= \sum_{k=1}^{\infty} \mathbf{P}\{\widehat{Z} \geq k\}\\ &= \sum_{k=1}^{\widehat{B}} \mathbf{P}\{\widehat{Z} \geq k\} + \sum_{k=1}^{\widehat{k}} \mathbf{P}\{\widehat{Z} \geq \widehat{B}+k\}\\ &\quad + \sum_{k=\widehat{k}+1}^{\infty} \mathbf{P}\{\widehat{Z} \geq \widehat{B}+k\}\\ &\leq \frac{B}{\gamma} + \frac{4B}{\delta} + 2\sum_{k=1}^{\infty} e^{-(\widehat{B}+k)C}\\ &\quad \text{(since } \mathbf{P}(E) \leq 1 \text{ for every event } E, \text{ and (36))}\\ &= B\left(\frac{1}{\gamma} + \frac{4}{\delta}\right) + \frac{2\gamma e^{-\widehat{B}C}}{C}.\end{aligned} \tag{37}$$

Part (b): Let Z denote a r.v. indicating the time required to reach $B-\Gamma$ starting from full buffer. Without loss of generality, the buffer of the system is full at time zero. To reach the queue length $B-\Gamma$ from B, at least Γ departures must happen. Since at most one packet can depart in a slot, for every $k \leq \Gamma$

$$\mathbf{P}\{Z \geq k\} = 1. \tag{38}$$

For $k \geq 1$, $\mathbf{P}\{Z \geq \Gamma + k\}$

$$\begin{aligned}&= \mathbf{P}\left\{\bigcap_{u=0}^{k}\{z^{\Delta_{B,T}}(\Gamma+u) - \widehat{A}(\Gamma+u) < \Gamma\}\right\}\\ &= 1 - \mathbf{P}\left\{\bigcup_{u=0}^{k}\{z^{\Delta_{B,T}}(\Gamma+u) - \widehat{A}(\Gamma+u) \geq \Gamma\}\right\}\\ &\geq 1 - \min\left\{1, \sum_{u=0}^{k}\mathbf{P}\{z^{\Delta_{B,T}}(\Gamma+u) - \widehat{A}(\Gamma+u) \geq \Gamma\}\right\}.\end{aligned} \tag{39}$$

The sender's buffer is never empty until time Z as it is the first time the sender's queue length becomes $B-\Gamma$ starting from B. Hence, $z^{\Delta_{B,T}}(\Gamma+u) = D^{\Delta_{B,T}}(\Gamma+u)$. Let $Q(m)$ denotes the sender's queue length in slot m. Now, $Q(0) = B$. Also,

$$\begin{aligned}&\mathbf{P}\{z^{\Delta_{B,T}}(\Gamma+u) - \widehat{A}(\Gamma+u) \geq \Gamma\}\\ &= \mathbf{P}\left\{\bigcup_{v=0}^{u}\left\{\{D^{\Delta_{B,T}}(\Gamma+u) - \widehat{A}(\Gamma+u) \geq \Gamma\}\bigcap\right.\right.\\ &\qquad \{Q(v) = B, Q(m) < B : \forall m \in \{v+1, \Gamma+u\}\} \qquad (40)\\ &\leq \sum_{v=0}^{u}\mathbf{P}\left\{\{D^{\Delta_{B,T}}(\Gamma+u) - \widehat{A}(\Gamma+u) \geq \Gamma\}\bigcap\right.\\ &\qquad \left.\{Q(v) = B, Q(m) < B : \forall m \in \{v+1, \Gamma+u\}\}\right\}. \qquad (41)\end{aligned}$$

Relation (40) follows from the observation that there should be at least Γ contiguous loss free slots (queue length $< B$) at the end of interval $[0, Z]$ for the difference between the departures and arrivals in $[0, Z]$ to be greater than Γ. Furthermore, if $Q(m) = B$ for some m, then the difference between the number of arrivals and departures in $[0, m]$ is 0, since $Q(0) = B$. Now, (41) follows from union bound property. Furthermore, if $Q(v) = B$ and $Q(m) < B, \forall m \in \{v+1, \Gamma+u\}$, then

$$\begin{aligned}&D^{\Delta_{B,T}}(\Gamma+u) - \widehat{A}(\Gamma+u)\\ &\quad = D^{\Delta_{B,T}}(\Gamma+u) - \widehat{A}(\Gamma+u) - (D^{\Delta_{B,T}}(v) - \widehat{A}(v))\\ &\quad = (D^{\Delta_{B,T}}(\Gamma+u) - D^{\Delta_{B,T}}(v)) - (A(\Gamma+u) - A(v)).\end{aligned}$$

From the stationarity of the arrival and readiness processes, for every k

$$\begin{aligned}&\mathbf{P}\{A(\Gamma+u) - A(v) = k\}\\ &\quad = \mathbf{P}\{A(\Gamma+u-v) = k\} \quad \text{and} \qquad (42)\\ &\mathbf{P}\{D^{\Delta_{B,T}}(\Gamma+u) - D^{\Delta_{B,T}}(v) = k\}\\ &\quad = \mathbf{P}\{D^{\Delta_{B,T}}(\Gamma+u-v) = k\}. \qquad (43)\\ &\mathbf{P}\{z^{\Delta_{B,T}}(\Gamma+u) - \widehat{A}(\Gamma+u) \geq \Gamma\}\\ &\quad \leq \sum_{v=1}^{u}\mathbf{P}\left\{D^{\Delta_{B,T}}(\Gamma+u-v) - A(\Gamma+u-v) \geq \Gamma\right\}\\ &\qquad \text{(from (42) and (43))} \qquad (44)\\ &\quad = \sum_{v=0}^{u}\mathbf{P}\left\{D^{\Delta_{B,T}}(\Gamma+v) - A(\Gamma+v) \geq \Gamma\right\}. \qquad (45)\end{aligned}$$

Let $t = \Gamma + v$ and consider

$$\begin{aligned}
&\mathbf{P}\{D^{\Delta_{B,T}}(t) - A(t) \geq \Gamma\} \\
&= \mathbf{P}\left\{\frac{D^{\Delta_{B,T}}(t)}{t} - \frac{A(t)}{t} \geq \frac{\Gamma}{t}\right\} \\
&= \mathbf{P}\left\{\left(\frac{D^{\Delta_{B,T}}(t)}{t} - (\lambda - \delta)\right) - \left(\frac{A(t)}{t} - \lambda\right) \geq \delta + \frac{\Gamma}{t}\right\} \\
&\leq \mathbf{P}\left\{\frac{D^{\Delta_{B,T}}(t)}{t} - (\lambda - \delta) \geq \frac{\delta}{2}\right\} + \mathbf{P}\left\{\frac{A(t)}{t} - \lambda \leq \frac{-\delta}{2}\right\}.
\end{aligned} \tag{46}$$

From (30), (31), and large deviation bound for finite ergodic MC's [25] we obtain

$$\mathbf{P}\left\{\left(\frac{A(t)}{t} - \lambda\right) \leq \frac{-\delta}{2}\right\} \leq e^{-tC_1} \quad \text{and} \tag{47}$$

$$\mathbf{P}\left\{\left(\frac{D^{\Delta_{B,T}}(t)}{t} - (\lambda - \delta)\right) > \frac{\delta}{2}\right\} \leq e^{-tC_2} \tag{48}$$

where $C_1 > 0$ and $C_2 > 0$ are constants independent of Γ and the initial arrival and readiness state. Let $\widehat{C} = \min\{C_1, C_2\}$. Hence, from (46), (47) and (48)

$$\mathbf{P}\{D^{\Delta_{B,T}}(t) - A(t) > \Gamma\} \leq 2e^{-t\widehat{C}}. \tag{49}$$

$$\begin{aligned}
\mathbf{P}\{z^{\Delta_{B,T}}(\Gamma + u) - \widehat{A}(\Gamma + u) > \Gamma\} &\leq \sum_{v=0}^{u} 2e^{-t\widehat{C}} \quad \text{(from (45) and (49))} \\
&= \sum_{v=0}^{u} 2e^{-(\Gamma+v)\widehat{C}} \quad (\text{since } t = \Gamma + v) \\
&\leq \frac{2e^{-\Gamma\widehat{C}}}{\widehat{C}}.
\end{aligned} \tag{50}$$

Using (50) in (39), we obtain

$$\mathbf{P}\{Z \geq \Gamma + k\} \geq 1 - \min\left\{1, \frac{2(k+1)e^{-\Gamma\widehat{C}}}{\widehat{C}}\right\}. \tag{51}$$

$$\begin{aligned}
\min\left\{1, \frac{2(k+1)e^{-\Gamma\widehat{C}}}{\widehat{C}}\right\} &= \frac{2ke^{-\Gamma\widehat{C}}}{\widehat{C}} \quad \forall k < \frac{\widehat{C}}{2}e^{\Gamma\widehat{C}} - 1 \\
&= 1 \quad \forall k \geq \frac{\widehat{C}}{2}e^{\Gamma\widehat{C}} - 1.
\end{aligned} \tag{52}$$

Thus,

$$\begin{aligned}
\mathbf{E}[Z] &\geq \Gamma + \sum_{k=1}^{\infty}\left(1 - \min\left\{1, \frac{2ke^{-\Gamma\widehat{C}}}{\widehat{C}}\right\}\right) \quad \text{(from (38))} \\
&\geq \Gamma + \sum_{k=1}^{\left\lfloor \widehat{C}/2e^{\Gamma\widehat{C}} \right\rfloor}\left(1 - \frac{2ke^{-\Gamma\widehat{C}}}{\widehat{C}}\right) \quad \text{(from (51) and (52))} \\
&\geq \Gamma + \frac{\widehat{C}}{4}e^{\Gamma\widehat{C}} - \frac{1}{2}.
\end{aligned} \tag{53}$$

Part (c): Since the system has finite buffer, $\{\widehat{Y}_n^{\Delta_B} : n \geq 1\}$ is a positive recurrent and ergodic DTMC. Now, we consider the epochs at which the queue length is B. Let us call a time duration between two successive epochs as a *cycle length*. Let $m(t)$ denote the number of cycles completed until time t, and $\widehat{m}(t)$ denote the number of cycles until time t in which queue length becomes less than or equal to $B - \Gamma$. Let X_u denote a random variable indicating the length of the uth cycle. Furthermore, let X_{u+} and X_{u-} denote the random variables indicating the time spent in the states with queue length $> B - \Gamma$ and the states with queue length $\leq B - \Gamma$, respectively. By ergodicity

$$\sum_{k=0}^{B-\Gamma} \pi_k = \lim_{t\to\infty} \frac{\sum_{u=1}^{m(t)} X_{u-}}{t} \quad \text{w.p. 1.}$$

Since $t \geq \sum_{u=1}^{m(t)} X_u$, $\quad \sum_{k=0}^{B-\Gamma} \pi_k \leq \lim_{t\to\infty} \frac{\sum_{u=1}^{m(t)} X_{u-}}{\sum_{u=1}^{m(t)} X_u}$ w.p. 1.

We note that $X_{u-} = 0$ if the queue length is always greater than $B - \Gamma$ in the uth cycle, then

$$\sum_{k=0}^{B-\Gamma} \pi_k \leq \lim_{t\to\infty} \frac{\sum_{i=1}^{\widehat{m}(t)} X_{u_i-}}{\sum_{i=1}^{\widehat{m}(t)} X_{u_i}} \quad \text{w.p. 1}$$

where u_i denote the subsequence of cycles in which the queue length becomes $\leq B - \Gamma$. Now we note that $X_{u_i-} \leq \widehat{Z}$ and $X_{u_i} \geq X_{u_i+} \geq Z$ for every i. Furthermore, since the DTMC is positive recurrent, we conclude the following: (a) $\widehat{m}(t) \to \infty$ w.p. 1 as $t \to \infty$ (b) $\mathbf{E}[Z] < \infty$ and $\mathbf{E}[\widehat{Z}] < \infty$. Hence, by Kolmogorov's Strong Law of Large Numbers

$$\sum_{k=0}^{B-\Gamma} \pi_k \leq \frac{\mathbf{E}[\widehat{Z}]}{\mathbf{E}[Z]} \quad \text{w.p. 1.} \tag{54}$$

From (37), (53), and (54)

$$\sum_{k=0}^{B-\Gamma} \widehat{\pi}_k \leq \frac{B\left(\frac{1}{\gamma} + \frac{4}{\delta}\right) + \frac{2\gamma e^{-\widehat{B}C}}{C}}{\Gamma + + \frac{\widehat{C}}{4}e^{\Gamma\widehat{C}} - \frac{1}{2}}.$$

Thus, $\quad \lim_{\Gamma\to\infty} \sum_{k=0}^{B-\Gamma} \widehat{\pi}_k = 0.$

This proves the required result. ■

4) Intuition for Lemmas 5 and 6: We prove Lemma 5 by showing that the sender's queue length under $\Delta_O(\Gamma)$ is greater than $(G - T + 1)\Gamma$ implies that the queue length under Δ_T is greater than Γ on every sample path. Thus, the steady state probability that the sender's queue length is greater than $(G - T + 1)\Gamma$ under $\Delta_O(\Gamma)$ is less than or equal the steady state probability that the sender's queue length is greater than Γ under Δ_T. Now, the result follows from Lemma 3.

To show the required, we note that if the sender's queue length under $\Delta_O(\Gamma)$ exceeds $(G - T)\Gamma$, then the the quorum under $\Delta_O(\Gamma)$ is less than or equal T. Since the quorum under Δ_T is always T, we conclude that the quorum under $\Delta_O(\Gamma)$ is less than or equal to the quorum under Δ_T if the sender's queue length under $\Delta_O(\Gamma)$ exceeds $(G-T)\Gamma$. Hence, when the queue length under $\Delta_O(\Gamma)$ exceeds $(G - T)\Gamma$, $\Delta_O(\Gamma)$ also transmits in the slots in which Δ_T transmits. Now, consider a slot t such that the queue length under $\Delta_O(\Gamma)$ becomes $(G-T)\Gamma$. In t, the queue length under Δ_T is at least 0. Since the arrivals for $\Delta_O(\Gamma)$ and Δ_T are the same, the queue length under $\Delta_O(\Gamma)$ exceeds $(G - T + 1)\Gamma$ only if the queue length under Δ_T exceeds Γ. Refer to [26] for the formal proof.

We prove Lemma 6 by showing that the sender's queue length under $\Delta_O(\Gamma)$ is greater than or equal to that under $\Delta_{B,T}$ in every slot on every sample path. Thus, the steady state probability that the sender's queue length is less than or equal to $(G-T)\Gamma$ under $\Delta_O(\Gamma)$ is less than or equal that under $\Delta_{B,T}$. Now, the result follows from Lemma 4. Recall that $B = (G-T+1)\Gamma$. To show the required, we note that if the queue length under $\Delta_O(\Gamma)$ is less than B, then the quorum under $\Delta_O(\Gamma)$ is greater than or equal to T. Thus, for these values of queue lengths $\Delta_{T,B}$ transmits in every slot in which $\Delta_O(\Gamma)$ transmits. Moreover, the queue length under $\Delta_{T,B}$ can at most be B. Thus, the result follows. Refer to [26] for the formal proof.

REFERENCES

[1] S. Marti, T. Giuli, K. Lai, and M. Baker, "Mitigating routing misbehaviour in mobile ad hoc networks," in *Proc. ACM MobiCom'00*, Boston, MA, 2000, pp. 255–265.

[2] S. Mueller and D. Ghosal, "Multipath routing in mobile ad hoc networks: Issues and challenges," in *Performance Tools and Application to Networked Systems*, M. C. Calzarossa and E. Gelenbe, Eds. Berlin: Springer, 2004, vol. 2965, Lecture Notes in Computer Science, pp. 209–234.

[3] D. Ganesan, R. Govindan, S. Shenker, and D. Estrin, "Highly resilient, energy efficient multipath routing in wireless sensor networks," *ACM SIGMOBILE Mobile Comput. Commun. Rev.*, vol. 5, no. 4, pp. 11–25, Oct. 2001.

[4] Z. J. Haas, J. Y. Halpern, and L. Li, "Gossip-based ad hoc routing," in *Proc. IEEE INFOCOM*, New York, Jun. 2002, vol. 3, pp. 1707–1716.

[5] P. Samar and Z. J. Haas, "Strategies for broadcasting updates by proactive routing protocols in mobile adhoc networks," in *Proc. IEEE MILCOM'02*, Anaheim, CA, Oct. 2002, vol. 2, pp. 873–878.

[6] C. Intanagonwiwat, R. Govindan, and D. Estrin, "Directed diffusion: A scalable and robust communication paradigm for sensor networks," in *Proc. ACM MobiCom'00*, Boston, MA, 2000, pp. 56–67.

[7] D. Braginsky and D. Estrin, "Rumor routing algorithm for sensor networks," presented at the 1st Workshop on Sensor Networks and Applications (WSNA), Atlanta, GA, Sep. 2002.

[8] V. D. Park and J. P. Macker, "Anycast routing for mobile networking," in *Proc. IEEE MILCOM'99*, Vienna, VA, Oct. 1999, vol. 1, pp. 1–5.

[9] T. Gopalsamy, M. Singhal, D. Panda, and P. Sadayappan, "A reliable multicast algorithm for mobile ad hoc networks," in *Proc. IEEE ICDCS'02*, Vienna, Austria, Jul. 2002, pp. 563–570.

[10] J. Wieselthier, G. Nguyen, and A. Ephermides, "On the construction of energy-efficient broadcast and multicast trees in wireless networks," in *Proc. IEEE INFOCOM*, Tel Aviv, Israel, Mar. 2000, vol. 2, pp. 585–594.

[11] S. Singh, C. S. Raghavendra, and J. Stepanek, "Power efficient broadcasting in mobile ad hoc networks," presented at the PIMRC'99, Osaka, Japan, Sep. 1999.

[12] K. Wang, C. Chiasserini, J. Proakis, and R. Rao, "Joint scheduling and power control for multicasting in wireless ad hoc networks," in *Proc. IEEE VTC'03-Fall*, Orlando, FL, Oct. 2003, vol. 5, pp. 2915–2920.

[13] C. Jaikaeo and C. Shen, "Multicast communication in ad hoc networks with directional antennas," *Telecommun. Syst.*, vol. 28, no. 3–4, pp. 297–316, 2005.

[14] J. Kuri and S. Kasera, "Reliable multicast in multi-access wireless LANs," *Wireless Netw.*, vol. 7, pp. 359–369, Jul. 2001.

[15] K. Tang and M. Gerla, "MAC reliable broadcast in ad hoc networks," in *Proc. IEEE MILCOM'01*, Vienna, VA, Oct. 2001, vol. 2, pp. 1008–1013.

[16] ——, "MAC layer broadcast support in 802.11 wireless networks," in *Proc. IEEE MILCOM'00*, Los Angeles, CA, Oct. 2000, vol. 1, pp. 544–548.

[17] ——, "Random access MAC for efficient broadcast support in ad hoc networks," in *Proc. IEEE WCNC'00*, Sep. 2000, vol. 1, pp. 454–459.

[18] L. H. Min-Te Sun, A. Arora, and T.-H. Lai, "Reliable MAC layer multicast in ieee 802.11 wireless networks," presented at the ICPP'02, Vancouver, B.C., Canada, Aug. 2002.

[19] R. Chandra, V. Ramasubramaniam, and P. Birman, "Anonymous gossip: Improving multicast reliability in mobile ad hoc networks," in *Proc. IEEE ICDCS'01*, Phoenix (Mesa), AZ, Apr. 2001, p. 275.

[20] E. Pagani and G. Rossi, "Reliable broadcast in mobile multihop packet networks," in *Proc. ACM/IEEE MobiCom'97*, Budapest, Hungary, Sep. 1997, pp. 34–42.

[21] G. Fayolle, V. A. Malyshev, and M. V. Menshikov, *Topics in the Constructive Theory of Countable Markov Chains*. Cambridge, U.K.: Cambridge Univ. Press, 1995.

[22] L. Tassiulas and A. Ephremides, "Stability properties of constrained queueing systems and scheduling for maximum throughput in multihop radio networks," *IEEE Trans. Inf. Theory*, vol. 37, no. 12, pp. 1936–1949, Dec. 1992.

[23] S. Sarkar and L. Tassiulas, "A framework for routing and congestion control for multicast information flows," *IEEE Trans. Inf. Theory*, vol. 48, no. 10, pp. 2690–2708, Oct. 2002.

[24] T. Nandagopal, T. Kim, X. Gao, and V. Bharghavan, "Achieving MAC layer fairness in wireless packet networks," in *Proc. ACM Mobicom'00*, Boston, MA, Aug. 2000, pp. 87–98.

[25] A. Shwartz and A. Weiss, *Large Deviations for Performance Analysis: Queues, Communications, and Computing*. Cleveland, OH: CRC Press, 1995.

[26] P. Chaporkar, S. Sarkar, A. Bhat, and R. R. Shetty, "Dynamic strategies for maximizing throughput in multiparty MAC," Univ. Pennsylvania, Philadelphia, Tech. Rep., 2004 [Online]. Available: http://www.seas.upenn.edu/~swati/publication.htm

Prasanna Chaporkar received the B.E. degree in electronics from Walchand College of Engineering, Sangli, India, in 1996, the M.S. degree in engineering from the Indian Institute of Science, Bangalore, in 2000, and the Ph.D. degree from the Department of Electrical and Systems Engineering at the University of Pennsylvania, Philadelphia, in 2005.

He is currently a Postdoctoral Fellow at TREC, INRIA, Paris, France. His research interests are in resource allocation in communication networks, stochastic control, and queueing theory.

Saswati Sarkar (S'98–M'00) received the M.Eng. degree in electrical communication engineering from the Indian Institute of Science, Bangalore, in 1996, and the Ph.D. degree in electrical and computer engineering from the University of Maryland, College Park, in 2000.

From 2000 to 2006, she was an Assistant Professor in the Department of Electrical and Systems Engineering, University of Pennsylvania, and is currently an Associate Professor. Her research interests are in resource allocation and performance analysis in communication networks.

Dr. Sarkar received the Motorola Gold Medal for the Best Master's Student in the Division of Electrical Sciences at the Indian Institute of Science, and a National Science Foundation (NSF) Faculty Early Career Development Award in 2003. She was an associate editor of IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS from 2001 to 2006.

Rahul Shetty received the B.E. degree in electronics and telecommunication from Manipal Institute of Technology, Manipal, India, in 1998, and the M.S.E. degree in telecommunication and network engineering from the University of Pennsylvania, Philadelphia, in 2004.

He is currently a Solution Architect for GTL Limited, India. His work is focused on designing and implementing network security, forensic analysis, incident handling, and penetration testing.

Mr. Shetty currently holds CCNA, CCNP, CEH and ITIL certifications.