Dynamic Quorum Policy for Maximizing Throughput in Limited Information Multiparty MAC

Prasanna Chaporkar, Saswati Sarkar, Rahul Shetty
Department of Electrical and Systems Engineering
University of Pennsylvania
{prasanna, swati, rshetty}@seas.upenn.edu

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 adhoc networks.

Index Terms

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

[1]This research was supported in part by NSF grants ANI01-06984 and NCR02-38340. Parts of this paper were presented at Wiopt 2004, March, Cambridge, England and MOBIHOC 2004, May, Tokyo, Japan.

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*.

Communicating Routing Updates: Routing updates are communicated through limited flooding [4], [5], [6], [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 significantly benefit 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 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. Thus S_1 needs to decide whether it should transmit even when R_2 is not ready, or it should wait till 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.*

*Paths may be link-disjoint, node-disjoint, and braided or partially-disjoint.

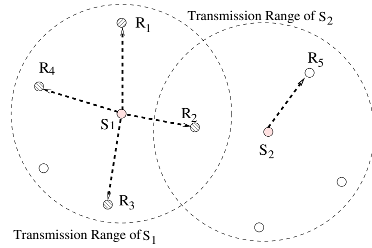


Fig. 1. An example to demonstrate the advantages and the challenges associated with wireless multiparty MAC. The figure shows two senders S_1, S_2 and 5 receivers R_1 to R_5. R_1 to R_4 are S_1's receivers, and R_5 is S_2's receiver. Dashed circle indicates the communication range of a sender.

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 adhoc networks, as it uses (a) simple computations, (b) no information about system statistics, (c) limited control message exchange and (d) limited information about its neighbors. We prove that the quorum-policy maximizes throughput among all policies that stabilize the system (appendix). A sender's optimal quorum value depends only on its queue length and its transmission decisions depend on the the number of its ready receivers. The first quantity is easily available at a sender. In Section V, we propose MAC protocols 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 multiparty 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 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 also proposed the *quorum*-1 *multiparty* scheme where a sender transmits a packet whenever at least one receiver is ready to receive [16], [17]. Sun *et al.* have proposed the *full-quorum* multiparty policy, where a sender transmits only when all receivers are ready [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. The full-quorum multiparty policy exploits the broadcast nature of wireless medium, but has a small stability region as it transmits rarely. This is because the receivers are seldom ready simultaneously, except when the network has very light load. 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 [19], [20], [11], [10], transmitting routing updates [4], [5], [6], [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 initially assume that each packet can be transmitted in a single slot; we relax this assumption in Section IV-C.

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. 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 [21], [22]. 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 Figure 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

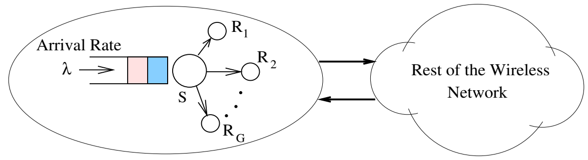


Fig. 2. Figure shows a MAC layer multiparty session and its interaction with the rest of the network. Node S is a sender, where packets arrive at the rate λ. Sender S transmits the packets to receivers R_1 to R_G.

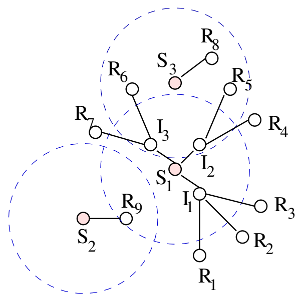


Fig. 3. Figure shows a multicast session from S_1 to receivers R_1 to R_7 and two unicast sessions from S_2 to R_9 and S_3 to R_8 in a multi-hop wireless network. First we observe that a single network layer multicast session corresponds to many MAC layer multiparty sessions, e.g., using multiparty communication 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.

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, 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 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.

a) 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 [21] .

b) 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.

c) 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. In 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 traded 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.

C. Mathematical framework and system objectives for dynamic multiparty MAC based on active information exchange

We present a mathematical framework to model the active information exchange scenario. Figure 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 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. This 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_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 Figure 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 l^{th} 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 controlled by selecting appropriate backoff intervals (e.g., as in [23]) 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 steady state 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.

This class includes *offline* policies that decide transmissions based on knowledge of packet arrivals and readiness vectors in each 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.

We next 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 till 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 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 $\vec{j} = [j_0 \; j_1 \; \dots \; j_G]$ that represents the readiness state. Note that the MC has a finite number of states, since $j_l \in \{0, 1\}$ for every $l \in \{0, \dots, G\}$. 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. 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,

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

In general, from ergodicity we cannot conclude anything about the rate of convergence of the empirical distribution $\lim_{t\to\infty} \frac{n_u(t)}{t}$ to the steady state distribution b_u^R. But for finite, aperiodic MC's, empirical distribution converges to the steady state distribution exponentially fast [24], [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.*

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}.

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 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 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 can not 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_0. 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 samples 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 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 adhoc 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.

From Theorem 3, 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 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.

C. *Generalizations of Analytical Results*

The optimality of $\Delta_O(\Gamma)$ also holds under the following more general scenarios.

- 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. Note that now $j_{l,i}$ may neither be 0 nor 1, but is a real number between 0 and 1. Probabilistic readiness states occur in presence of fading.

  Let $r_l = \sum_{i=1}^{G} j_{l,i}$. Note that r_l is the expected reward the sender would achieve if it transmits in readiness state $[j_{l,1}, \ldots, j_{l,G}]$. Let $R_0, \ldots, R_m$ denote the distinct values r_l can take. Without loss of generality we assume that $0 \leq R_u \leq R_{u+1}$ for every $u \in \{0, \ldots, m-1\}$. Furthermore, let $\widehat{b}_u^R$ denote the steady state probability of the sender being ready and the expected achievable reward being R_u. Now we describe a transmission policy $\Delta'_O(\Gamma)$ which is a generalization of $\Delta_O(\Gamma)$ for the generalized readiness process which is described above. $\Delta'_O(\Gamma)$ transmits a packet if (a) the sender is ready to transmit, (b) the expected number of ready receivers is greater than or equal to the quorum, and (c) the sender has a packet to transmit. Let Q denote the queue length at the sender and let parameter Γ be a fixed positive integer. For $1 \leq T \leq m$, the quorum is R_T if $(m-T)\Gamma < Q \leq (m-T+1)\Gamma$ and quorum is R_0 if $Q > m\Gamma$.

  Theorem 4: If the arrival rate λ is less than the steady state probability that the sender is ready ($\sum_{u=0}^{m} \widehat{b}_u^R$), 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 $\lambda > \sum_{u=0}^{G} \widehat{b}_u^R$.

- $\Delta_O(\Gamma)$ maximizes throughput even when lengths of the packets are arbitrary iid random variables. When the packet transmissions span multiple slots, we assume that the receiver readiness states do not change in the duration of packet transmissions. This assumption is justified if the MAC layer protocol reserves the medium for packet transmission duration. In particular, this assumption holds for the MAC layer protocols we propose in Section V. Let $\mathbf{E}[V]$ denote the expected packet length.

  Theorem 5: If the steady state arrival rate is less than $\frac{\sum_{u=0}^{G} b_u^R}{\mathbf{E}[V]}$, then for every $\epsilon > 0$ there exists Γ_O^g such that $\Delta_O(\Gamma)$ is ϵ-throughput optimal for every $\Gamma \geq \Gamma_O^g$. Formally, $\Omega_{opt} - \Omega^{\Delta_O(\Gamma)} \leq \epsilon$ w.p. 1. Further, no policy is stable if $\lambda > \frac{\sum_{u=0}^{G} b_u^R}{\mathbf{E}[V]}$. Moreover, if steady state arrival rates λ_1 and λ_2 and the expected packet sizes $\mathbf{E}[V_1]$ and $\mathbf{E}[V_2]$ satisfy $\lambda_1 \mathbf{E}[V_1] = \lambda_2 \mathbf{E}[V_2]$, then the throughput under $\Delta_O(\Gamma)$ in the system with λ_1 and $\mathbf{E}[V_1]$ is equal to that in the system with λ_2 and $\mathbf{E}[V_2]$.

- If a sender's queue length is greater than $G\Gamma$, then $\Delta_O(\Gamma)$ has quorum 0. Thus, the sender may transmit even when no receiver is ready. There exists a policy $\widehat{\Delta}_O(\Gamma)$ that maximizes throughput subject to stability among all policies that do not transmit unless at least one receiver is ready. $\widehat{\Delta}_O(\Gamma)$ is similar to $\Delta_O(\Gamma)$, except that it selects quorum T if $(G-T)\Gamma < Q \leq (G-T+1)\Gamma$, for $T > 1$, and quorum 1 if $Q > (G-1)\Gamma$. Now, $\widehat{\Delta}_O(\Gamma)$'s and $\Delta_O(\Gamma)$'s transmission rules are the same once the quorums are selected. $\widehat{\Delta}_O(\Gamma)$ maximizes throughput subject to stability among all policies that do not transmit unless at least one receiver is ready.

  Definition 10: $\mathcal{C}$ 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 6: 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 6 is similar to that for Theorem 2. Like Theorem 2, Theorem 6 hold for any ergodic readiness process (not necessarily Markovian) for which the empirical distribution converges to the steady state distribution at exponential rate.

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 two different approaches 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 can not decide whether to transmit. Also, other nodes will not know whether to defer their transmissions.

We now propose a sequential CTS (SCTS) protocol so as to prevent the collision of the CTS's (Figure 4). Sun *et al.* have used a similar design principle [18]. 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. If the maximum difference between the propagation delays to the receivers is less than $s/2$, then the CTS's will not collide. We formally state this in the following theorem.

Theorem 7: If the maximum difference between the propagation delays to the receivers is less than $s/2$, then the CTS responses from the receivers under the SCTS protocol will not collide at sender under. Furthermore, the CTS responses from the receivers under the SCTS protocol will arrive at the sender in order of their sequence number.

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 $\hat{\Delta}_O(\Gamma)$). Note that the protocol does not need clock synchronization.

Each RTS has a duration/ID field with value $P + Gc + (G+1)s$, where P is the duration of the packet. The i^{th} 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 NAV duration.

In unicast case if the RTS-CTS exchange is successful then the sender always transmits a data packet. But 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. Subsequently, the receivers also transmit release messages sequentially. 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 iid (uniform) random interval.

This scheme has several advantages. It requires only a minor modification of IEEE 802.11, and reduces to IEEE 802.11 when a sender has one receiver. Thus, it can co-exist with IEEE 802.11, i.e., unicast and multiparty senders can implement IEEE 802.11 and SCTS respectively. Its disadvantage is that it requires $O(cG)$ control overhead per data packet.

We now sketch a busy-tone based scheme [26] in which the overhead is independent of G (Figures 5 and 6). Each system has 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 party that are ready to receive send a busy-tone. Since power is additive, a sender estimates the number of ready receivers by measuring the power of the busy-tone signal [27]. If this estimate is greater than the quorum, 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 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

Procedure Multiparty_MAC()
begin

P,c,s: durations of a data packet, CTS, SIFS respectively.

MAC at the sender node

if (sender's back-off timer > 0) **then**
  decrement back-off timer
else
  if (sender's queue length > 0 and the sender is ready) **then**
    /* Sender is ready if there is no ongoing transmission in its neighborhood */
    Transmit an RTS message with duration/ID field value $P + Gc + (G + 1)s$;
    /* Nodes in the sender's neighborhood set their NAV accordingly */
    Wait for CTS responses for duration $Gc + (G + 1)s$;
    if (Number of CTS messages received $\geq$ Quorum value) **then**
      /* Quorum is determined by $\Delta_O(\Gamma)$ */
      Transmit a packet;
    else
      Transmit a Release message;
      /* Nodes in the neighborhood of the sender reset their NAV */
    Choose an integer I uniformly from $\{1, \dots, CW\}$;
    Set back-off timer for time duration $I \times slot\_length$;

MAC at receiver node i

if (RTS is received and the node is ready) **then**
  Wait for duration $ic + (i + 1)s$;
  Transmit CTS message with duration/ID field value $P + (G - i)s + (G - i - 1)c$;
  /* Nodes in the neighborhood of the receiver set their NAV based on duration/ID field value */
if (Release message is received from the sender) **then**
  Transmit a Release message;
  /* Nodes in the neighborhood of the receiver reset their NAV */

end

Fig. 4. Pseudo code for SCTS MAC protocol

busy tone once they receive the release signal. Refer to Figures 5 for pseudo code.

The receivers estimate their distances from the sender using the power level of the received RTS, assuming that all the senders transmit RTS at the same power. Subsequently, the receivers regulate the busy tone power so that all receivers' busy tones reach the sender at the same power level. This resolves the "near-far effect".

The receivers' power consumption increase due to the transmission of busy tone throughout the data transmission. Also, a receiver can transmit a busy tone while receiving the data only if it has two radios. These problems can be addressed with an additional control channel [28]. Receivers can send a short CTS tone on one control channel before receiving the data, receive data, and finally send a short ACK tone on another control channel. A node will not initiate any communication till it receives ACK tone for every CTS tone it received.

The busy tone scheme is not totally compatible with IEEE 802.11. It requires power control and power measurement. 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 13). Further, future extensions for IEEE 802.11 are likely to support these functionalities [29]. Nevertheless, the scheme will require additional hardware and protocol changes. Thus the choice between the SCTS and busy tone schemes can be made on a case by case basis. Now, we compare the proposed protocols with the already existing MAC layer multiparty protocols in multi-hop wireless networks.

A. Comparison with the Existing Protocols

The protocol in [16] extends the IEEE 802.11 for broadcast, i.e., no RTS/CTS handshake. In this protocol, when there is a packet to transmit, the sender broadcasts RTS and waits for CTS for WAIT_FOR_CTS time units. If a node receives RTS frame and it is ready, then it sends back CTS, and waits for WAIT_FOR_DATA time units. If the sender receives CTS frame before WAIT_FOR_CTS timer expires, it transmits data frame. If the sender does not receive a CTS frame before the timer expires, it backs off and repeat the same procedure after the back off duration. Note that the sender transmits DATA if it receives at least one CTS, hence it implements quorum-1 multiparty. The proposed protocol has following limitations. Under this protocol there is no co-ordination among the receivers while transmitting CTS, and hence these messages can collide. To address this problem, the authors assume that the sender has Direct Sequence (DS) capture capability. But as pointed out in [18], for capture to work, the strongest CTS's signal to noise and interference ratio (SINR) needs to be at least 10dB. If there are two nodes sending CTS with the same power, then it requires that one node has to be at least 1.5 times far away from the sender than the other node. Furthermore, when multiple nodes are transmitting such SINR is difficult to achieve. Hence, this protocols works only in limited topologies.

In [17], the authors augment protocol in [16] by introducing NAK frames. This protocol is called Broadcast Support Multiple Access (BSMA). Here, a receiver transmits a NAK frame if it does not receive correct DATA frame before WAIT_FOR_DATA timer expires. After transmitting DATA frame, the sender waits for WAIT_FOR_NAK timer. If NAK frame arrives in this duration, then the sender retransmits the DATA frame after random back-off; otherwise the sender assumes that the packet is delivered successfully. Like in [16], here also no co-ordination among the receivers is maintained, and DS capture is used to receive CTS and NAK frames at the sender. Thus, BSMA also suffers from the same limitations as that in [16].

In [15], the multiparty transmissions are carried out as G unicasts, where G is the size of party. This protocol is called Broadcast Medium Window (BMW). In [18], the authors propose a reliable multiparty MAC protocol called Batch Mode Multicast MAC (BMMM). BMMM does not require DS capture. In this protocol, the sender transmits RTS to each of the party members sequentially. After each RTS transmission, it waits for CTS from the receiver. If it receives at least one CTS, then it transmits the DATA frame. After transmitting DATA, the sender requests

ACK frames from each receiver by transmitting RAK (Request for ACK) frame sequentially to each receiver. If the party has two members, then the complete transmission sequence will be as follows.

RTS1—CTS1—RTS2—CTS2—DATA—RAK1—ACK1—RAK2—ACK2.

If the ACKs from certain receivers are missing then BSMA retransmits the DATA frame using the same hand-shake described above to those receivers. The DATA frame is transmitted multiple times till all the receivers successfully receives it. Because of multiple transmissions, the energy consumption is high, and also the network load is high.

Note that the BMMM can also be used to implement the decision mechanism $\Delta_O(\Gamma)$. Unlike BSMA, SCTS and BMMM do not use DS capture. In BMMM, separate RTS transmissions are used to co-ordinate the CTS responses from the receivers, while in the SCTS protocol we propose, the co-ordination is achieved through sequence numbers. 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 (a) $\Delta_O(\Gamma)$, (b) broadcast based multiparty, (c) unicast based multiparty, (d) quorum-1 multiparty and (e) 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*. We have not compared the performance of $\Delta_O(\Gamma)$ with that of the MAC policies proposed for wireless LANs, e.g. [14], because they can only be used when a single node is in the transmission range of all other nodes in the network. This assumption does not hold in adhoc networks.

A. Simulation Scenario for SCTS Protocol

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 [30]. 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 sequential CTS protocol (Figure 4) to implement $\Delta_O(\Gamma)$ and quorum-1, full-quorum multiparty policies. Unicast senders use the standard IEEE 802.11 MAC [31]. We consider a time interval of 2000 seconds. To study the system in a steady state, we collect the relevant data only in the last 1500 seconds. Each channel has capacity $C = 11$ Mbps [31]. 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 length of a slot is 20 μs and the maximum propagation delay is 2 μs.

We consider carefully selected topologies that represent different interference extremes (Figure 8). In each topology, 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. We study the impact of high network load on the multiparty session by increasing λ_U (Figures 10 and 12). We have considered large randomly generated topologies in Section VI-A.

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

1) Discussion on Simulation Results: We have proved that any stable policy that selects any two consecutive quorums 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), i.e., $\frac{n_u(t)}{t}$ 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 (Figure 7(a)), and $\frac{n_u(t)}{t}$ converges (Figure 9). This validates the optimality result. Now, Theorem 2 guarantees that $\Delta_O(\Gamma)$ would select two consecutive quorums 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 quorums for low values of Γ further validates the theorem. Also, Figure 7(c) shows that as Γ increases the throughput of the multiparty session under $\Delta_O(\Gamma)$ converges to optimum value. We note that the optimality is achieved for small values of Γ.

Now, we compare the throughput of the multiparty session in topologies 1 and 2 under $\Delta_O(\Gamma)$ with that under broadcast based multiparty, quorum-1 multiparty, full-quorum multiparty and unicast based multiparty policies. We observe that Δ_O achieves much higher throughput than the other existing policies in both the topologies (Figures 10, 11 and 12). We next explain the performance difference.

The broadcast based multiparty 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 throughput. Quorum-1 policy exchanges hand-shake 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/sec in Topology 1). 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 hand-shake 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.

From Figures 10(b) and 12(b), we observe that the throughput gains of $\Delta_O(\Gamma)$ over the unicast based multiparty and full-quorum multiparty policies increase with increase in λ_M. This is because the last two policies' stability regions are small, and hence their throughputs saturate even for small values of λ_M (Figures 10(a) and 12(a)). On the other hand, $\Delta_O(\Gamma)$ has a 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 quorum 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 12(a)). Now, the throughput gains of $\Delta_O(\Gamma)$ over quorum-1 and broadcast based multiparty policies decrease with the increase in λ_M. This is because as λ_M increases the quorums chosen by $\Delta_O(\Gamma)$ decrease and become closer to 1. Thus these policies' transmission decisions become similar.

From Figures 10(e) and 12(e), we observe that as λ_U increases the throughput gains of $\Delta_O(\Gamma)$ over all the existing policies increase. 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 multiparty receivers are ready most of the time resulting in a large reward per transmission under all the policies. Thus, the choice of quorum does not affect the throughput. But for higher values of λ_U, large number of multiparty 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 quorum.

Figures 10(c) and 12(c) show that in both the topologies when the multiparty sender uses $\Delta_O(\Gamma)$, the throughput of unicast sessions is similar to or higher than that under any other policy for the multiparty sender. Thus, $\Delta_O(\Gamma)$ increases the throughput of the multiparty 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 quorum-1 policy and $\Delta_O(\Gamma)$ is small for all values of λ_M in topology 2 (Figure 12(a)). Recall that in topology 2, M and m_i are not ready when u_i is receiving a packet. Thus, when M is ready, all the multiparty receivers are also ready. Hence, the reward per packet should be 8 irrespective of the choice of quorum. Thus, $\Delta_O(\Gamma)$ and quorum-1 multiparty policy should achieve the same throughput. The throughput values differ slightly because of the collision of RTS messages from U_i and M at m_i, and also because of transmission of RTS by U_i before m_i could transmit a CTS. In these cases, m_i will not transmit the CTS, and it will not receive the data packet even if M transmits. Since $\Delta_O(\Gamma)$ chooses an appropriate quorum value, unlike quorum-1 multiparty policy which always has quorum 1, it may not transmit when m_i does not reply with the CTS, and hence 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 quorum-1 multiparty, e.g., the throughput gain is about 25% in Figure 10(b).

We now evaluate the control overhead of different policies. The overhead decreases the throughput as transmission of the control packets and the SIFS intervals increase 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 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 (Figure 11(c)) and energy consumption (Figure 11(d)) for different packet sizes.

Figure 11(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 ($\alpha\frac{P_M+O}{g\times P_M}$ Joules/byte). Without loss of generality, we assume $\alpha = 1$. Figure 11(d) shows that for moderate packet sizes (≥ 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 $\frac{1}{g}\left(1 + \frac{O}{P_M}\right)$ 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 multiparty is much higher than that under other policies as it transmits a packet separately to each receiver. Moreover, EPs under various schemes implemented using SCTS and broadcast based multiparty are similar for various values of λ_M and λ_U (Figure 11(e) and (f)).

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 $\Delta_O(\Gamma)$ (Section IV-C). 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.

B. Simulation Scenario for Busy Tone based Protocol

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 [30]. 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 V (Figure 5 and 6) to implement $\Delta_O(\Gamma)$ and quorum-1 multiparty.

We consider sample topologies shown in Figure 8. In both the topologies, we assume that the unicast sender U_i generates packets at rate λ_U and the multiparty 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 multiparty session is small when the packet size for unicast sessions is large. We, however, study the impact of high network load on the multiparty session by increasing λ_U (Figures 14 and 15).

1) Discussion on Simulation Results: We compare the throughput of the multiparty session in topologies 1 and 2 under $\Delta_O(\Gamma)$ with that under broadcast based multiparty, quorum-1 multiparty and unicast based multiparty policies. We observe that Δ_O achieves much higher throughput than the other existing policies in both the topologies (Figures 14 and 15). We next explain the performance difference.

The broadcast based multiparty 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. Quorum-1 policy exchanges hand-shake messages and avoids the 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 policy $\Delta_O(\Gamma)$ outperforms these policies as it prevents data packet collisions by exchanging hand-shake massages, and it also prevents transmission when only a few receivers are ready by choosing an appropriate quorum value. The unicast based multiparty policy uses separate transmissions to reach different members of the party 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 14(b) and 15(b), we observe that the throughput gain of $\Delta_O(\Gamma)$ over the unicast based multiparty 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 14(a) and 15(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 quorum 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 14(a) and 15(a)).

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

From Figures 14(e) and 15(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 multiparty receivers are ready most of the time resulting in a large reward per transmission under all the policies. Thus, the choice of quorum does not affect the throughput. But for higher values of λ_U, large number of multiparty 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 quorum. 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 13. Simulation results show that the throughput of multiparty 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 13).

Figures 14(c) and 15(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 multiparty 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 quorum-1 policy and $\Delta_O(\Gamma)$ is small for all values of λ_M in topology 2 (Figure 15(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 multiparty receivers are also ready. Hence, the reward per packet should be 8 irrespective of the choice of quorum. In other words, $\Delta_O(\Gamma)$ and quorum-1 multiparty 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 quorum value, it may not transmit when m_i does not put up a busy tone, and thus unlike quorum-1 multiparty policy which always has quorum 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 quorum-1 multiparty, e.g. Figure 14(b).

Note that since M is ready when all the receivers m_i's are ready, the energy per unit payload byte (EP) is approximately same under all the schemes implemented using SCTS.

C. Simulation Setup for Large Topologies

We assume that each node can communicate in the circle of radius $r = \sqrt{\frac{2\log(n)}{n}}$ units, where n is the number of nodes in the network. This choice of communication range is based on the connectivity results obtained in [32]. We construct a graph $G = (V, E)$, where V is a set of all the nodes and E is a set of edges formed such that $(i, j) \in E$ if the distance between the nodes i and j is less than r. We obtain a multiparty routing tree from a sender to its group members by constructing the shortest path from the sender to each receiver. Every hop in the tree corresponds to a MAC layer multiparty session (e.g., refer to Figure 3).

Unlike Sections VI-A and VI-B, here we use event driven simulator written in C instead of ns-2. For simplicity, we assume that time is slotted, and clocks at all the nodes are synchronized. RTS and CTS have equal sizes, and they can be transmitted in a single slot. Propagation delay is 0. The decision schemes are implemented using BCTS protocol.

We place 500 nodes uniformly in a square of unit area. The first 10 nodes are session sources. A node is a multiparty group member for a session with probability p_m. We choose $p_m = 0.1$. We observe that a node can be a member of multiple multiparty groups.

The arrival process at each of the sources is ON-OFF Markovian. A packet arrives in a slot only if the process is in ON state. Let α (β) denote the transition probability from OFF (ON) state to ON (OFF) state. We note that the expected arrival rate λ in the steady state is equal to the steady state probability that the arrival process is in ON state. Thus, $\lambda = \frac{\alpha}{\alpha+\beta}$. We consider bursty arrival process. We fix $\beta = 0.1$ and choose α to obtain different values of λ. The packet lengths are exponential (rounded to the closest integer) with mean 5. New packets enter the wireless network at the source nodes, and leave at nodes that are the members of the corresponding multiparty groups. The back-off intervals are uniformly distributed between 1 and 6. We run the simulation for 100,000 slots. To obtain the steady state performance, we collect the relevant data in the last 50,000 slots. We take average over 5 simulation runs.

We use the multiparty MAC proposed in Figure 4 for simulating the optimal strategy $\Delta_O(\Gamma)$. We refer to this implementation as quorum based multiparty. Observe that the policy is developed for a single session, while in the network case multiple sessions pass through a single node. Packets corresponding to all the multiparty sessions are stored in a single queue and the packet scheduling is FIFO over all the sessions. We compare the performance of the quorum based scheme with unicast based multiparty, broadcast based multiparty and broadcast based multiparty(1) proposed in [16]. In the last scheme, a sender transmits whenever at least one receiver is ready.

1) Discussion on the Simulation Results: In the simulations we measure the *network throughput*. The throughput for a receiver is the total number of packets it receives per unit time. The network throughput is the sum of the receiver throughputs. We also obtain the throughput gain of the quorum based multiparty scheme $\Delta_O(\Gamma)$ over other policies. The *throughput gain* of $\Delta_O(\Gamma)$ over a policy Δ is computed as $\frac{100\times(\Omega^{\Delta_O(\Gamma)}-\Omega^{\Delta})}{\Omega^{\Delta}}$.

Figure 16 compares the throughput performance of the schemes. We observe that the quorum based multiparty strategy obtains significantly higher throughput than the other schemes (Figure 16(b)). The performance difference with the broadcast based scheme can be explained as follows. The broadcast based scheme does not exchange any hand shake messages, and thus results in frequent data packet collisions. Furthermore, it transmits a data packet even when only a few receivers are ready as it does not know the readiness states of the receivers before the transmission decision. Thus the reward per data packet may be low resulting in low system throughput. The handshake mechanism in the quorum based multiparty scheme prevents any data packet collision, as also data transmission when only a few receivers are ready. For heavy network load, the total number of packets transmitted by a sender does not increase with increase in arrival rate, and hence the throughput saturates for all schemes. A higher packet arrival rate causes senders' queues to blow up. In this case, the quorum based multiparty strategy sets the minimum quorum.

We now explain the performance difference between the quorum and the unicast based multiparty schemes. Unicast based multiparty uses separate transmissions to reach different members of the multiparty group even when they can be reached using one transmission. This increases channel contention for neighboring sources, and reduces the stability region for this policy. As Figure 16(a) shows, the throughput of the unicast based multiparty saturates at small values of arrival rates. The unicast based multiparty scheme delivers every packet to every receiver by virtue of separate transmission for each receiver. Thus it does not suffer from any packet loss, but this zero-loss performance is attained at the cost of significant throughput reduction. The quorum based multiparty scheme has significantly higher throughput.

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 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) maximizing performance in full-coordination and passive observation cases, (b) studying the multiparty MAC's interaction with higher layer protocols for utilities like multicast, anycast, transmitting routing updates, attaining reliability through multipath diversity etc., and (c) maximizing the performance in presence of mobility, dynamic group membership changes, security concerns, etc.

References

[1] S. Marti, T. Giuli, K. Lai, and M. Baker, "Mitigating routing misbehaviour in mobile ad hoc networks," in *ACM Mobicom*, 2000.

[2] S. Mueller and D. Ghosal, "Multipath routing in mobile ad hoc networks: Issues and challenges," in *Lecture Notes in Computer Science (Invited paper)*, M. C. Calzarossa and E. Gelenbe, Eds., 2004.

[3] D. Ganesan, R. Govindan, S. Shenker, and D. Estrin, "Highly resilient, energy efficient multipath routing in wireless sensor networks," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 5, no. 4, 2001.

[4] Z. J. Haas, J. Y. Halpern, and L. Li, "Gossip-based ad hoc routing," in *INFOCOM 2002*, New York, NY, Jun 2002.

[5] P. Samar and Z. J. Haas, "Strategies for broadcasting updates by proactive routing protocols in mobile adhoc networks," in *MILCOM 2002*, Anaheim, CA, Oct 2002.

[6] C. Intanagonwiwat, R. Govindan, and D. Estrin, "Directed diffusion: A scalable and robust communication paradigm for sensor networks," in *MobiCOM 2000*, Boston, Massachusetts, 2000.

[7] D. Braginsky and D. Estrin, "Rumor routing algorithm for sensor networks," in *First Workshop on Sensor Networks and Applications (WSNA)*, 2002.

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

[9] T. Gopalsamy, M. Singhal, D. Panda, and P. Sadayappan, "A reliable multicast algorithm for mobile ad hoc networks," in *ICDCS'02*, 2002.

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

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

[12] 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.

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

[14] J. Kuri and S. Kasera, "Reliable multicast in multi-access wireless lans," in *INFOCOM'99*, 1999.

[15] K. Tang and M. Gerla, "Mac reliable broadcast in ad hoc networks," in *Proceedings of IEEE MILCOM'01*, 2001.

[16] ——, "Mac layer broadcast support in 802.11 wireless networks," in *Proceedings of IEEE MILCOM 2000*, 2000.

[17] ——, "Random access mac for efficient broadcast support in ad hoc networks," in *Proceedings of IEEE WCNC 2000*, 2000.

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

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

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

[21] 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*, vol. 37, no. 12, pp. 1936–1949, Dec 1992.

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

[23] T. Nandagopal, T. Kim, X. Gao, and V. Bharghavan, "Achieving mac layer fairness in wireless packet networks," *Proceedings of ACM Mobicom 2000*, 2000.

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

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

[26] 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*, vol. 23, no. 12, 1975.

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

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

[29] "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 extentions in the 5ghz band in europe," March 2003.

[30] 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.

[31] *IEEE Standerd 802.11b - Wireless LAN Medium Access Control(MAC) and Physical Layer(PHY) specifications: High Speed Physical Layer (PHY) in the 2.4 GHz Band*, IEEE Standard, 1999.

[32] P. Gupta and P. R. Kumar, "Critical power for asymptotic connectivity in wireless networks," in *Stochastic Analysis, Control, Optimization and Applications: A Volume in Honor of W.H. Fleming*, 1998.

[33] P. Chaporkar, S. Sarkar, A. Bhat, and R. R. Shetty, "Dynamic strategies for maximizing throughput in multiparty mac," University of Pennsylvania, http://www.seas.upenn.edu/∼prasanna/techrep, Tech. Rep., Apr 2004.

[34] P. Chaporkar and S. Sarkar, "Wireless multicast: Theory and approaches," *IEEE Transactions on Information Theory*, 2005, to appear in Jun 2005.

APPENDIX

First we present some definitions.

Definition 11: 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 12: 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 n^{th} time slot. Since, the readiness and the arrival processes are Markovian, the process $\{Y_n^{\Delta} : n \geq 1\}$ is a 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.

Proof of Theorem 1

Intuition: A stable policy needs to maximize the expected reward per packet in order to maximize its throughput. Hence, Δ_1 achieves a higher throughput than the throughput achieved by Δ only if it achieves a higher reward infinitely often. Since Δ chooses quorum T or $T+1$ in most of the slots, it sends packets in all such slots when $T+1$ or more receivers are ready. Further, the remaining packets achieve reward T. Hence, Δ_1 can achieve a higher reward only in the slots in which Δ does not have a packet or the quorum is neither T nor $T+1$. Since, such slots are few, the result follows.

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, $\lim_{t\to\infty} \frac{z^{\Delta_1}(t)}{t} = \lambda$ w.p. 1, where $z^{\Delta_1}(t)$ denote the number of packets transmitted under policy Δ_1 till 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 till time t in which the quorum under Δ is T or $T+1$ and let $\bar{S}^{\Delta}(t)$ denote the remaining slots till 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 till 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 till 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 $\widehat{z}^{\Delta}(t)$ denote the number of packets that departed till time t when the quorum under Δ is T or $T+1$.

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

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 following relation.

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

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

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

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

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

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

We observe that the policy that achieves the upper bound for the system throughput achieves maximum reward by exploiting all the samples with $T+1$ or more ready receivers and the remaining packets are transmitted for the reward T. The upper bound in (16) may not be tight depending on the choice of T and the policy.

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

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 $\Gamma \geq \Gamma_2(\epsilon, T)$

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

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,

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

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 [33] for the formal proofs.

A. *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

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

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

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,

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

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 till 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). ■

B. *Proof for Lemma 2*

Intuition: Since Δ_T 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.

Proof: Let $\mathcal{B}$ denote a random variable indicating the length of a busy period under Δ_T. We show that $\mathbf{E}[\mathcal{B}] < \infty$. Thus, stability of Δ_T follows. Let $A(t)$ denote the number of arrivals till time t. Without loss of generality, we assume that the busy period under consideration starts in slot 1, i.e., $A(1) > 0$. Now, $A_1 \leq \gamma < \infty$. Let $z^{\Delta_T}(t)$ denote the departures in the system till time t.

We first consider a fictitious system in which the sender's queue is never empty. Let $D^{\Delta_T}(t)$ denote the number of departures till time t under Δ_T. We assume that the actual and the fictitious systems have the same readiness states in every slot. If the busy period length is greater than t on a sample path ω, i.e., $\mathcal{B}(\omega) \geq t$, then

$$z^{\Delta_T}(\omega, t) = D^{\Delta_T}(\omega, t). \tag{30}$$

Now, we obtain some properties of $D^{\Delta_T}(t)$ and then use (30) to show the required.

In the fictitious system, the number of departures $D^{\Delta_T}(t)$ is equal to the number of slots with ready sender and T or more ready receivers. Hence, from the ergodicity of the arrival and the readiness processes,

$$\begin{aligned} \lim_{t\to\infty} \frac{A(t)}{t} &= \lambda \qquad \text{w.p. } 1 \qquad \text{and} && (31) \\ \lim_{t\to\infty} \frac{D^{\Delta_T}(t)}{t} &= \sum_{u=T}^{G} b_u^R \qquad \text{w.p. } 1. && (32) \end{aligned}$$

We observe that if condition (17) is satisfied then there exists a $\delta > 0$ such that

$$\lim_{t\to\infty} \frac{D^{\Delta_T}(t)}{t} = \lambda + \delta \qquad \text{w.p. } 1. \tag{33}$$

Now, using the properties derived above for the fictitious system, we show that the expected length of a busy period is bounded in the actual system. We consider an event where the busy period under consideration is larger than t.

$$\begin{aligned} \{\omega : \mathcal{B}(\omega) \geq t\} &= \left\{ \bigcap_{\widehat{t}=1}^{t} \{\omega : A(\omega, \widehat{t}) - z^{\Delta_T}(\omega, \widehat{t}) > 0\} \right\} \\ &= \left\{ \bigcap_{\widehat{t}=1}^{t} \{\omega : A(\omega, \widehat{t}) - D^{\Delta_T}(\omega, \widehat{t}) > 0\} \right\}. && (34) \end{aligned}$$

The last equality follows from (30).

$$\begin{aligned} &\mathbf{P}\{\mathcal{B} \geq t\} \\ &\quad= \mathbf{P}\left\{ \bigcap_{\widehat{t}=1}^{t} \{A(\widehat{t}) - z^{\Delta_T}(\widehat{t}) > 0\} \right\} \\ &\quad= \mathbf{P}\left\{ \bigcap_{\widehat{t}=1}^{t} \{A(\widehat{t}) - D^{\Delta_T}(\widehat{t}) > 0\} \right\} \qquad \text{(from (34))} \\ &\quad\leq \mathbf{P}\left\{ A(t) - D^{\Delta_T}(t) > 0 \right\} \\ &\quad= \mathbf{P}\left\{ \frac{A(t)}{t} - \frac{D^{\Delta_T}(t)}{t} > 0 \right\} \\ &\quad= 1 - \mathbf{P}\left\{ \frac{A(t)}{t} - \frac{D^{\Delta_T}(t)}{t} \leq 0 \right\} \\ &\quad\leq 1 - \mathbf{P}\left\{ \{\frac{A(t)}{t} \leq \lambda + \frac{\delta}{2}\} \cap \{\frac{D^{\Delta_T}(t)}{t} \geq \lambda + \frac{\delta}{2}\} \right\} \\ &\quad= \mathbf{P}\left\{ \{\frac{A(t)}{t} > \lambda + \frac{\delta}{2}\} \cup \{\frac{D^{\Delta_T}(t)}{t} < \lambda + \frac{\delta}{2}\} \right\} \\ &\quad\leq \mathbf{P}\left\{ \frac{A(t)}{t} > \lambda + \frac{\delta}{2} \right\} + \mathbf{P}\left\{ \frac{D^{\Delta_T}(t)}{t} < \lambda + \frac{\delta}{2} \right\}. && (35) \end{aligned}$$

Now, $\frac{A(1)}{t} \to 0$ as $t \to \infty$. Further, from (31), (33) and exponential convergence of empirical distribution to the unique stationary distribution for finite, ergodic MC's [25] we conclude that

$$\sum_{t=1}^{\infty} \mathbf{P}\left\{\frac{A(t)}{t} > \lambda + \frac{\delta}{2}\right\} < \infty. \tag{36}$$

$$\sum_{t=1}^{\infty} \mathbf{P}\left\{\frac{D^{\Delta_T}(t)}{t} < \lambda + \frac{\delta}{2}\right\} < \infty. \tag{37}$$

$$\mathbf{E}[\mathcal{B}] = \sum_{t=1}^{\infty} \mathbf{P}\left\{\mathcal{B} \geq t\right\} < \infty \quad \text{(from (35), (36) and (37))}.$$

This proves the Lemma. ■

C. *Proof for Lemma 3*

Proof: From Lemma 2, we know that if T satisfies (17), then there exist 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. ■

D. *Proof for Lemma 4*

Intuition: If quorum T satisfies (19), then the rate at which slots with ready sender and T or more ready receivers arrive is less than the packet arrival rate. Hence, the drift under $\Delta_{T,B}$ is positive, and the queue length increases. Thus, if the value of Γ increases, then the probability that the queue length is less than or equal to $B - \Gamma$ decreases.

Proof: We prove this Lemma in three steps. (a) 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. (b) 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. (c) 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 $\widehat{B} = \frac{B}{\gamma}$.

Let $A(t)$ denote the arrivals in the system till 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

$$\lim_{t\to\infty} \frac{A(t)}{t} = \lambda \;\; \text{w.p. 1}. \tag{38}$$

Furthermore, let $z^{\Delta_{B,T}}(t)$ denote the number of departures till 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 till 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

$$\lim_{t\to\infty} \frac{D^{\Delta_{B,T}}(t)}{t} = \sum_{u=T}^{G} b_u^R = \lambda - \delta \;\; \text{w.p. 1}. \tag{39}$$

$$\text{Now for every } k \le \widehat{B}, \quad \mathbf{P}\{\widehat{Z} \ge k\} = 1, \tag{40}$$

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 $k \ge 1$

$$\mathbf{P}\{\widehat{Z} \ge \widehat{B} + k\} = \mathbf{P}\left\{\bigcap_{u=0}^{k}\{\widehat{A}(\widehat{B}+u) - z^{\Delta_{B,T}}(\widehat{B}+u) < B\}\right\}. \tag{41}$$

Further, for every $u \le k$

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

Hence, from (41) it follows that

$$\begin{aligned}\mathbf{P}\{\widehat{Z} \ge \widehat{B} + k\} &\le \mathbf{P}\left\{\bigcap_{u=1}^{k}\{A(\widehat{B}+u) - z^{\Delta_{B,T}}(\widehat{B}+u) < B\}\right\}\\ &\le \mathbf{P}\left\{A(\widehat{B}+k) - z^{\Delta_{B,T}}(\widehat{B}+k) < B\right\}.\end{aligned} \tag{42}$$

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 (42) becomes

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

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

$$\text{for every} \quad k > \left(\frac{4}{\delta} - 1\right)B, \quad \frac{B}{t} < \frac{\delta}{4}. \tag{43}$$

Hence, for every k that satisfies (43),

$$\begin{aligned}&\mathbf{P}\{\widehat{Z} \ge t\}\\ &\quad\le \mathbf{P}\left\{\frac{A(t)}{t} - \lambda < \frac{-\delta}{4}\right\} + \mathbf{P}\left\{\frac{D^{\Delta_{B,T}}(t)}{t} - (\lambda-\delta) \ge \frac{\delta}{2}\right\}\\ &\quad\le 2e^{-(\widehat{B}+k)C},\end{aligned} \tag{44}$$

where C is a constant independent of the initial arrival and receiver readiness state. The upper bound (44) follows from (38), (39), substituting $t = \widehat{B} + k$, and the large deviation bounds for the finite, ergodic Markov chains [25].

Let $\widehat{k} = \left(\frac{4}{\delta} - 1\right) B$. Now,

$$\begin{aligned}
&\mathbf{E}[\widehat{Z}] \\
&= \sum_{k=1}^{\infty} \mathbf{P}\{\widehat{Z} \geq k\} \\
&= \sum_{k=1}^{\widehat{B}} \mathbf{P}\{\widehat{Z} \geq k\} + \sum_{k=1}^{\widehat{k}} \mathbf{P}\{\widehat{Z} \geq \widehat{B} + k\} \\
&\quad + \sum_{k=\widehat{k}+1}^{\infty} \mathbf{P}\{\widehat{Z} \geq \widehat{B} + k\} \\
&\leq \frac{B}{\gamma} + \frac{4B}{\delta} + 2\sum_{k=1}^{\infty} e^{-(\widehat{B}+k)C} \\
&\quad \text{(since } \mathbf{P}(E) \leq 1 \text{ for every event } E\text{, and (44))} \\
&= B\left(\frac{1}{\gamma} + \frac{4}{\delta}\right) + \frac{2\gamma e^{-\widehat{B}C}}{C}. \qquad (45)
\end{aligned}$$

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 $k \leq \Gamma$

$$\mathbf{P}\{Z \geq k\} = 1. \qquad (46)$$

$$\begin{aligned}
&\text{For } k \geq 1, \quad \mathbf{P}\left\{Z \geq \Gamma + k\right\} \\
&= \mathbf{P}\left\{\bigcap_{u=0}^{k} \{z^{\Delta_{B,T}}(\Gamma+u) - \widehat{A}(\Gamma+u) < \Gamma\}\right\} \\
&= 1 - \mathbf{P}\left\{\bigcup_{u=0}^{k} \{z^{\Delta_{B,T}}(\Gamma+u) - \widehat{A}(\Gamma+u) \geq \Gamma\}\right\} \\
&\geq 1 - \min\left\{1, \sum_{u=0}^{k} \mathbf{P}\{z^{\Delta_{B,T}}(\Gamma+u) - \widehat{A}(\Gamma+u) \geq \Gamma\}\right\}. \qquad (47)
\end{aligned}$$

The sender's buffer is never empty till 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,

$$\begin{aligned}
&\mathbf{P}\left\{z^{\Delta_{B,T}}(\Gamma+u) - \widehat{A}(\Gamma+u) \geq \Gamma\right\} \\
&= \mathbf{P}\left\{\bigcup_{v=0}^{u} \left\{\{D^{\Delta_{B,T}}(\Gamma+u) - \widehat{A}(\Gamma+u) \geq \Gamma\} \bigcap \right.\right. \\
&\quad \left.\left.\{Q(v) = B, Q(m) < B : \forall\, m \in \{v+1, \Gamma+u\}\right\}\right\}, \qquad (48) \\
&\leq \sum_{v=0}^{u} \mathbf{P}\left\{\{D^{\Delta_{B,T}}(\Gamma+u) - \widehat{A}(\Gamma+u) \geq \Gamma\} \bigcap \right. \\
&\quad \left.\{Q(v) = B, Q(m) < B : \forall\, m \in \{v+1, \Gamma+u\}\right\}. \qquad (49)
\end{aligned}$$

Relation (48) 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, (49) follows from union bound property. Furthermore, if $Q(v) = B$ and $Q(m) < B, \forall\, m \in \{v+1, \Gamma+u\}$, then

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

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

$$\begin{aligned} & \mathbf{P}\{A(\Gamma+u) - A(v) = k\} \\ = \;& \mathbf{P}\{A(\Gamma+u-v) = k\}, \text{and} \qquad (50) \\ & \mathbf{P}\{D^{\Delta_{B,T}}(\Gamma+u) - D^{\Delta_{B,T}}(v) = k\} \\ = \;& \mathbf{P}\{D^{\Delta_{B,T}}(\Gamma+u-v) = k\}. \qquad (51) \end{aligned}$$

$$\begin{aligned} & \mathbf{P}\left\{z^{\Delta_{B,T}}(\Gamma+u) - \widehat{A}(\Gamma+u) \geq \Gamma\right\} \\ \leq \;& \sum_{v=1}^{u} \mathbf{P}\left\{D^{\Delta_{B,T}}(\Gamma+u-v) - A(\Gamma+u-v) \geq \Gamma\right\} \\ & \text{(from (50) and (51))} \qquad (52) \\ = \;& \sum_{v=0}^{u} \mathbf{P}\left\{D^{\Delta_{B,T}}(\Gamma+v) - A(\Gamma+v) \geq \Gamma\right\}. \qquad (53) \end{aligned}$$

Let $t = \Gamma + v$ and consider

$$\begin{aligned} & \mathbf{P}\{D^{\Delta_{B,T}}(t) - A(t) \geq \Gamma\} \\ & = \mathbf{P}\left\{\frac{D^{\Delta_{B,T}}(t)}{t} - \frac{A(t)}{t} \geq \frac{\Gamma}{t}\right\} \\ & = \mathbf{P}\left\{\left(\frac{D^{\Delta_{B,T}}(t)}{t} - (\lambda-\delta)\right) - \left(\frac{A(t)}{t} - \lambda\right) \geq \delta + \frac{\Gamma}{t}\right\} \\ & \leq \mathbf{P}\left\{\frac{D^{\Delta_{B,T}}(t)}{t} - (\lambda-\delta) \geq \frac{\delta}{2}\right\} + \mathbf{P}\left\{\frac{A(t)}{t} - \lambda \leq \frac{-\delta}{2}\right\}. \qquad (54) \end{aligned}$$

From (38), (39), and large deviation bound for finite ergodic MC's [25] we obtain

$$\mathbf{P}\left\{\left(\frac{A(t)}{t} - \lambda\right) \leq \frac{-\delta}{2}\right\} \leq e^{-tC_1} \qquad \text{and} \qquad (55)$$

$$\mathbf{P}\left\{\left(\frac{D^{\Delta_{B,T}}(t)}{t} - (\lambda-\delta)\right) > \frac{\delta}{2}\right\} \leq e^{-tC_2}, \qquad (56)$$

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 (54), (55) and (56)

$$\mathbf{P}\{D^{\Delta_{B,T}}(t) - A(t) > \Gamma\} \leq 2e^{-t\widehat{C}}. \qquad (57)$$

$$\begin{aligned}
&\mathbf{P}\left\{z^{\Delta_{B,T}}(\Gamma+u)-\widehat{A}(\Gamma+u)>\Gamma\right\} \\
&\le \sum_{v=0}^{u} 2e^{-t\widehat{C}} \quad \text{(from (53) and (57))} \\
&= \sum_{v=0}^{u} 2e^{-(\Gamma+v)\widehat{C}} \quad \text{(Since } t=\Gamma+v\text{)} \\
&\le \frac{2e^{-\Gamma\widehat{C}}}{\widehat{C}}.
\end{aligned} \tag{58}$$

Using (58) in (47), we obtain

$$\mathbf{P}\left\{Z \ge \Gamma + k\right\} \ge 1 - \min\left\{1, \frac{2(k+1)e^{-\Gamma\widehat{C}}}{\widehat{C}}\right\}. \tag{59}$$

$$\begin{aligned}
\min\left\{1, \frac{2(k+1)e^{-\Gamma\widehat{C}}}{\widehat{C}}\right\} &= \frac{2ke^{-\Gamma\widehat{C}}}{\widehat{C}} \quad \forall\, k < \frac{\widehat{C}}{2}e^{\Gamma\widehat{C}}-1 \\
&= 1 \quad \forall\, k \ge \frac{\widehat{C}}{2}e^{\Gamma\widehat{C}}-1.
\end{aligned} \tag{60}$$

Thus,

$$\begin{aligned}
\mathbf{E}[Z] &\ge \Gamma + \sum_{k=1}^{\infty}\left(1-\min\left\{1, \frac{2ke^{-\Gamma\widehat{C}}}{\widehat{C}}\right\}\right) \quad \text{(from (46))} \\
&\ge \Gamma + \sum_{k=1}^{\left\lfloor \frac{\widehat{C}}{2}e^{\Gamma\widehat{C}}\right\rfloor}\left(1-\frac{2ke^{-\Gamma\widehat{C}}}{\widehat{C}}\right) \quad \text{(from (59) and (60))} \\
&\ge \Gamma + \frac{\widehat{C}}{4}e^{\Gamma\widehat{C}} - \frac{1}{2}.
\end{aligned} \tag{61}$$

Part (c): Since the system has finite buffer, $\{\widehat{Y}_n^{\Delta_B} : n \ge 1\}$ 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 till time t, and $\widehat{m}(t)$ denote the number of cycles till 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 u^{th} 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 $\le B-\Gamma$, respectively. By ergodicity

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

$$\text{Since } t \ge \textstyle\sum_{u=1}^{m(t)} X_u, \quad \displaystyle\sum_{k=0}^{B-\Gamma}\pi_k \le \lim_{t\to\infty}\frac{\sum_{u=1}^{m(t)} X_{u-}}{\sum_{u=1}^{m(t)} X_u} \qquad \text{w.p.1.}$$

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

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

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

$$\sum_{k=0}^{B-\Gamma} \pi_k \leq \frac{\mathbf{E}[\widehat{Z}]}{\mathbf{E}[Z]} \qquad \text{w.p. } 1. \tag{62}$$

$$\text{From (45), (61) and (62)} \quad \sum_{k=0}^{B-\Gamma} \widehat{\pi}_k \leq \frac{B\left(\frac{1}{\gamma} + \frac{4}{\delta}\right) + \frac{2\gamma e^{-\widehat{B}C}}{C}}{\Gamma + + \frac{\widehat{C}}{4} e^{\Gamma \widehat{C}} - \frac{1}{2}}.$$

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

This proves the required result. ■

E. Proof for Lemma 5

Proof: Let P_k denote the stationary probability that the sender's queue length is k under policy $\Delta_O(\Gamma)$. Then,

$$P_{T_-} = \sum_{k=(G-T+1)\Gamma+1}^{\infty} P_k.$$

Let $Q(\omega, t)$ and $Q_T(\omega, t)$ be the sender's queue length under policies $\Delta_O(\Gamma)$ and Δ_T at the beginning of slot t, respectively. We note that

$$\begin{aligned} Q(\omega, t+1) &= \max\{0, Q(\omega, t) + A(\omega, t) - d^{\Delta_O(\Gamma)}(\omega, t)\} \\ Q_T(\omega, t+1) &= \max\{0, Q_T(\omega, t) + A(\omega, t) - d^{\Delta_T}(\omega, t)\}, \end{aligned}$$

where $A(t)$ and $d^{\Delta}(t)$ denote the number of arrivals and departures in slot t, respectively. Further, $d^{\Delta}(t) \in \{0, 1\}$.

It suffices to show that on every sample path ω and every time slot t, $Q(\omega, t) \geq (G - T + 1)\Gamma + 1$ implies that $Q_T(\omega, t) \geq \Gamma + 1$. To prove this, we prove that for every t and ω

$$Q(\omega, t) - (G - T)\Gamma \leq Q_T(\omega, t). \tag{63}$$

We prove (63) by contradiction. Let (63) not be true on some sample path, say $\widehat{\omega}$. Since $Q(\widehat{\omega}, 0) = Q_T(\widehat{\omega}, 0) = 0$; (63) is true for $t = 0$. Hence, there exists a time $\widehat{t}$ such that

$$Q(\widehat{\omega}, \widehat{t}) - (G - T)\Gamma > Q_T(\widehat{\omega}, \widehat{t}) \tag{64}$$

and (63) is satisfied for all $t \leq \widehat{t} - 1$. In other words, $\widehat{t}$ is the first time slot in which condition (63) is violated.

To summarize, we know the following facts. (a) Given a sample path, arrivals are the same in both the systems. (b) Given a sample path, both the systems have the same readiness states. (c) At most one departure can occur in a given time slot. (d) (63) holds for every $t \leq \widehat{t} - 1$.

Therefore, we conclude that for (64) to be true the following conditions must hold.

$$Q(\widehat{\omega},\widehat{t}-1)-(G-T)\Gamma = Q_T(\widehat{\omega},\widehat{t}-1),$$

and in slot $\widehat{t}-1$ a packet departs under Δ_T but not under $\Delta_O(\Gamma)$. Since a departure occurs under Δ_T in slot $\widehat{t}-1$, $Q_T(\omega,\widehat{t}-1) > 0$. Thus,

$$\begin{aligned} Q(\widehat{\omega},\widehat{t}-1)-(G-T)\Gamma &> 0 \\ \Rightarrow \qquad Q(\widehat{\omega},\widehat{t}-1) &> (G-T)\Gamma. \end{aligned} \tag{65}$$

Thus, from (65), $\Delta_O(\Gamma)$ will choose quorum less than or equal to T in slot $\widehat{t}-1$. But then a packet must depart in slot $\widehat{t}-1$, under $\Delta_O(\Gamma)$ as well. This is a contradiction. Hence, the result follows. ■

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 $\Delta_{B,T}$. The arguments used to show the required are similar to those used in Lemma 5. Refer to [33] for the formal proof.

F. *Proof for Lemma 6*

Proof: Let P_k denote the stationary probability that the sender's queue length is k under policy $\Delta_O(\Gamma)$. Then,

$$P_{T_+} = \sum_{k=0}^{(G-T)\Gamma} P_k.$$

It suffices to show that on every sample path ω and every time slot t, $Q(\omega,t) \le (G-T)\Gamma$ implies that $Q_{T,B}(\omega,t) \le (G-T)\Gamma$, where $Q(\omega,t)$ and $Q_{T,B}(\omega,t)$ are the sender's queue length under policies $\Delta_O(\Gamma)$ and $\Delta_{B,T}$ at the beginning of slot t, respectively. To prove this, we prove that for every t and ω

$$Q(\omega,t) \ge Q_{T,B}(\omega,t). \tag{66}$$

We prove (66) by contradiction. Let (66) not be true on some sample path, say $\widehat{\omega}$. Since $Q(\widehat{\omega},0) = Q_{T,B}(\widehat{\omega},0) = 0$; (66) is true for $t=0$. Hence, there exists a time $\widehat{t}$ such that

$$Q(\widehat{\omega},\widehat{t}) < Q_{T,B}(\widehat{\omega},\widehat{t}) \tag{67}$$

and (66) is satisfied for all $t \le \widehat{t}-1$. In other words, $\widehat{t}$ is the first time slot in which condition (66) is violated.

To summarize, we know the following facts. (a) Given a sample path, arrivals are the same in both the systems. (b) Given a sample path, both the systems have the same readiness states. (c) At most one departure can occur in a slot. (d) (66) holds for every $t \le \widehat{t}-1$.

Thus, we conclude that for (67) to be true, the following conditions must hold.

$$Q(\widehat{\omega},\widehat{t}-1)\Gamma = Q_{T,B}(\widehat{\omega},\widehat{t}-1),$$

and in slot $\widehat{t}-1$ a packet departs under policy $\Delta_O(\Gamma)$ but not under policy $\Delta_{B,T}$. Since the sender's buffer has size B, $Q_{T,B}(\omega,t) \leq B$ for every t and every ω. Hence,

$$\begin{aligned} Q(\widehat{\omega},\widehat{t}-1) &\leq B \\ Q(\widehat{\omega},\widehat{t}-1) &\leq (G-T+1)\Gamma-1. \end{aligned} \tag{68}$$

Thus, from (68), $\Delta_O(\Gamma)$ will choose quorum greater than or equal to T in time slot $\widehat{t}-1$. Therefore, in slot $\widehat{t}-1$, a packet must also depart under $\Delta_{B,T}$. This is a contradiction. Hence, the result follows. ■

Proof for Theorem 3

Proof: Consider arbitrary stable policy Δ. Let the expected reward per packet in steady state under $\Delta_O(\Gamma)$ and Δ be $R^{\Delta_O(\Gamma)}$ and R^{Δ} respectively. Then, the system loss under these policies is

$$\begin{aligned} L^{\Delta_O(\Gamma)} &= G - R^{\Delta_O(\Gamma)} \quad \text{and} \\ L^{\Delta} &= G - R^{\Delta}. \end{aligned}$$

$$\begin{aligned} L^{\Delta_O(\Gamma)} - L^{\Delta} &= R^{\Delta} - R^{\Delta_O(\Gamma)} \\ &= \frac{1}{\lambda}(\Omega^{\Delta} - \Omega^{\Delta_O(\Gamma)}) \ \ \text{(Since } \Omega^{\Delta} = \lambda R^{\Delta}\text{)} \\ &\leq \epsilon \quad \text{(Since } \Delta_O(\Gamma) \text{ is } (\lambda\epsilon)\text{-optimal).} \end{aligned}$$

This proves the required. ■

Proof for Theorem 4

First, we show that Theorem 1 can be generalized to the case when the readiness states are probabilistic. Let $s_T^{\Delta}(t)$ denote the number of busy slots in which quorum R_T is chosen till time t under a generalized quorum policy Δ. Recall that it is not necessary to select a quorum when queue length is zero, as a packet cannot be transmitted in this case. The generalization is as follows.

Lemma 7 (Generalization of Theorem 1): For any $\epsilon > 0$, a stable generalized quorum policy Δ is ϵ-throughput optimal w.p. 1 if the following condition holds for some $T \in \{0,\ldots,m\}$.

$$\lim_{t\to\infty} \frac{s_T^{\Delta}(t) + s_{T+1}^{\Delta}(t)}{t} \geq 1 - \frac{\epsilon}{2m} \quad \text{w.p. 1.} \tag{69}$$

Proof: We observe that if $\Delta_O'(\Gamma)$ transmits only finite number of packets when the readiness vector is $[j_{l,1} \ \ldots \ j_{l,G}]$, then the reward achieved in these transmissions does not contribute in the throughput of the policy. This is because the throughput is the expected reward per unit time as $t \to \infty$. Thus, it suffices to consider readiness vectors in which the sender transmits infinitely often. But, if the sender transmits infinitely often in readiness state $[j_{l,1} \ \ldots \ j_{l,G}]$, then by the strong law of large numbers the expected reward per packet achieved in these transmissions is $\sum_{i=1}^{G} j_{l,i}$ w.p. 1.

Now, Lemma 7 follows from similar arguments as that in Theorem 1. Only difference in that now G has to be replaced by m, as there are m distinct values of quorums are possible and not G. ■

Now, we sketch the proof of Theorem 4.

Proof: Note that in view of Lemma 7, it suffices to show that $\Delta'_O(\Gamma)$ satisfies (69). The proof for this follows using exactly the same arguments as that in the proof of Theorem 2. Again, the main difference is that G and the steady state distribution b_u^R, $u \in \{0, \ldots, G\}$, in the proof of Theorem 2 has to be replaced by m and the steady state distribution $\widehat{b}_u^R$, $u \in \{0, \ldots, m\}$, respectively. ■

PROOF OF THEOREM 5

Proof: We have shown that a single quorum policy(T) Δ_T is stable if $\lambda < \frac{\sum_{u=T}^{G} b_u^R}{\mathbf{E}[V]}$, where $\mathbf{E}[V]$ is the expected packet transmission duration ([34], page 10, Lemma 1). In other words, the expected queue length under Δ_T is bounded if $\lambda < \frac{\sum_{u=T}^{G} b_u^R}{\mathbf{E}[V]}$. Using similar arguments as that in the proof of Lemma 5, it can be shown that on every sample path ω, the queue length under $\Delta_O(\Gamma)$ at time t ($Q(\omega, t)$) is less than or equal to the queue length Δ_T at time t ($Q_T(\omega, t)$) plus $(G - T)\Gamma$, i.e.,

$$Q(\omega, t) \leq Q_T(\omega, t) + (G - T)\Gamma,$$

for every ω and t. Since the expected queue length under Δ_T is bounded if $\lambda < \frac{\sum_{u=T}^{G} b_u^R}{\mathbf{E}[V]}$, it follows that the expected queue length under $\Delta_O(\Gamma)$ is also bounded if $\lambda < \frac{\sum_{u=T}^{G} b_u^R}{\mathbf{E}[V]}$ for any $T \in \{0, \ldots, G\}$. Alternatively, $\Delta_O(\Gamma)$ is stable if $\lambda < \frac{\sum_{u=T}^{G} b_u^R}{\mathbf{E}[V]}$ for any $T \in \{0, \ldots, G\}$. Thus, $\Delta_O(\Gamma)$ is stable if $\lambda < \frac{\sum_{u=0}^{G} b_u^R}{\mathbf{E}[V]}$.

Now, to show that $\Delta_O(\Gamma)$ is ϵ-throughput optimal even when packet transmission durations are iid, we observe the following. The fraction of slots in which any stable policy transmits is equal to $\lambda\mathbf{E}[V]$ w.p. 1. Recall that the readiness states do not change in the duration of packet transmissions. Hence, we can restrict our attention only to the slots in which a policy is not transmitting. Now, since the readiness states do not change in the duration of transmission, statistics of the readiness process observed by any stable policy in the slots in which it is not transmitting is the same. Thus, the throughput optimality of $\Delta_O(\Gamma)$ follows using the similar arguments in the proof of Theorem 2 except here we need to restrict our attention only to the slots in which a policy is not transmitting. Note that even though $\Delta_O(\Gamma)$ is ϵ-throughput optimal when the packet transmission durations are iid, its stability region is scaled down by the factor of $\mathbf{E}[V]$. ■

PROOF OF THEOREM 7

Proof: Consider two multiparty receivers with sequence numbers n and m. Without loss of generality, let RTS transmission be over at time 0 and let $n < m$. Let τ_n and τ_m denote the propagation delay from the sender to receivers n and m, respectively. Note that the RTS will arrive at the receivers n and m at τ_n and τ_m respectively. Now, the receiver n will start transmitting CTS at $\tau_n + nc + (n+1)s$ (Figure 4). Similarly, the receiver m will start transmitting CTS at $\tau_m + mc + (m+1)s$. The CTS transmissions from n and m will be over at $\tau_n + (n+1)c + (n+1)s$ and $\tau_m + (m+1)c + (m+1)s$, respectively.

Let $[C_u^a, C_u^f]$ denote the interval in which CTS from the receiver u arrives at the sender. Thus,

$$\begin{aligned}
C_n^a &= 2\tau_n + \tau_n + nc + (n+1)s, \\
C_n^f &= 2\tau_n + (n+1)c + (n+1)s, \\
C_m^a &= 2\tau_m + \tau_m + mc + (m+1)s, \qquad \text{and} \\
C_m^f &= 2\tau_m + (m+1)c + (m+1)s.
\end{aligned}$$

Now, we show that $C_n^f < C_m^a$ if $|\tau_n - \tau_m| < s/2$. This implies that the CTS from m starts arriving at the sender only after the CTS from n is completely arrived at the sender. This in turn shows that the CTS's from these node will not collide, and they arrive in the order of their sequence number at the sender. Since, n and m are arbitrary, the result follows.

The proof is by contradiction. Suppose, $C_n^f \geq C_m^a$. This implies that

$$\begin{aligned}
2\tau_n + (n+1)c + (n+1)s &\leq 2\tau_m + mc + (m+1)s \\
\Rightarrow 2(\tau_m - \tau_n) + (m-n-1)c + (m-n)s &\leq 0 \\
\Rightarrow -2|\tau_m - \tau_n| + (m-n-1)c + (m-n)s &\leq 0 \\
\Rightarrow -2|\tau_m - \tau_n| + s &\leq 0 \qquad \text{(since } n < m\text{)} \\
\Rightarrow |\tau_m - \tau_n| &\geq \frac{s}{2} \qquad \text{(Contradiction.)}
\end{aligned}$$

■

```
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 quorum T based on the queue length as described in Section IV;
  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 quorum T based on the queue_length as described in Section IV;
      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
```

Fig. 5. 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 propagation delay;

if (Data packet begins to arrive before RcvTimer expires) **then**

Receive the data packet and pass it to the higher layer;

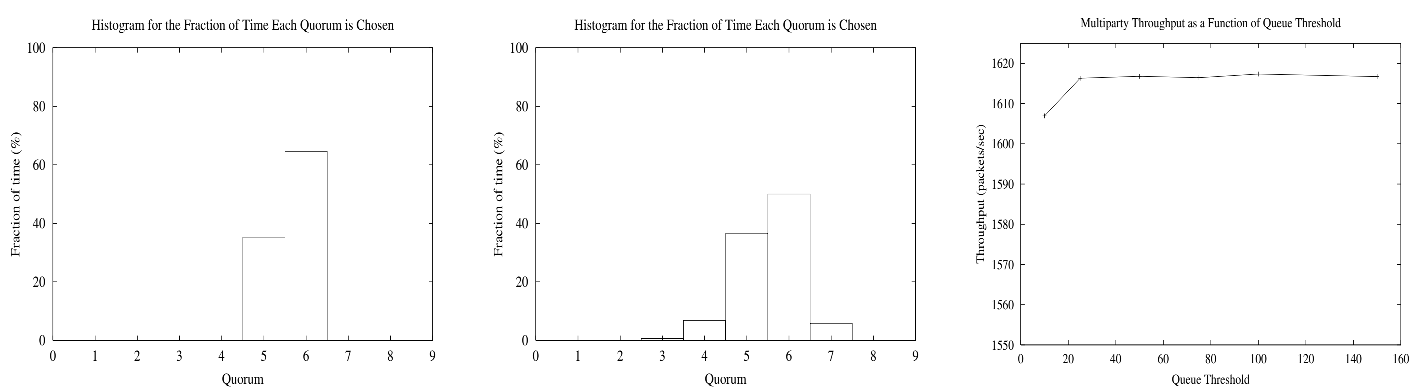
Stop the busy tone;

else

Stop the busy tone;

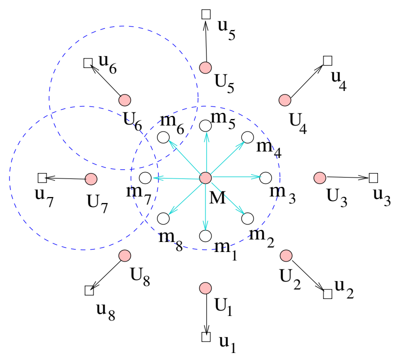
end

Fig. 6. Pseudo code describes the busy tone based MAC layer protocol at the receiver.

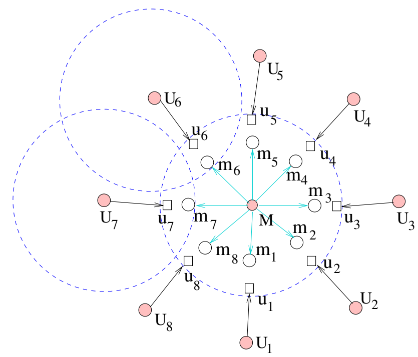


(a) $\Gamma = 75$ (b) $\Gamma = 10$ (c)

Fig. 7. We consider topology 1 with $\lambda_M = 200$ packets/sec and $\lambda_U = 500$ packets/sec. Figures (a) and (b) plot the fraction of time each quorum is chosen by $\Delta_O(\Gamma)$ for Γ equal to 75 and 10, respectively. Figure (c) shows the throughput of the multiparty session under $\Delta_O(\Gamma)$ as a function of Γ. In the figures, Γ is referred as *queue threshold*.



(a) Topology 1



(b) Topology 2

Fig. 8. We evaluate the the performance of various multiparty strategies in topologies 1 and 2. Each topology has a multiparty 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 u_is, are not in M's transmission range in topology 1, while they are in M's transmission range in topology 2.

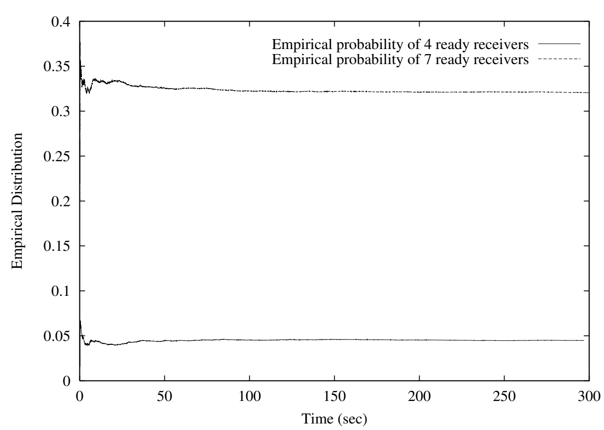
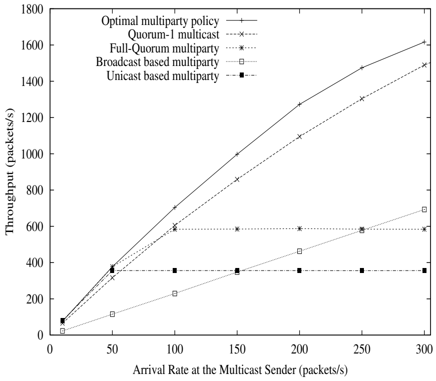
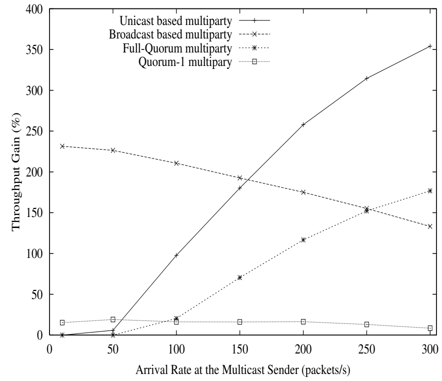


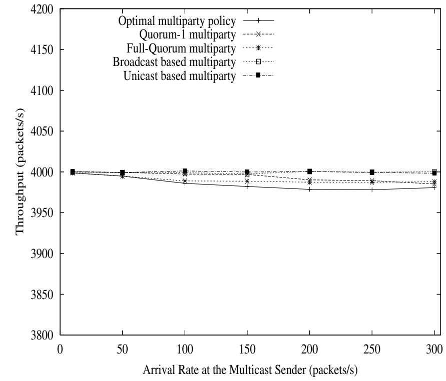
Fig. 9. Figure shows $\frac{n_u(t)}{t}$ for u equal to 4 and 7.



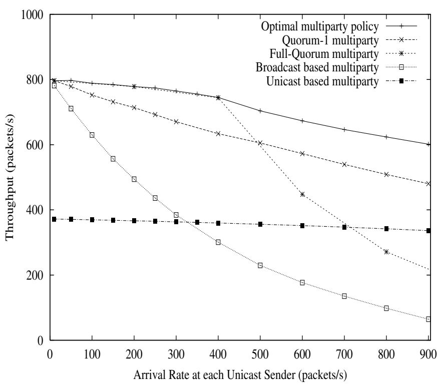
(a) Throughput of Multiparty Session Vs. λ_M



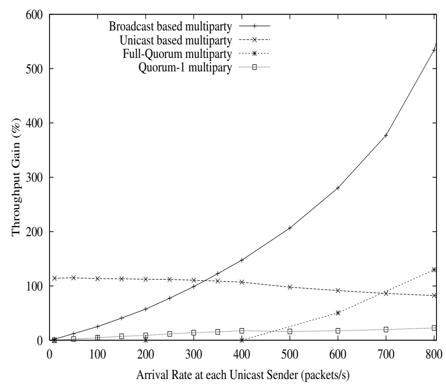
(b) Throughput Gain Vs. λ_M



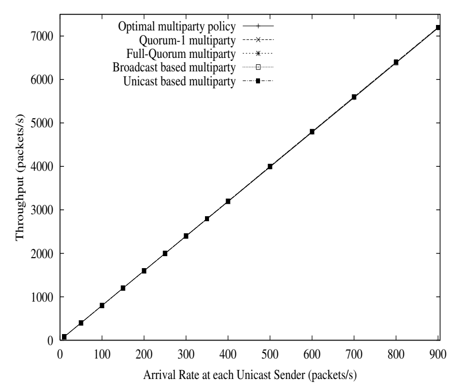
(c) Throughput of Unicast Sessions Vs. λ_M



(d) Throughput of Multiparty Session Vs. λ_U

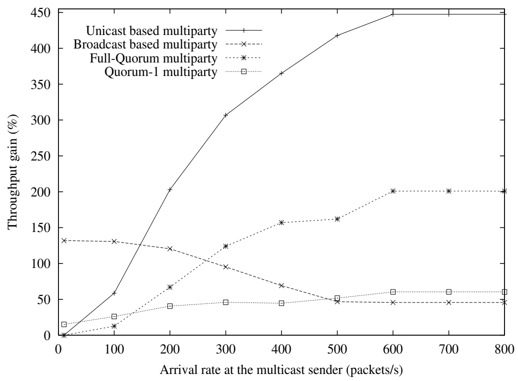


(e) Throughput Gain Vs. λ_U

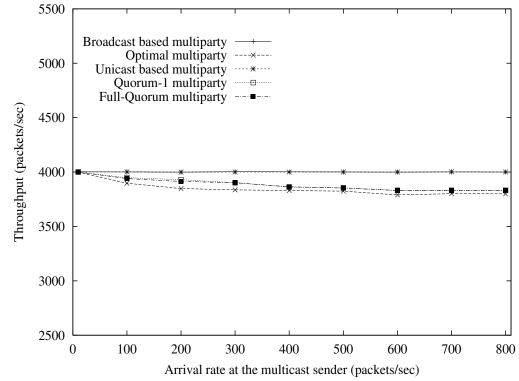


(f) Throughput of Unicast Sessions Vs. λ_U

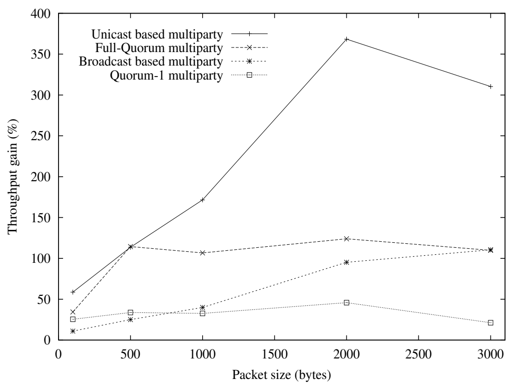
Fig. 10. The figure plots the performance of various transmission strategies in topology 1. Figures (a) and (c) plot the throughputs of the multiparty and unicast sessions, respectively, under various multiparty policies as a function of λ_M. Figure (b) plots the throughput gain of $\Delta_O(\Gamma)$ over other polices as a function of λ_M for the multiparty 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, and multiparty packet size (P_M) is 3000 bytes. Figures (d) and (f) plot the throughputs of the multiparty 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 multiparty session. Here, $\lambda_M = 50$ packets/sec, and $\Delta_O(\Gamma)$ is referred as *Optimal multiparty policy*.



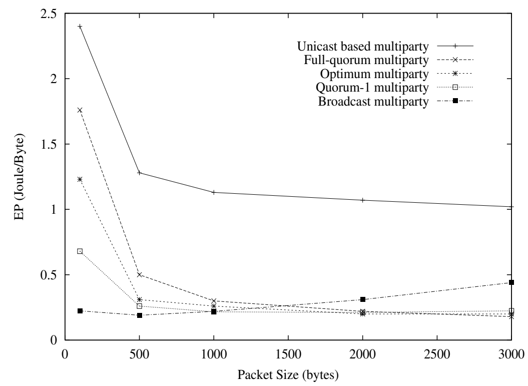
(a) Throughput Gain Vs. λ_M



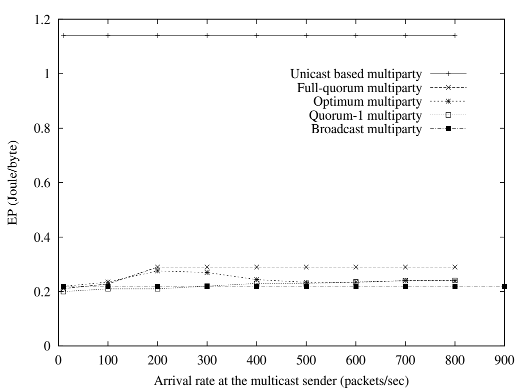
(b) Throughput of Unicast Sessions Vs. λ_M



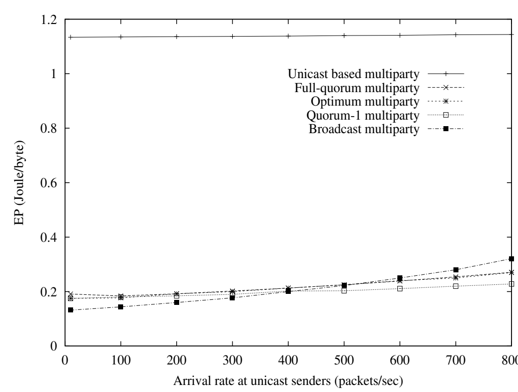
(c) Throughput Gain Vs. P_M



(d) EP Vs. P_M

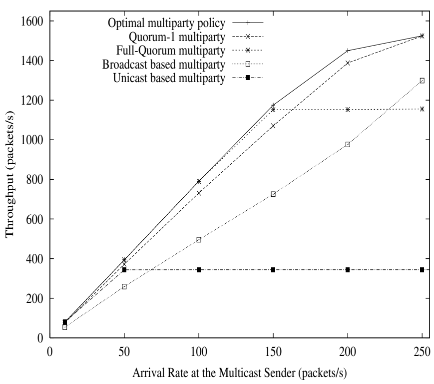


(e) EP Vs. λ_M

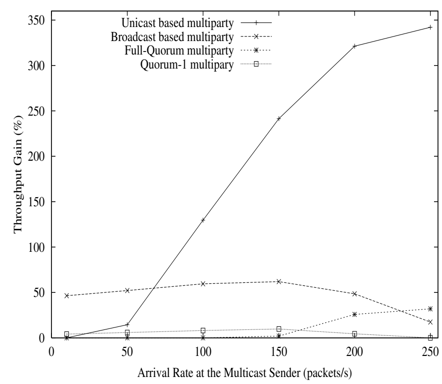


(f) EP Vs. λ_U

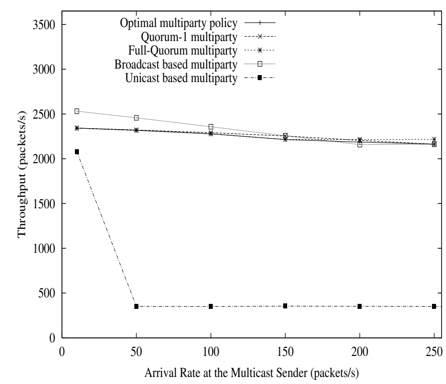
Fig. 11. We evaluate the the performance of various multiparty strategies in the topology shown in Figure 7(a). In all the figures, $\lambda_U =$ 500 packets/sec, and $\Delta_O(\Gamma)$ is referred as *Optimal multiparty policy*. Figure (a) plots the throughput gain of $\Delta_O(\Gamma)$ over other polices as a function of λ_M for the multiparty session. The throughput gain of $\Delta_O(\Gamma)$ over Δ is computed as $\frac{100\times(\Omega^{\Delta_O(\Gamma)}-\Omega^{\Delta})}{\Omega^{\Delta}}$%. In Figure (b) we plot the throughput of unicast sessions under various policies. In Figures (a) and (b), P_M is 2000 Bytes. In Figures (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/sec. EP is computed as $\frac{P_M+O}{g+P_M}$ Joules/byte. In Figures (e) and (f), we plot EP as a function of λ_M and λ_U respectively. In Figure (e), $P_M = 1000$ bytes and $\lambda_U = 500$ packets/sec. In Figure (f), $P_M = 1000$ bytes and $\lambda_M = 50$ packets/sec.



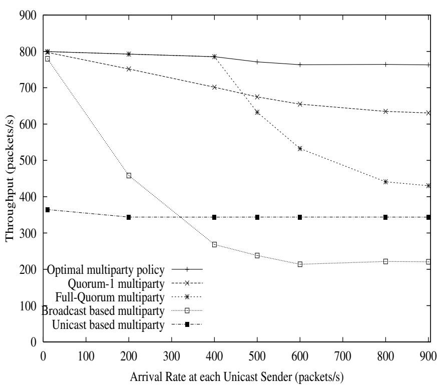
(a) Throughput of Multiparty Session Vs. λ_M



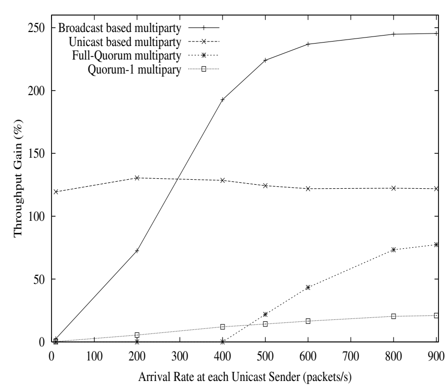
(b) Throughput Gain Vs. λ_M



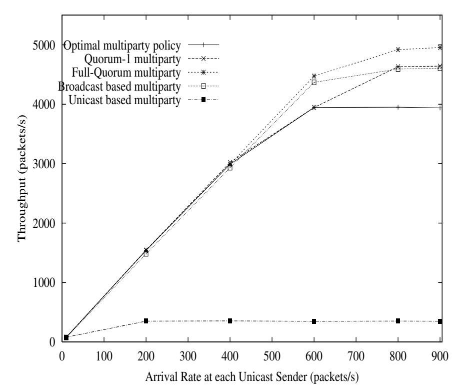
(c) Throughput of Unicast Sessions Vs. λ_M



(d) Throughput of Multiparty Session Vs. λ_U



(e) Throughput Gain Vs. λ_U



(f) Throughput of Unicast Sessions Vs. λ_U

Fig. 12. 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 except here $\lambda_U = 300$ packets/sec for figures (a), (b) and (c).

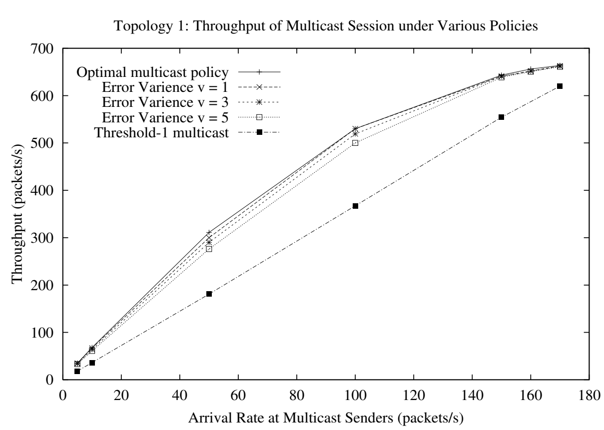
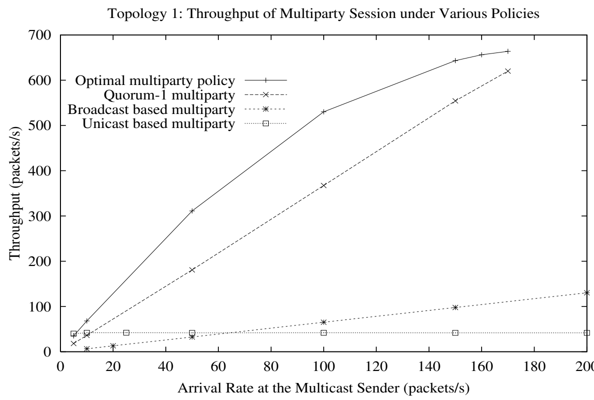
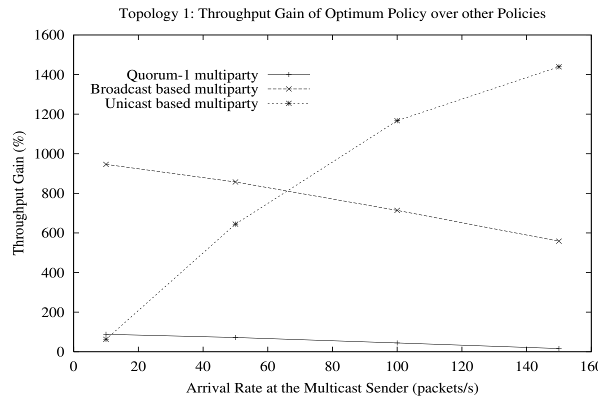


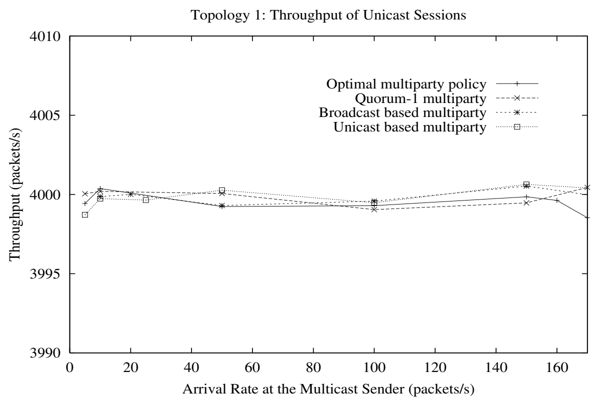
Fig. 13. We plot the performance of $\Delta_O(\Gamma)$ when the multiparty sender M makes random error E_t in estimating the number of ready receivers at time t. We assume that $|E_t|$ is binomial random variable with mean 0 and variance v, and E_t is positive with probability 1/2.



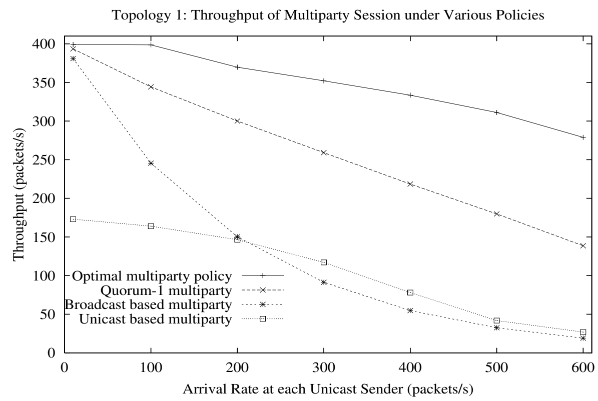
(a) Throughput of Multicast Session Vs. λ_M



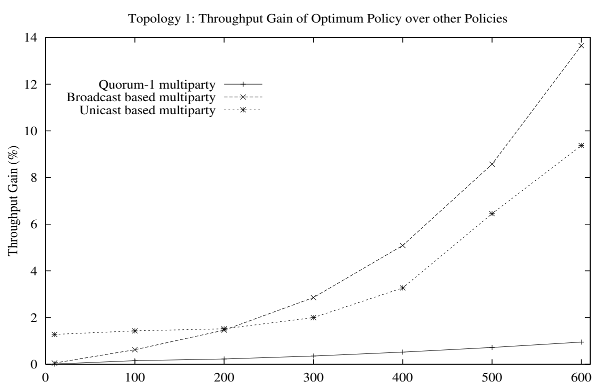
(b) Throughput Gain Vs. λ_M



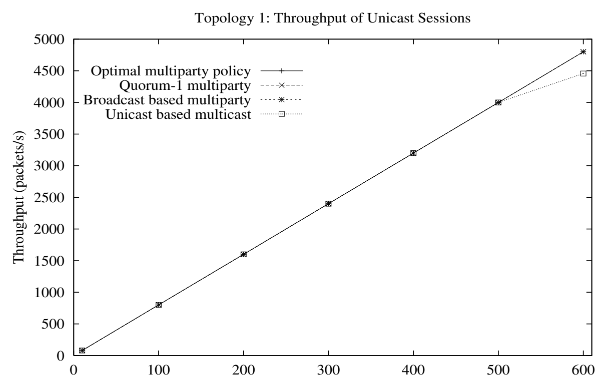
(c) Throughput of Unicast Sessions Vs. λ_M



(d) Throughput of Multicast Session Vs. λ_U

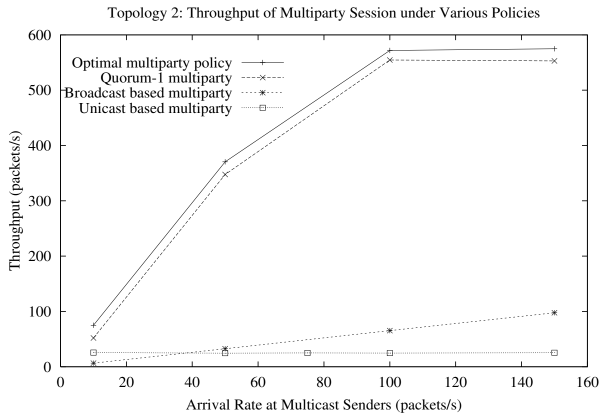


(e) Throughput Gain Vs. λ_U

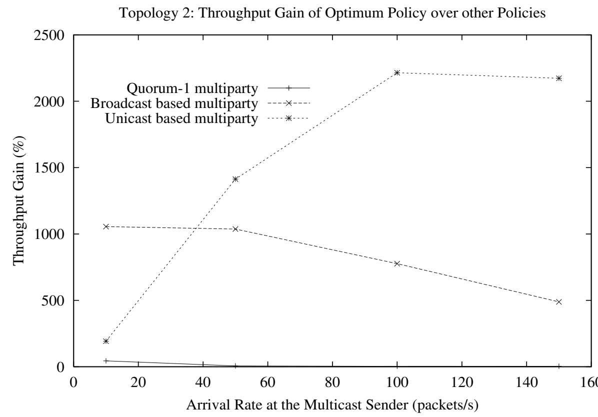


(f) Throughput of Unicast Sessions Vs. λ_U

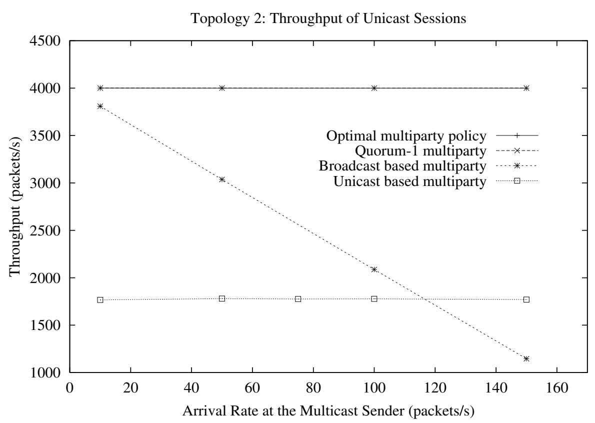
Fig. 14. The figure plots the performance of various transmission strategies in topology 1. Figures (a) and (c) plot the throughputs of the multiparty and unicast sessions, respectively, under various multiparty policies as a function of λ_M. Figure (b) plots the throughput gain of $\Delta_O(\Gamma)$ over other polices as a function of λ_M for the multiparty 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 multiparty 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 multiparty session. Here, $\lambda_M = 50$ packets/sec. In the figures, $\Delta_O(\Gamma)$ is referred as *Optimal multiparty policy*.



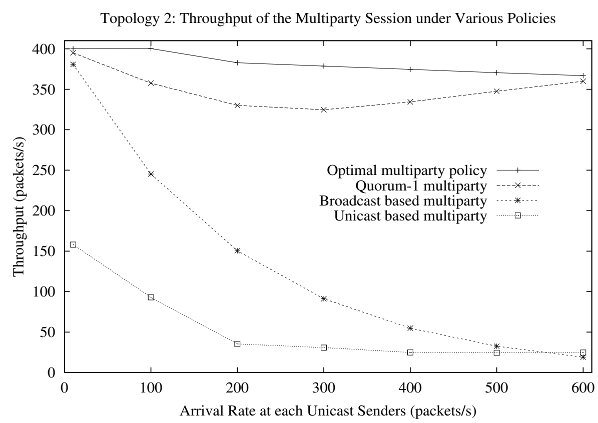
(a) Throughput of Multicast Session Vs. λ_M



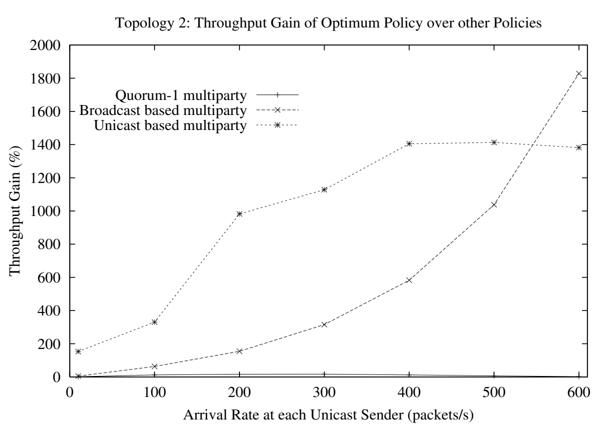
(b) Throughput Gain Vs. λ_M



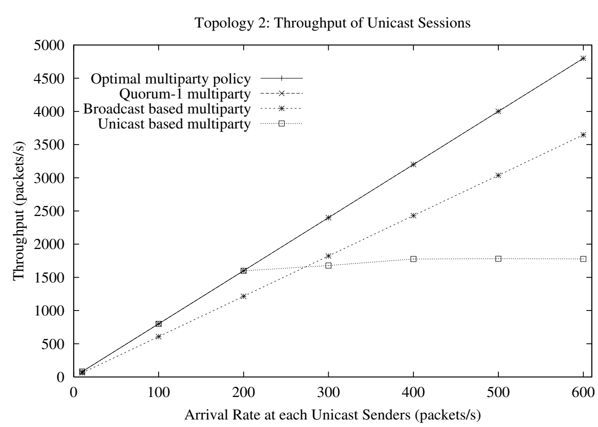
(c) Throughput of Unicast Sessions Vs. λ_M



(d) Throughput of Multicast Session Vs. λ_U



(e) Throughput Gain Vs. λ_U



(f) Throughput of Unicast Sessions Vs. λ_U

Fig. 15. 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 14.

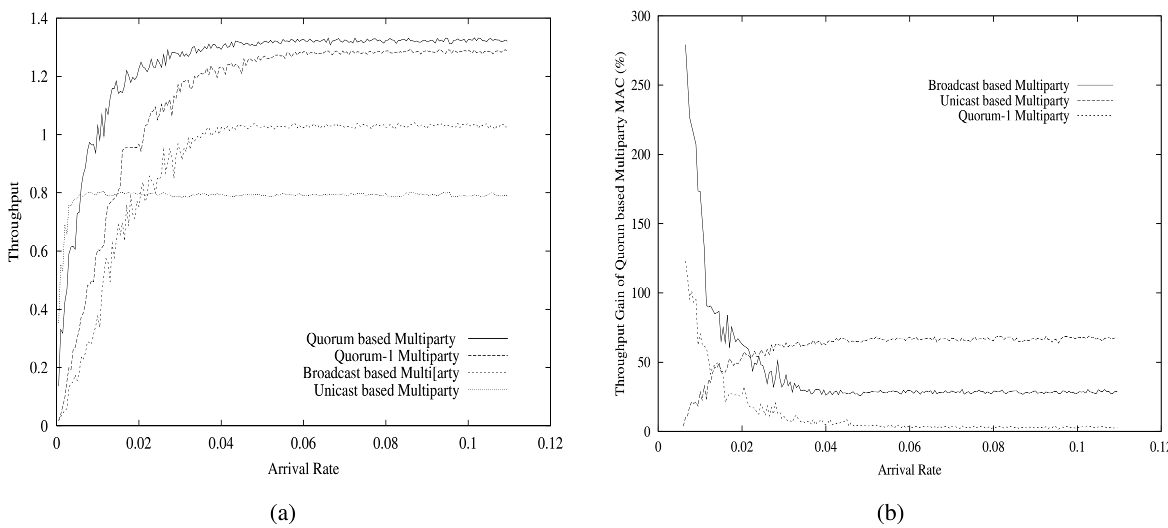


Fig. 16. Figure shows the throughput and throughput gain of the quorum based multiparty over the existing policies