On-line Optimal Wireless Multicast

Prasanna Chaporkar and Saswati Sarkar

University of Pennsylvania, Philadelphia PA 19104, USA
prasanna@seas.upenn.edu, swati@ee.upenn.edu

Abstract. The paper is directed towards designing bandwidth efficient strategies for MAC layer multicast. Bandwidth efficiency of wireless multicast can be improved substantially by exploiting the fact that several receivers can be reached at the MAC layer by a single transmission. However, this feature can be effectively exploited only by resolving several decision problems and protocol challenges. The first issue is to determine when a sender should transmit in order to maximize its throughput while maintaining the system stability. We provide an on-line strategy and prove using the large deviation theory that it solves the decision problem optimally without requiring any system statistics. The protocol challenge is to design a message exchange protocol to acquire the information necessary for execution of the optimal transmission strategy. We provide a MAC protocols for this purpose. The performance evaluations demonstrate that the optimal strategy significantly outperforms the existing approaches in multi-hop multicast wireless networks.

1 Introduction

1.1 Motivation

Many current day wireless applications need *multicast* communication, e.g., sensor networks and military operations. Wireless communication is inherently broadcast in nature, i.e., all nodes in the transmission range of a sender can receive a transmission from the sender. This broadcast nature of the transmissions can be used to enhance bandwidth efficiency of wireless multicast. Figure 1 illustrates the point. A single transmission from the sender S_1 can reach all its receivers $R_1, \ldots, R_4$. Hence, at the MAC layer, it suffices to transmit each packet only once in order to reach all the intended receivers. Thus, multicast in the wireless case can be more efficient than the wireline counterpart as in the latter case a packet has to be transmitted on each link (from sender to each of the receivers) separately. *We focus on designing an optimal MAC strategy for bandwidth efficient wireless multicast.*

Though the broadcast nature of wireless transmissions provides a possible approach to improve the efficiency of the multicast communication, it also imposes various difficulties. A multicast specific challenge is that some but *not all* the receivers may be ready to receive. For example, when sender S_2 is transmitting to receiver R_5, receiver R_2 can not receive the transmission from sender S_1 as both the transmissions will collide at R_2. But receivers R_1, R_3 and R_4 can still receive the transmission. The readiness state of a receiver depends on the network load. The policy decision in this case is whether the sender S_1 should transmit or it should wait till all the receivers are ready. A transmission policy that does not transmit until a sufficient number of receivers are ready may render the system unstable. On the other hand, if the sender transmits 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 design of MAC layer transmission strategy involves the following issues.

1) When should a sender transmit to maximize its throughput while maintaining system stability? This is a *decision problem* at the sender.

2) What information about the receiver's state should be available at the sender to obtain efficient solution for the decision problem? How can this information be made available? These are the *protocol design problems*.

1.2 Literature Review

Most of the research efforts in this area are focused on the development of transport and network layer protocols. End-to-end error recovery protocols primarily address an issue of reliable loss recovery with minimum cost of information exchange among nodes, e.g., [1, 2]. At network layer, several multicast routing protocols have been proposed, e.g., [3–7]. Protocols for energy efficient multicast routing have been proposed in [8–10]. Zhou *et al.* have investigated content based multicast in adhoc networks [11]. Multicast in cellular networks has been investigated by Nagy *et al.* [12]. The proposed transport layer and network layer protocols can potentially work with any underlying MAC layer strategy. Though efficiency of these higher layer schemes depends on the efficiency of MAC layer strategy, MAC layer multicast has not been adequately explored.

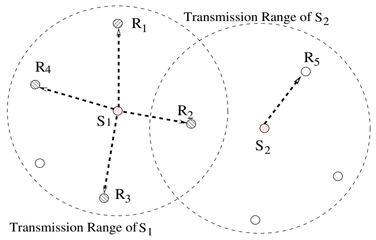


Figure 1. An example to demonstrate the advantages and the challenges associated with wireless multicast. The figure shows two senders S_1, S_2 and 5 receivers R_1 to R_5. Dashed circle indicates the communication range of a sender.

At MAC layer, a protocol for power aware broadcast is proposed [8]. Wang *et al.* proposed a scheduling and power control protocol to minimize transmit power at the sender nodes [13]. Jaikaeo *et al.* have studied multicast with directional antennas [14]. Kuri *et al.* have proposed a protocol for reliable packet delivery in wireless LANs [15]. The design of the protocol is based on many assumptions that hold in wireless LANs but not in multi-hop wireless networks. For multi-hop wireless networks, Tang *et al.* have proposed a reliable multicast scheme that transmits a packet to each receiver separately in round robin fashion [16]. Currently popular multiple access standard IEEE 802.11 implements multicast by broadcasting a packet after disabling all control messages. Thus, second hop interference is ignored. The former *unicast based multicast* policy does not exploit the broadcast nature of wireless medium, while the latter *broadcast based multicast* policy causes packet loss at receivers because of collision due to second hop interference. We have proposed a throughput optimal strategy, which depends on the network load and the statistics of the arrival process [17]. However note that the necessary information about the system statistics may not be available at the sender.

1.3 Our Contributions

We propose an adaptive transmission strategy that does not require any statistical information (Section 3). Using large deviation results, we prove that the proposed policy is throughput optimal. Our analytical model is presented in Section 2. The transmission decisions of the proposed policy are based on the queue length at the sender node and the number of ready receivers. The first quantity is easily available at the sender node. We propose a MAC protocol that allows a sender to obtain the number of ready receivers (Section 4). We evaluate the performance of various multicast schemes via simulations in a wireless network consisting of several multi-hop multicast sessions (Section 5). Simulation results show that the proposed optimal policy provides significantly higher throughput. We conclude in Section 6.

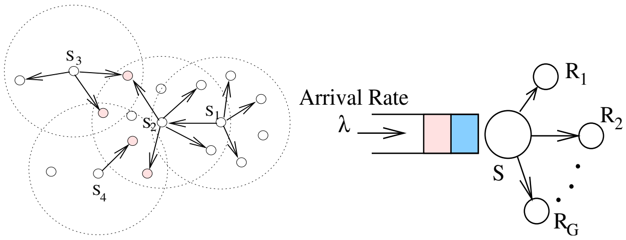
We would like to mention that we have not considered some implementation specific issues, e.g., node mobility, dynamic group membership changes, security concerns, etc. The paper's contributions are primarily in solving the bandwidth management problem in wireless multicast from a theoretical perspective. This is a prerequisite towards developing an operational MAC layer multicast protocol. A MAC protocol presented in Section 4 constitutes a first step in that direction. In future research, we plan to investigate the implementation specific issues in further details.

2 System Model

We consider a wireless network with several MAC layer multicast sessions. A typical scenario is shown in Figure 2(a). Each multicast session comprises of a sender and a set of receivers (multicast group). At the MAC layer all the receivers are within the transmission range of the sender. The scenario described above is motivated by the network level view of the multicast communication in a multi-hop wireless network.

We assume a slotted time axis. The arrival process is modeled as an irreducible, aperiodic and time homogeneous Markov Chain (MC) of γ states. A state of the MC is 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 steady state distribution. Every packet can be transmitted in a single slot.

For analytical purposes, we consider a single multicast session in isolation (refer to Figure 2(b)). The impact of the network and the channel errors on the multicast session is that the receivers are not always ready to receive. The receivers may not be ready because of transmissions in their neighborhood or on account of the power saving mode. Thus, the readiness states of different receivers are correlated in the same time slot. The correlation across time slots is due to bursty channel errors. We model the readiness process of all the receivers as an irreducible, aperiodic and time homogeneous markov chain. Let G be the number of receivers for the MAC layer multicast



(a) (b)

Figure 2. (a) Figure shows four MAC layer multicast sessions. Nodes S_1 to S_4 are the multicast senders that multicast packets to the receivers in their neighborhood. Arrows from a sender point to its designated receivers. (b) Figure shows a MAC layer multicast session that we study analytically. Node S is a sender, where packets arrive at the rate λ. Sender S transmits the packets to receivers R_1 to R_G. The receivers are ready to receive in accordance with a stochastic process that models the channel errors and the interference due to other transmissions.

session under consideration. A state of the MC is considered to be a G dimensional vector $\boldsymbol{j} = [j_1\ j_2\ \dots\ j_G]$, where the component j_l is 1 if the l^{th} receiver is ready and it is 0 otherwise. Let b_u be the unique steady state probability that u receivers are ready to receive.

We relax the analytical assumptions in simulations. Using simulations, we study the end-to-end performance of multi-hop multicast sessions when the proposed MAC layer strategy is used at each hop.

We consider transmission of data traffic and assume FIFO selection of packets for transmission. Our performance metric is system throughput. We now present some definitions.

Definition 1 *A transmission policy is an algorithm at a sender node that decides when to transmit a packet.*

Definition 2 *A reward for a transmission is the number of receivers that receive the transmission successfully.*

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

Definition 4 *A system is said to be stable if the mean queue length is bounded. Further, a transmission policy that stabilizes the system is called a stable policy.*

Definition 5 *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 achieved by Δ.*

A sender may transmit a packet several times till sufficient number of receivers receive the packet, e.g., in Figure 1, sender S_1 may transmit a packet to the receivers R_1, R_3 and R_4 even when the receiver R_2 is not ready and then retransmit the packet when receiver R_2 becomes ready. But each additional transmission consumes additional energy at the sender. Thus we assume that a packet can be transmitted at most once at the MAC layer on account of energy constraints. Hence, a packet is lost at a receiver if the receiver is not ready at the time of transmission. In multicast, small loss tolerance increases throughput significantly. The lost information can be recovered by using coding redundancy. If the loss is constrained to a small value, then a higher layer protocol can also be used to retrieve the lost packets without significant performance penalty.

Now we relate the throughput and the loss for a stable system. We consider the average loss experienced by all the receivers rather than the individual loss experienced by any receiver. This is because a lost packet can often be retrieved easily from the neighboring receivers. In a stable system, the throughput of a transmission policy is λR, where R is the average reward received by the policy per transmission. Thus, we need to maximize the mean reward R in order to maximize the throughput subject to the system stability. The average loss experienced by all the receivers in the system is $G - R$. Thus, a throughput optimal policy minimizes the average loss at the receivers. Hence, if the mean loss for the throughput optimal transmission policy is more than the required loss constraint, it implies that the required loss constraint can not be guaranteed by any stable transmission policy. So, we do not consider the loss constraints explicitly in this paper.

3 Throughput Optimal On-line Transmission Policy ($\Delta_O(\Gamma)$)

We describe a parameterized transmission policy $\Delta_O(\Gamma)$ that we prove to be ϵ-throughput optimal. The policy selects a threshold value based on the queue length at the sender in each slot. A packet is transmitted if (a) the number of ready receivers is greater than or equal to the threshold and (b) the sender has a packet to transmit. The threshold values are selected as follows. Let Q denote the queue length at the sender and let Γ be some fixed positive integer. For $T \geq 1$, the threshold is set T if $(G-T)\Gamma < Q \leq (G-T+1)\Gamma$ and threshold is set to 0 if $Q > G\Gamma$. Broadly, the threshold value is small(large) when the queue length is large(small).

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

Theorem 1. *Suppose there exists some policy that stabilizes the system. Then, the policy $\Delta_O(\Gamma)$ is also stable for every $\Gamma \geq 0$. Further, for any given $\epsilon > 0$ there exists $\widehat{\Gamma}$ such that $\Delta_O(\Gamma)$ is ϵ-throughput optimal for every $\Gamma \geq \widehat{\Gamma}$. Formally, $\Omega_{opt} - \Omega^{\Delta_O(\Gamma)} \leq \epsilon$ w.p. 1.*

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 the policy $\Delta_O(\Gamma)$. This holds even though $\Delta_O(\Gamma)$ takes transmission decisions based on only the current packet availability and the number of ready receivers at the current time.

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

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

The throughput can be further improved by randomizing between the threshold values T_O and T_O+1 appropriately. The randomization should be such that the system stability is maintained. This is the basic idea behind the design of the static optimum policy [17]. Intuitively an on-line optimum policy should select the thresholds T_O and T_O+1 most of the time. The difficulty, however, is that the λ and b_u's are not known.

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

The value of $\widehat{\Gamma}$ depends on the system parameters. However, since the optimality of $\Delta_O(\Gamma)$ is guaranteed for all large Γ only a rough estimate of $\widehat{\Gamma}$ is necessary. The time required for convergence increases with increase in Γ. Thus, a large convergence time is the cost we pay for inaccurate knowledge of $\widehat{\Gamma}$. Also, the proposed policy can be executed at each node for multi-hop sessions; there is, however, no guarantee that it attains the maximum end-to-end throughput. The simulation in a multi-hop network, however, demonstrates significant throughput improvement over the existing MAC layer multicast strategies (Section 5).

The policy $\Delta_O(\Gamma)$ provides a computationally simple transmission rule. For implementation of $\Delta_O(\Gamma)$ the sender needs the number of ready receivers. We propose a possible MAC protocol in the next section to obtain the required information at the sender.

4 MAC Layer Protocol

In this section, we propose a MAC protocol to implement $\Delta_O(\Gamma)$. First we present the system challenges experienced by the existing IEEE 802.11 protocol in case of multiple receivers. IEEE 802.11 exchanges RTS-CTS-DATA-ACK for reliable communication. The control message exchange works well if there is only one intended receiver for each session. If all the nodes in a multicast group send CTS simultaneously in response to a RTS from a sender, then these CTS messages will collide at the sender. Hence, the sender will not know whether the receivers are ready to receive the data. Similar problems exist for ACK message. Hence, we need to modify IEEE 802.11 protocol.

We describe a sequential CTS transmission scheme that eliminates the CTS collisions. The sender allots a unique sequence number $i \in \{0, \ldots, G-1\}$ to each receiver. When a sender wishes to transmit a packet, it first sends an RTS addressed to the MAC layer multicast group. 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, then the CTS's will reach the sender without collision. The sender transmits the packet if

```
Procedure Multicast_MAC()
begin
  MAC at the sender node
  if (sender's queue length > 0) then
    Transmit an RTS message with duration/ID field value 1 + Gc + (G + 1)s;
    /* Nodes in the neighborhood of the sender set their NAV based on duration/ID field value */
    Wait for CTS responses for duration Gc + (G + 1)s;
    if (Number of CTS messages received ≥ Threshold value) then
      /* Threshold is determined by the queue length at the sender as explained in Section 3 */
      Transmit a packet;
    else
      Transmit a Release message;
      /* Nodes in the neighborhood of the sender reset their NAV */

  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 1 + (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
```

Figure 3. Pseudo code for MAC protocol to implement the optimal policy $\Delta_O(\Gamma)$

the number of CTS responses it received is greater than or equal to threshold value. Recall that the threshold value is determined by the queue length at the sender. Note that the protocol does not need clock synchronization.

By RTS-CTS control exchange, the senders and receivers reserve the medium for the duration of packet transfer. RTS and CTS messages contain duration/ID field which specifies the duration for which the medium is required for the subsequent transmission. The nodes in the neighborhood of the sender (receivers) set their Network Allocation Vector (NAV) equal to the maximum of the current value and the value of duration/ID field in RTS (CTS) message, and does not transmit a packet in NAV duration. Recall that every packet can be transmitted in unit time duration. Hence, the value of duration/ID field for RTS message is $1 + Gc + (G + 1)s$. Further, the value of the field in the CTS message by i^{th} receiver is $1 + (G - i - 1)c + (G - i)s$.

In unicast case if the RTS-CTS exchange is successful then the source always transmits a data packet. In other words, if the resources are reserved, then their utilization is guaranteed. But in the multicast case, depending on the number of CTS messages received, the sender may not transmit a packet even after an RTS-CTS exchange. We augment the protocol to enable the neighboring nodes of sender and receivers to utilize channel in this case. If the sender decides not to transmit a packet, then it transmits a release message. On receiving the release message from the sender, 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. Pseudo code for the MAC protocol is given in Figure 3.

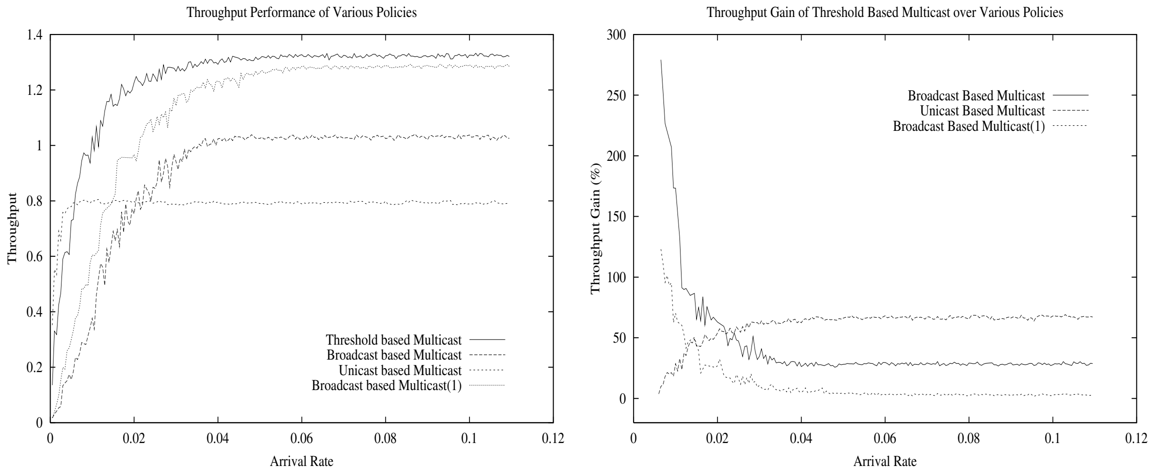
This control message exchange introduces overhead on data communication. The overhead increases as the value of G increases. We expect that G will be moderate in practical scenario. If the packets are much larger than the CTS, then the ration of control overhead and payload is small, and the protocol provides high efficiency.

5 Simulation Results and Discussion

We have considered a single MAC layer multicast session in isolation. In this section, we study the performance of the optimal policy in wireless networks with several multi-hop multicast sessions contending for bandwidth. Note that the readiness states are generated on account of the actual transmissions in the network. The simulation results substantiate the claim that the optimal policy provides substantial bandwidth advantage as compared to the other existing approaches.

5.1 Simulation Setup

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 [18]. 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 multicast routing tree from a sender



(a) (b)

Figure 4. Figure shows the throughput and throughput gain of the threshold based multicast over the existing policies

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 multicast session (e.g., refer to Figure 2).

We place 500 nodes uniformly in a square of unit area. The first 10 nodes are session sources. A node is a multicast 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 multicast 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 multicast 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 multicast MAC proposed in Figure 3 for simulating the optimal strategy $\Delta_O(\Gamma)$. We refer to this implementation as threshold based multicast. 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 multicast sessions are stored in a single queue and the packet scheduling is FIFO over all the sessions. We compare the performance of the threshold based scheme with unicast based multicast, broadcast based multicast and broadcast based multicast(1) proposed in [19]. In the last scheme, a sender transmits whenever at least one receiver is ready.

5.2 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 threshold based multicast 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 4 compares the throughput performance of the schemes. We observe that the threshold based multicast strategy obtains significantly higher throughput than the other schemes (Figure 4(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 threshold based multicast 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 threshold based multicast strategy sets the minimum threshold.

We now explain the performance difference between the threshold and the unicast based multicast schemes. Unicast based multicast uses separate transmissions to reach different members of the multicast 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 4(a) shows, the throughput of the unicast based multicast saturates at small values of arrival rates. The unicast based multicast 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 threshold based multicast scheme has significantly higher throughput.

6 Conclusion

We design a throughput optimal on-line policy. The important feature of the policy is that it achieves the maximum throughput without requiring the statistics of the arrival or the receiver readiness process. We prove the optimality of the policy using results from the large deviation theory. The proposed policy requires the number of ready receivers to take transmission decisions. We propose a MAC layer protocol that can acquire the necessary information. The simulation studies demonstrate that the proposed policy significantly outperforms the existing approaches even in multi-hop wireless networks. The framework and the proof can be extended to the continuous time case with the Poisson arrivals.

References

1. Chandra, R., Ramasubramaniam, V., Birman, P.: Anonymus Gossip: Improving multicast reliability in mobile ad-hoc networks. In: 21st International Conf on Distr Computing Systems. (2001)
2. Pagani, E., Rossi, G.: Reliable broadcast in mobile multihop packet networks. In: MOBICOM'97. (1997) 34–42
3. Chiang, C.C., Gerla, M., Zhang, L.: Forwarding group multicast protocol (FGMP) for multihop, mobile wireless networks. Baltzer Cluster Computing **1** (1998) 187–196
4. Lee, S.J., Gerla, M., Chiang, C.C.: On-demand multicast routing protocol. In: Proceedings of IEEE WCNC'99. (1999)
5. Ozaki, T., Kim, J.B., Suda, T.: Bandwidth-efficient multicast routing for multihop, ad-hoc wireless networks. In: INFOCOM'01. (2001)
6. Royer, E.M., Perkins, C.E.: Multicast operation of the ad-hoc on-demand distance vector routing protocol. In: Proceedings of ACM/IEEE MOBICOM'99. (1999)
7. Sinha, P., Shivkumar, R., Bharghavan, V.: MCEDAR: Multicast core-extraction distributed ad hoc routing. In: Proceedings of IEEE WCNC'99. (1999)
8. Singh, S., Raghavendra, C.S., Stepanek, J.: Power efficient broadcasting in mobile ad hoc networks. In: PIMRC'99. (1999)
9. Wan, P.J., Calinescu, G., Li, X., Frieder, O.: Minimum-energy broadcast routing in static ad hoc wireless networks. In: INFOCOM'01. (2001)
10. Wieselthier, J., Nguyen, G., Ephermides, A.: On the construction of energy-efficient broadcast and multicast trees in wireless networks. In: INFOCOM'00. (2000)
11. Zhou, H., Singh, S.: Content-based multicast for mobile ad hoc networks. In: Proc. Mobihoc 2000. (2000)
12. Nagy, M., Singh, S.: Multicast scheduling algorithms in mobile networks. Cluster Computing (1998)
13. Wang, K., Srinivasan, V., Chiasserini, C., Rao, R.: Joint scheduling and power control for multicasting in wireless ad hoc networks. In: IEEE VTC Fall 2003, Orlando, FL (2003)
14. Jaikaeo, C., chung Shen, C.: Multicast communication in ad hoc networks with directional antennas. In: ICCCN 2003, Dallas, Texas (2003)
15. Kuri, J., Kasera, S.: Reliable multicast in multi-access wireless LANs. In: INFOCOM'99. (1999)
16. Tang, K., Gerla, M.: MAC reliable broadcast in ad hoc networks. In: Proceedings of IEEE MILCOM'01. (2001)
17. Chaporkar, P., Sarkar, S.: Stochastic control techniques for throughput optimal wireless multicast. In: Control and Decision Conf. (CDC), Maui, Hawaii (2003)
18. Gupta, P., Kumar, P.R.: Critical power for asymptotic connectivity in wireless networks. In: Stochastic Analysis, Control, Optimization and Applications: A Volume in Honor of W.H. Fleming. (1998)
19. Tang, K., Gerla, M.: MAC layer broadcast support in 802.11 wireless networks. In: Proceedings of IEEE MILCOM 2000. (2000)
20. Chaporkar, P., Sarkar, S.: On-line optimal wireless multicast. Technical report, Univ. of Pennsylvania, http://www.seas.upenn.edu/∼prasanna (2003)

A Notation and General Properties

Because of space constraints, we only present the outline of the proof for Theorem 1. Refer to [20] for complete formal proof. First we present some definitions.

Definition 6 *The busy slots are the time slots in which the sender's queue is non-empty.*

Definition 7 *A policy Δ belongs to the class of generalized threshold policies, if it sets threshold $T(t) \in \{0, \ldots, G+1\}$ in every busy slot t based on arbitrary rules and then transmits a packet only when the sender and $T(t)$ or more receivers are ready. The threshold may be selected based on past, present and future arrivals and readiness states.*

Definition 8 *A single threshold transmission policy(T) (denoted by Δ_T) is a generalized threshold policy for which $T(t) = T$ in every busy slot t.*

Definition 9 *A single threshold 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 threshold policies. Without loss of generality, we assume that a generalized threshold policy chooses threshold $G+1$ when the sender's queue is empty. In this case, the choice of threshold 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 $Y_n^{\Delta} = (k, \boldsymbol{j}, a)$, where k is the queue length at the sender, $\boldsymbol{j}$ 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.

B Proof for ϵ-Throughput Optimality of On-line Policy $\Delta_O(\Gamma)$ (Theorem 1)

We use the following results to prove Theorem 1. For presenting the first result, we define some notation. Let Δ denote an arbitrary generalized threshold policy. Let $s_T^{\Delta}(t)$ denote the total number of slots in which threshold T is chosen till time t under policy Δ. We note that

$$\sum_{T=0}^{G} s_T^{\Delta}(t) = n_{\mathcal{B}}(t),$$

where $n_{\mathcal{B}}(t)$ is the total number of busy slots.

Lemma 1. *For any $\epsilon > 0$, a generalized threshold policy Δ is ϵ-throughput optimal w.p. 1 if the following condition holds.*

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

Intuition: We note that a stable policy needs to maximize the expected reward per packet in order to maximize its throughput. Hence, a policy Δ_1 can achieve higher throughput than policy Δ only if it can achieve higher reward infinitely often. Now, since policy Δ chooses threshold T or $T+1$ in most of the slots, it exploits almost all the slots with $T+1$ or more ready receivers. Further, the remaining packets achieve reward T. Hence, Δ_1 can achieve a higher reward only in the slots when Δ does not have a packet or the threshold chosen is not T or $T+1$. Since, such slots are few, the result follows.

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

Lemma 2. *If $\lambda < 1$, then the policy $\Delta_O(\Gamma)$ is stable for every $\Gamma < \infty$.*

Intuition: We note that if the queue length at the sender becomes greater than $G\Gamma$, then the policy $\Delta_O(\Gamma)$ transmits in every slot. This is because the threshold is chosen to be 0. Since, the steady state arrival rate $\lambda < 1$, the system is stable.

Lemma 3. *If a threshold T satisfies*

$$\sum_{u=T}^{G} b_u > \lambda, \tag{3}$$

then the single threshold policy Δ_T is a stable transmission policy.

Intuition: Since policy Δ satisfies (3), the rate at which the slots with T or more ready receivers arrive is higher than that of packet arrivals in a steady state. Hence, the expected busy period length is finite.

Lemma 4. *Consider any given $\epsilon > 0$ and a threshold T that satisfies (3). Then, for the policy Δ_T, there exists a value $\Gamma_1(\epsilon, T)$ such that for every $\Gamma \geq \Gamma_1(\epsilon, T)$*

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

Intuition: From Lemma 3, we know that if T satisfies (3), 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.

Lemma 5. *Consider any given $\epsilon > 0$, a threshold T such that*

$$\sum_{u=T}^{G} b_u < \lambda, \tag{5}$$

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

$$\sum_{k=0}^{B-\Gamma} \widehat{\pi}_k \leq \frac{\epsilon}{4G}. \tag{6}$$

Intuition: We note that if threshold T satisfies (5), then the rate at which slots with T or more ready receivers arrive is less than that of packet arrivals. Hence, the drift is positive under policy $\Delta_{T,B}$ and the queue length tend to increase. Thus, if the value of Γ increases, then the probability that the queue length is less than or equal to $B - \Gamma$ decreases.

Lemma 6. *Consider the queue length based on-line policy $\Delta_O(\Gamma)$. Let P_{T_-} denote the steady state probability that the queue length Q at the sender is greater than to $(G - T + 1)\Gamma$. If T satisfies (3), then*

$$P_{T_-} \leq \sum_{k=\Gamma+1}^{\infty} \pi_k. \tag{7}$$

Intuition: We use sample path arguments to establish a relation between the queue length under $\Delta_O(\Gamma)$ and that under Δ_T. In particular, we show that in any slot t if the queue length under $\Delta_O(\Gamma)$ is greater than or equal to $(G-T+1)\Gamma+1$, then the queue length under Δ_T is greater than or equal to $\Gamma+1$ on every sample path. Then, the result follows from Lemma 4.

Lemma 7. *Consider the queue length based on-line policy $\Delta_O(\Gamma)$. Let P_{T_+} denote the steady state probability that the queue length Q at the sender is less than or equal to $(G - T)\Gamma$. Furthermore, let $\widehat{\pi}_k$ be the steady state probability that the sender's queue length is k under policy $\Delta_{T,B}$, where the buffer capacity $B = (G - T + 1)\Gamma$. Then,*

$$P_{T_+} \leq \sum_{k=0}^{(G-T)\Gamma} \widehat{\pi}_k. \tag{8}$$

Intuition: In this case, we use sample path arguments to establish a relation between the queue length under $\Delta_O(\Gamma)$ and that under $\Delta_{T,B}$. In particular, we show that in every slot t the queue length under $\Delta_O(\Gamma)$ is greater than or equal that under $\Delta_{T,B}$ on every sample path. Then, the result follows from Lemma 5.

B.1 Proof of Theorem 1

Proof. In view of Lemmas 1 and 2, it suffices to show that there exists a threshold T_O such that (2) is satisfied. Let

$$T_O = \arg\max_{0\leq T\leq G} \left\{ \sum_{u=T}^{G} b_u > \lambda \right\}. \tag{9}$$

If $\lambda < \sum_{u=0}^{G} b_u$, then $0 \leq T_O \leq G$ exists. Note that since $\boldsymbol{b} = [b_0 \ \ldots \ b_G]$ is a probability measure, $\sum_{u=0}^{G} b_u = 1$.

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 2, we know that the DTMC is ergodic. Hence,

$$\lim_{t\to\infty} \frac{s_{T_O}^{\Delta_O(\Gamma)}(t) + s_{T_O+1}^{\Delta_O(\Gamma)}(t)}{t} = 1 - P_{(T_O+1)_+} - P_{T_O-} \quad \text{w.p. 1.} \tag{10}$$

Now, we fix $\Gamma \geq \max\{\Gamma_1(\epsilon, T_O), \Gamma_2(\epsilon, T_O+1)\}$. Since T_O satisfies (3) and T_O+1 satisfies (5), from Lemmas 4 and 5

$$\sum_{u=\Gamma+1}^{\infty} \pi_k \leq \frac{\epsilon}{4G} \qquad \text{and} \tag{11}$$

$$\sum_{u=0}^{(G-T_O-1)\Gamma} \widehat{\pi}_k \leq \frac{\epsilon}{4G}. \tag{12}$$

Thus, from Lemmas 6 and 7 the result follows.

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

$$\lim_{t\to\infty} \frac{s_{T_O}^{\Delta_O(\Gamma)}(t) + s_{T_O+1}^{\Delta_O(\Gamma)}(t)}{t} = 1 - P_{T_O-} \quad \text{w.p. 1 (from (10)).}$$

Further, since T_O satisfies (3), relation (11) holds by Lemma 4. Thus the result follows from Lemma 6.

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

$$\lim_{t\to\infty} \frac{s_{T_O}^{\Delta_O(\Gamma)}(t) + s_{T_O+1}^{\Delta_O(\Gamma)}(t)}{t} = 1 - P_{(T_O+1)_+} \quad \text{w.p. 1 (from (10)).}$$

Further, since $T_O + 1$ satisfies (5), relation (12) holds by Lemma 5. Thus the result follows from Lemma 5.

Now, we show that if $\lambda > 1$, then no policy can stabilize the system. Let $\delta = \lambda - 1$. 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

$$\lim_{t\to\infty} \frac{A(t) - z^{\Delta}(t)}{t} \geq \delta \quad \text{w.p. 1.} \tag{13}$$

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

$$\lim_{t\to\infty} \frac{A(t)}{t} = \lambda \quad \text{w.p. 1, and} \tag{14}$$

$$\lim_{t\to\infty} \frac{z^{\Delta}(t)}{t} \leq 1 = \lambda - \delta \quad \text{w.p. 1.} \tag{15}$$

Relation (13) follows from (14) and (15). $\square$