POWMAC: A Single-Channel Power-Control Protocol for Throughput Enhancement in Wireless Ad Hoc Networks

Alaa Muqattash and Marwan Krunz
Department of Electrical and Computer Engineering
The University of Arizona
Tucson, AZ 85721
{alaa,krunz}@ece.arizona.edu

Abstract*— Transmission power control (TPC) has great potential to increase the throughput of a mobile ad hoc network (MANET). Existing TPC schemes achieve this goal by using additional hardware (e.g., multiple transceivers), by compromising the collision avoidance property of the channel access scheme, by making impractical assumptions on the operation of the MAC protocol, or by overlooking the protection of link-layer acknowledgement packets. In this paper, we present a novel power controlled MAC protocol called POWMAC, which enjoys the same single-channel, single-transceiver design of the IEEE 802.11 Ad Hoc MAC protocol but which achieves a significant throughput improvement over the 802.11 protocol. Instead of alternating between the transmission of control (RTS/CTS) and data packets, as done in the 802.11 scheme, POWMAC uses an access window (AW) to allow for a series of RTS/CTS exchanges to take place before several concurrent data packet transmissions can commence. The length of the AW is dynamically adjusted based on localized information to allow for multiple *interference-limited* concurrent transmissions to take place in the same vicinity of a receiving terminal. Collision avoidance information is inserted into the CTS packet and is used to *bound* the transmission power of potentially interfering terminals in the vicinity of the receiver, rather than silencing such terminals. Simulation results are used to demonstrate the significant throughput and energy gains that can be obtained under the POWMAC protocol.

***Index Terms*—Power control, IEEE 802.11, ad hoc networks, throughput enhancement.**

I. Introduction

Extensive research efforts are being dedicated to the design of mobile ad hoc networks (MANETs). The interest in such networks is attributed to the flexibility offered by their distributed and infrastructureless nature, which allows for instant deployment and rerouting of traffic around failed or forged terminals. Given that today's military operations require communicating a large amount of information over a limited spectrum, one of the main challenges in designing MANETs for the military is to provide *high-throughput, reliable, and low-complexity* wireless access to mobile terminals. Several attempts have been made and many others are currently underway to address this issue [1], [32].

So far, the Ad Hoc mode of the IEEE 802.11 standard [2] has been used as the *de facto* MAC protocol for MANETs. This protocol uses a 4-way handshake to resolve channel contention; when a terminal, say A, wants to send data to another terminal, say B, it first sends a request-to-send (RTS) packet to B, which replies back using a clear-to-send (CTS) packet. The data transmission $A \rightarrow B$[1] can now proceed, and once completed, terminal B sends back an acknowledgement (Ack) packet to A. The RTS and CTS packets include the duration of the ensuing data packet and are needed to reserve a *transmission floor* for the subsequent data packet. Any other terminal that hears the RTS or the CTS message defers its transmission until the ongoing transmission is over. The CTS message prevents collisions with the data packet at the destination terminal B, while the RTS message prevents collisions with the Ack packet at the source terminal A. Terminals transmit their control and data packets at a *fixed (maximum) power level*.

Despite its appealing simplicity, the 802.11 MAC approach can be overly conservative, leading to an unnecessary reduction in network throughput. To illustrate, consider the situation in Figure 1, where terminal A uses its maximum transmission power (TP) to send packets to terminal B (we assume omnidirectional antennas, so a terminal's reserved floor is represented by a circle in the 2D space). According to the IEEE 802.11 scheme:

1) When terminal D hears A's RTS, it refrains from transmitting to terminal C to avoid corrupting A's reception of B's Ack packet. The inability of terminal D to transmit while A is transmitting its data packet is the well-known *exposed terminal problem.*
2) Terminal D also refrains from receiving from terminal C to avoid having its reception corrupted by A's data transmission.
3) Terminal E hears B's CTS and, therefore, refrains from transmitting to terminal F to avoid corrupting B's reception of A's data packet.
4) Terminal E also refrains from receiving from terminal F to avoid having its reception corrupted by B's Ack transmission.

However, it is not hard to show that the three transmissions $A \rightarrow B$, $C \rightarrow D$, and $E \rightarrow F$ can, in principle, proceed simultaneously if terminals are able to select their TPs appropriately. Enabling multiple transmissions to take place within the same neighborhood leads to an increase in network throughput and possibly a reduction in the overall energy consumption. The scheme proposed in this paper is intended to allow for such transmissions to take place.

The previous discussion motivates the need for an interference-aware transmission power control (TPC) protocol to improve network throughput by means of increasing the channel spatial reuse.

This work was supported by the National Science Foundation under grants ANI-0095626, ANI-0313234, and ANI-0325979, and by the Center for Low Power Electronics (CLPE) at the University of Arizona.

[1] Throughout this paper, the notation $j \rightarrow i$ indicates a data transmission from j to i and an Ack transmission from i to j. We also refer to the data transmitter (the Ack recipient) as the source, and to the data receiver (the Ack transmitter) as the sink. Finally, we use the term "activity" to mean either a transmission or a reception.

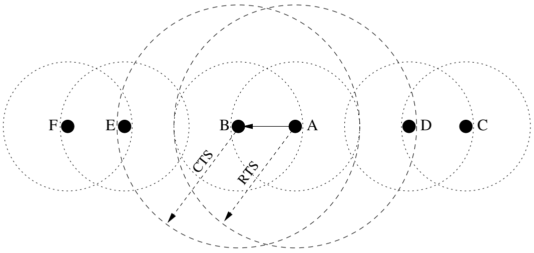


Fig. 1. Inefficiency of the classic RTS/CTS approach. Terminals A and B are allowed to communicate, but terminals D and E are not. Dashed circles indicate the maximum transmission ranges, while dotted ones indicate the ranges of the minimum transmission powers needed for coherent reception at the respective receivers.

Theoretical studies [16] and simulation results [23], [24] have demonstrated that TPC can provide significant gains in capacity and energy consumption, not to mention its benefits in providing admission control and in quality of service (QoS) provisioning [8].

Many TPC schemes for MANETs have been proposed in the literature. However, as explained in Section II, these schemes suffer from one or more of the following deficiencies: (1) the TPC approach may yield energy reduction but not throughput gain, (2) the MAC design may not support collision avoidance, resulting in the well-known hidden terminal problem, (3) the TPC approach requires extra hardware (e.g., multiple transceivers), (4) lack of link-layer reliability, i.e., Ack packets are not protected, and (5) many of the assumptions made in the MAC design are unrealistic. Accordingly, we introduce a new TPC scheme for MANETs that ameliorates these deficiencies. Our scheme is based on a single-channel, single-transceiver approach, and is shown to provide a significantly higher network throughput than the IEEE 802.11 scheme while yet preserving the collision avoidance properties of the IEEE 802.11 scheme. To the best of our knowledge, this is the first TPC solution that is based on a single-channel, single-transceiver design, that can increase the throughput of a MANET relative to the IEEE 802.11 scheme, and that supports link-layer reliability.

The rest of the paper is organized as follows. In Section II, we present related TPC schemes for MANETs and show their limitations. The proposed POWMAC protocol is presented in Section III, followed by simulation results and discussion in Section IV. Finally, our main conclusions are drawn in Section V.

II. Related Work

TPC schemes for MANETs can be generally classified into two classes. In the first class (e.g., [13], [33], [36], [39]), TPC is used to control the network topology, indirectly impacting the set of next-hop neighbors of a terminal and the subsequent routing decisions taken by that terminal. The *same* TP is used by a terminal to transmit its packets to any of its neighbors. This TP is updated following a mobility-related topological change. For pedestrian speeds, such a change occurs at a time scale of hundreds of milliseconds to seconds (in contrast, packet transmission times are, at most, in the order of few milliseconds). The main design issue here is how to determine the minimum TP for a given terminal such that some topological properties (e.g., connectivity, node degree, etc.) are guaranteed. One limitation of this class of protocols is its reliance solely on CSMA for accessing/reserving the shared wireless channel. It is known that using CSMA alone for accessing the channel can significantly degrade network performance (throughput, delay, and power consumption) because of the hidden terminal problem [37]. Unfortunately, this issue cannot be addressed by simply using a standard RTS/CTS-like channel reservation approach (see [25] for details).

In the second class of TPC schemes, power control is applied on a per-packet basis, with the TP being dependent on both the transmitting and receiving terminals. The TP in this case is not directly tied to the routing layer or the topological properties of the network (although some schemes in this class indirectly influence the decisions taken by the routing layer). For a given next hop that is provided by the routing layer, the main question here is what TP to use for sending a given data packet to that next hop. This class of TPC schemes can be further divided into two subclasses: energy- and throughput-oriented schemes. The former subclass (e.g., [15], [19], [20], [30]) aims primarily at reducing energy consumption, with network throughput being a secondary factor. Terminals exchange their RTS and CTS packets at a maximum power (P_{max}), but send their data and Ack packets at the minimum power needed for reliable communication (P_{min}). The value of P_{min} is determined based on the required QoS (i.e., the signal-to-interference-plus-noise ratio (SINR)), the interference level at the receiver, and the channel gain between the transmitter and the receiver. In [19] the authors enhanced the performance of this approach by periodically increasing the TP of the data packet to P_{max} for enough time to protect the reception of the Ack at the source terminal. While this class of TPC protocols achieves good reduction in energy consumption (relative to the 802.11 MAC protocol), at best it gives comparable throughput to that of the 802.11 scheme. The main reason is that, as in the 802.11 approach, RTS and CTS messages are used to *silence* neighboring terminals, preventing concurrent transmissions from taking place over the maximum transmission range[2].

Throughput-oriented TPC schemes (e.g., [23], [24], [40]) use per-packet power control to increase the channel spatial reuse. These schemes allow for concurrent transmissions in the same vicinity of a receiver by locally broadcasting *collision avoidance information* (CAI) over a separate control channel. In the PCMA protocol [23], the receiver advertises its *interference margin*[3] by sending busytone pulses over a separate control channel. The use of a separate control channel in conjunction with a busytone scheme was proposed in [40], where the sender transmits data packets and busytones at reduced power, while the receiver transmits its busytones at the maximum power. The PCDC protocol [24] uses two frequency-separated channels for data and control. RTS and CTS packets are transmitted over the control channel, providing CAI that facilitates interference-limited concurrent transmissions in the same vicinity.

Although the simulations in [23], [24], [40] indicate impressive improvement in throughput over the 802.11 scheme, we see five major design problems with these schemes that make their practicality questionable:

- In [23], [24], [40], the channel gain is assumed to be the same for both the control (or busytone) and data channels, and that terminals are able to transmit on one channel and, simultaneously, receive on the other. It is very difficult to achieve these two assumptions simultaneously (see [25] for details).
- To be able to receive/transmit and simultaneously receive/transmit over two channels, the mobile terminal must

[2] The maximum transmission range of terminal i is the largest region around i over which i's maximum power transmission can be successfully received in the absence of interference from other terminals.

[3] The interference margin of a receiver is the amount of *additional* interference that the receiver can tolerate without violating its SINR requirement.

be equipped with two transceivers. The complexity and cost of the additional hardware may not justify the increase in throughput. Furthermore, it is unfair to compare the performance of these protocols to the single-channel, single-transceiver IEEE 802.11 scheme.

- Currently, most wireless devices implement the IEEE 802.11b standard. The class of two-channel protocols is not backward-compatible with the IEEE 802.11 standard, which makes it difficult to deploy such protocols in real networks.
- The above schemes do not provide reliability, i.e., they do not protect the reception of the Ack packet.
- Finally, the optimal allocation of the total spectrum between the data and control channels is load dependent. So for the allocation to be optimal under a varying traffic load, it has to be adjusted adaptively, which is not feasible in practice.

Before closing, we mention few other schemes in the literature that tackle the problem of power control from a completely different perspective. The COMPOW protocol [27] relies on routing-layer agents to converge to a *common* power level for all network terminals. However, for constantly moving terminals, the scheme (like any other routing-protocol-based scheme) incurs significant overhead, and convergence to a common power may not be possible. Moreover, in situations where network density varies widely (i.e., terminals are clustered), restricting all terminals to converge to a common power is a conservative approach. A clustering approach was proposed in [22], which simplifies the forwarding function for most terminals but at the expense of reducing network utilization (since all communications have to go through an elected terminal). This can also lead to the creation of bottlenecks.

A joint clustering/TPC protocol was proposed in [21], where clustering is implicit and is based on TP levels rather than on addresses or geographical locations. The routing overhead in this protocol grows in proportion to the number of routing agents, and can be significant even for simple mobility patterns (note that for the DSR routing protocol, for example, routing packets account for approximately 38% of the total received bytes [18]). The protocol in [6] is energy-oriented and is basically a mechanism to learn the minimum TP level required for a terminal to successfully transmit to a neighboring terminal. This approach, however, suffers from the hidden terminal problem (see [24] for more details). Another novel approach for TPC is based on joint scheduling and power control [12]. This approach requires a central controller to execute the scheduling algorithm, i.e., it is not a truly distributed solution. The Medium Access via Collision Avoidance with Enhanced Parallelism (MACA-P) proposed in [5] allows for parallel transmissions in situations only when two neighboring nodes are either both receivers or both transmitters, but a receiver and a transmitter are not neighbors. In addition, TPC was not considered in that work.

III. The POWMAC Protocol

A. Assumptions

In designing POWMAC, we assume that the channel gain is stationary for the duration of a few control and one data packet transmission periods. As discussed in Section III-K, this assumption holds for typical mobility patterns and transmission rates. We also assume that the gain between two terminals is the same in both directions. This is the underlying assumption in any RTS/CTS-based protocol, including the IEEE 802.11 scheme. Finally, we assume that the radio interface can provide the MAC layer with the average power of a received control signal as well as the average interference power. Off-the-shelf wireless cards (e.g., [4]) readily provide such measured values using SINR estimators like the ones discussed in [29]. In POWMAC, each terminal is equipped with one transceiver that has standard carrier-sense hardware (i.e., a basic IEEE 802.11-compliant transceiver).

B. Overview of POWMAC

POWMAC is distributed, asynchronous, and adaptive to channel changes. Its key features are as follows. First, unlike the IEEE 802.11 approach (and the schemes in [6], [15], [19], [20], [30]), POWMAC does not use the control packets (i.e., RTS/CTS) to silence neighboring terminals. Instead, CAI is inserted in the control packets and is used in conjunction with the received signal strength of these packets to dynamically *bound* the TP of potentially interfering terminals in the vicinity of a receiving terminal. The details of this mechanism are presented in Section III-D. The second main feature of POWMAC is that the required TP of a data packet is computed at the packet's intended receiver, say terminal i, according to a predetermined *maximum load factor*. The rationale behind this approach is to allow for some interference tolerance at receiver i, so that multiple interference-limited transmissions can simultaneously take place in the neighborhood of i. The tradeoffs involved in determining this load factor are discussed in Section III-C.

The third feature of POWMAC is that some control packets (CTS packets and newly defined Decide-to-Send (DTS) packets) are transmitted at an adjustable power level so that they reach all and only potentially interfering terminals. This improves the spatial reuse for the control packets themselves and reduces their collisions. Section III-H presents the details of this aspect of power control.

Finally, in POWMAC, after terminals exchange their control packets, they refrain from transmitting their data packets for a certain duration, referred to as the *access window* (AW). The AW allows several pairs of neighboring terminals to exchange their control packets such that (interfering) data transmissions can proceed simultaneously as long as collisions are prevented. The AW consists of an adjustable number of fixed-duration *access slots*. As explained later, this number is adaptively varied, depending on network load. The AW is needed for two reasons. First, it reduces the likelihood of collisions between control and data packets. Even when power controlled, control packets will, in general, be transmitted at a higher power than data packets, so that they can reach many potential interferers. So allowing these control packets to overlap in time with data packets (to enable concurrent RTS/CTS-based transmissions in the same neighborhood) would increase the likelihood of collisions. We remedy this situation by using an AW, whereby a receiving terminal i allows its neighbors to exchange their RTS/CTS packets before i's data reception starts, and when possible, to have these neighbors' own data transmissions proceed simultaneously with i's reception. Note that data packets are transmitted at a reduced power level to reach only the intended receiver, and so multiple data packets can be transmitted concurrently and still be received correctly.

The second purpose of the AW is to inform terminals that are currently transmitting or receiving of the ensuing data transmission. Because POWMAC uses a single-channel architecture, terminals can either transmit or receive at a given time, but not both. As a result, a terminal, say i, is basically "deaf" while transmitting, so it cannot hear any transmitted control packets in its vicinity. Consequently, when i becomes idle, its information about the ongoing receptions in its vicinity can be outdated, which can lead to collisions (if i decides to transmit again). The protocols in [23], [24], [40] alleviate this problem by using a two-channel, two-transceiver architecture; terminals are able to transmit/receive their data packets and still hear the control signals. However, as we discussed in Section II, these approaches are not desirable for several reasons.

We note here that allowing several RTS/CTS exchanges to take place prior to data-packet transmissions was also used in the MACA-P protocol [5]. However, in that work the objective was not to address TPC, but rather to prevent collisions between control and data packets.

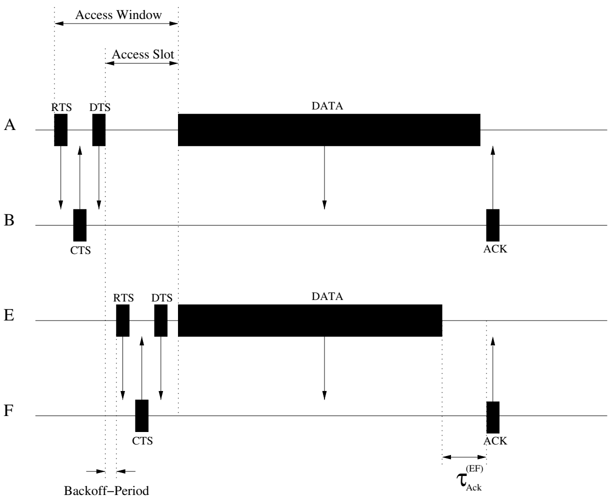


Fig. 2. Basic operation of POWMAC.

We conclude this section with an example that illustrates the basic operation of POWMAC (see Figure 2). The network topology is the one shown in Figure 1. Terminal A transmits an RTS to B at a maximum (known) power (P_{max}). Terminal B replies back with a CTS packet that is sent at an adjustable power level to reach all and only potentially interfering terminals. The RTS/CTS exchange allows terminals A and B to agree on the TP of the ensuing data packet. It also provides a way to inform potentially interfering terminals (e.g., terminal E) of the power that they can use without disturbing the scheduled reception of the data packet at B. Terminal A confirms that the transmission $A \rightarrow B$ can proceed using the newly defined DTS control packet. Besides other reasons mentioned in Section III-D, the DTS packet is used to inform A's neighbors of the power level that A intends to use for its data transmission. As explained later, this information is needed so that A's neighbors (i.e., terminal D) can determine whether or not they can receive a data packet from some other terminal (e.g., C) simultaneously while A is transmitting to B. In addition, the DTS provides a way to inform potentially interfering terminals (e.g., terminal D) of the power that they can use without disturbing the reception of the Ack packet at A. After the RTS/CTS/DTS exchange, terminal A refrains from sending its data packet for the remaining of the AW duration. During this duration, E and F can exchange control packets and decide if they can start the transmission $E \rightarrow F$ depending on whether or not this transmission will disturb the scheduled transmission $A \rightarrow B$.

C. Load Control

Load control is a concept that allows a prospective receiver to determine the appropriate TP for its upcoming data reception and the impact of this TP on ongoing as well as scheduled receptions of both data and Ack packets. If the power used to transmit a data packet to a terminal, say i, is just enough to overcome the current interference at i, then none of i's neighbors should be allowed to start new transmissions during i's reception. This silencing of neighboring terminals negatively impacts the aggregate throughput. On the other hand, if the TP is too high, it may induce high interference on other terminals in the vicinity of the transmitter, preventing them from receiving.

The load factor at terminal i, denoted by $\xi^{(i)}$, is a measure of the activity in terminal i's neighborhood. Formally, it is defined as[4]:

$$\xi^{(i)} \overset{\text{def}}{=} \frac{P_{\text{thermal}} + P_{\text{MAI}}^{(i)}}{P_{\text{thermal}}}, \tag{1}$$

where $P_{\text{MAI}}^{(i)}$ is the current multi-access interference (MAI) at receiver i[5]. Now, consider the transmission of a packet from j to i. Let d_{ij} be the distance between i and j, and let μ^* be the SINR threshold required to achieve a target bit error rate (BER) at receiver i. We assume that the TP attenuates with d_{ij} as k/d_{ij}^n, where k is a constant and $n \geq 2$ is the loss factor. Then, the minimum TP that is needed to achieve the target BER is

$$P_{\text{min}}^{(ji)} = \frac{\mu^*(P_{\text{thermal}} + P_{\text{MAI}}^{(i)})}{G_{ji}} = \frac{\mu^* \xi^{(i)} P_{\text{thermal}}}{G_{ji}} = \frac{\mu^* \xi^{(i)} P_{\text{thermal}} d_{ij}^n}{k}, \tag{2}$$

where $G_{ji} = k/d_{ij}^n$ is the channel gain from terminal j to terminal i ($G_{ji} \ll 1$). While more capacity can be achieved by increasing $\xi^{(i)}$ (i.e., allowing larger $P_{\text{MAI}}^{(i)}$), this also increases the power needed to transmit the packet, which in turn increases energy consumption. Energy is a scarce resource in MANETs, so it is undesirable to trade it off for throughput. Moreover, the Federal Communications Commission (FCC) regulations put a limit on the maximum power that can be used by terminals in the 2.4 GHz spectrum (e.g., 1 Watt for 802.11 devices). Given this limit, as the load is increased, the channel gain must be increased (with μ^* and P_{thermal} held constant), and so the maximum range (or coverage) for reliable communication will decrease.

Collectively, the above factors necessitate *load planning*, i.e., imposing a maximum load factor (MLF), denoted by ξ_{max}, that terminals are not allowed to exceed. This ξ_{max} is set at the design phase to reflect several goals, including throughput, network lifetime, etc. One possible choice is as follows. First, to increase the spatial channel reuse, terminal j uses a TP that results in the MLF at terminal i. This TP is given by (see Equation 2):

$$P_{\text{POWMAC}}^{(ji)} = \frac{\mu^* \xi_{\text{max}} P_{\text{thermal}}}{G_{ji}} = \frac{\mu^* \xi_{\text{max}} P_{\text{thermal}} d_{ij}^n}{k}. \tag{3}$$

Second, we require that the (interference-free) maximum transmission range for both POWMAC and the 802.11 scheme, denoted by d_{max}, to be the same. Then, assuming that d_{ij} is uniformly distributed between zero and d_{max} (other distance distributions, which could depend on the routing protocol, may also be used), we have

$$E[P_{\text{POWMAC}}^{(ji)}] = \frac{\mu^* \ \xi_{\text{max}} \ P_{\text{thermal}} \ d_{\text{max}}^n}{k \ (n+1)}. \tag{4}$$

[4] This definition is somewhat similar but not quite identical to the definition used in [28] for cellular systems.

[5] Traditionally, MAI has been used to refer to the interference between signals that are spread using different CDMA codes. Since terminals in the IEEE 802.11 scheme use the same spreading code, in this paper the term MAI will be used to refer to interference from unintended signals that are spread using the same code.

As for the 802.11 protocol, its corresponding TP is:

$$P_{802.11} = \frac{\mu^* \, P_{\text{thermal}} \, d_{\max}^n}{k}. \tag{5}$$

Note that $P_{802.11}$ does not depend on d_{ij} since the 802.11 scheme uses a fixed TP. To account for the energy-consumption factor, we require that $\xi_{\max}$ be chosen such that the two protocols consume the same average energy per bit. Equating (4) and (5), we end up with $\xi_{\max} = n + 1$. As an example, consider the two-ray propagation model with $n = 4$. Then $\xi_{\max} = 7$ dB, which lies within the range of values used in already deployed cellular systems [28]. Finally, we require that the *maximum* TP used in POWMAC be constrained by the FCC limit (from (3) and (5), this maximum power is given by $\xi_{\max} P_{802.11}$).

D. Channel Access Mechanism

Given a predetermined MLF, the purpose of the channel access mechanism is to allow the source and the sink to agree on the required TP such that the MLF is not exceeded at the source (Ack recipient) and at the sink (data recipient) during the reception periods. The access protocol should also ensure that the ensuing data transmission does not disturb any of the scheduled data/Ack receptions in the vicinities of the source and sink terminals. We now describe the details of the POWMAC access mechanism. In contrast to cellular systems where the base station makes the admission decision, in our case each terminal decides whether its transmission can proceed or not, depending on previously heard RTS, CTS, and DTS packets.

Each terminal i maintains a Power Constrained List denoted by PCL(i). This list is an extension of the Network Allocation Vector (NAV) used in the IEEE 802.11 scheme. Basically, PCL(i) encodes i's knowledge about other *active* terminals, i.e., terminals that are receiving, transmitting, or scheduled to do either function in i's vicinity. For every active terminal u in i's vicinity, PCL(i) contains the following entries (as explained shortly, these entries are computed using some information advertised by terminal u in its CTS or DTS control packets, and by measuring the signal strength of these control packets):

- The address of terminal u.
- The channel gain between terminals i and u (G_{iu}), computed using the received signal strength of u's control packet.
- The start time and duration of u's activities (data-reception/Ack-transmission or data-transmission/Ack-reception), as advertised by terminal u in its CTS or DTS packet.
- The maximum tolerable interference (MTI) of terminal u, denoted by $P_{\text{MTI}}^{(u)}$ during u's data or Ack reception. This is the maximum additional interference that terminal u can tolerate from an interfering terminal such as terminal i. As will be explained shortly, this information is advertised by terminal u.
- The TP that terminal u will use during its scheduled data or Ack transmission, advertised in terminal u's CTS or DTS packet.

Let $\pi_i(u)$ be the maximum TP that terminal i can use without disturbing u's reception. Using G_{iu} and $P_{\text{MTI}}^{(u)}$, terminal i computes $\pi_i(u)$ as:

$$\pi_i(u) = \min \left\{ \frac{P_{\text{MTI}}^{(u)}}{G_{iu}} \; , \; \xi_{\max} P_{\max} \right\}. \tag{6}$$

Let $\Psi(i)$ be the set of terminals in i's vicinity whose receptions overlap with i's transmission ($\Psi(i) \subset \text{PCL}(i)$). Then the maximum allowable TP that terminal i can use without disturbing any of its neighbors, denoted by $P_{\text{MAP}}(i)$, is given by:

$$P_{\text{MAP}}(i) = \min_{u \in \Psi(i)} \{\pi_i(u)\}. \tag{7}$$

Depending on the order in which terminals initiate their RTS messages in a given AW, we classify them into *master* and *slave* terminals. Terminal j is a master if it has a packet to send, its PCL is empty, and it does not sense any carrier signal. In this case, j's RTS packet announces the start of an AW (the size of this AW is also set by terminal j). On the other hand, a terminal, say k, is a slave terminal if it is in the vicinity of a master terminal, say j. In this case, terminal k may send an RTS message in any, but not the first, slot of the AW initiated by terminal j. Clearly, the master-slave designation is time-varying. We now explain the access rules for both master and slave terminals.

D.1 Master Terminals

Consider a master terminal, say j, that has a data packet to transmit to another terminal, say i. If j does not sense a carrier (for a random duration of time), it sends an RTS message at $P_{\max}$, and includes in this packet the values of $P_{\text{MAP}}^{(j)}$ and $N_{AW}^{(j)}$; the remaining number of slots in j's AW (how terminal j determines $N_{AW}^{(j)}$ will be explained shortly).

Upon receiving the RTS packet, receiver i uses the predetermined $P_{\max}$ value and the power of the received signal to estimate the channel gain G_{ji} between terminals j and i (note that we assume channel reciprocity, and so $G_{ij} = G_{ji}$) . The minimum TP that is needed so that i can decode the packet was given in (2). In that equation, $P_{\text{MAI}}^{(i)}$ represents the total MAI from *already ongoing* interfering transmissions, and it does not account for any interference tolerance[6]. Now, according to the load planning calculations in Section III-C, the power that terminal j is allowed to use to send to i was given by $P_{\text{POWMAC}}^{(ji)}$ in (3). If $P_{\text{POWMAC}}^{(ji)} < P_{\min}^{(ji)}$ (i.e., $\xi^{(i)} >$ MLF), then the MAI in the vicinity of terminal i is greater than the one allowed by the planned loading. In this case, i responds with a negative CTS, informing j that it cannot proceed with its transmission (the negative CTS is used to prevent multiple RTS retransmissions from j). The philosophy behind this design is to prevent transmissions from taking place over links that perceive high MAI. This consequently increases the number of active links in the network, subject to the available power constraints, and limits the energy consumed in the $j \rightarrow i$ communication.

On the other hand, if $P_{\text{POWMAC}}^{(ji)} > P_{\min}^{(ji)}$, then it is possible for i to receive j's signal. In that case, i calculates the *maximum additional interference power* ($P_{\text{MAI-add}}^{(i)}$) that it can endure from future unintended transmitters so that the SINR at i does not drop below μ^*. This $P_{\text{MAI-add}}^{(i)}$ is given by:

$$P_{\text{MAI-add}}^{(i)} = \frac{G_{ji}}{\mu^*}(P_{\text{POWMAC}}^{(ji)} - P_{\min}^{(ji)}) = \left(\xi_{\max} - \xi^{(i)}\right) P_{\text{thermal}}. \tag{8}$$

The next step is to equitably distribute $P_{\text{MAI-add}}^{(i)}$ among future potential interferers in the vicinity of i. The rational behind this distribution is to prevent one neighbor from consuming the entire $P_{\text{MAI-add}}^{(i)}$. In other words, we think of $P_{\text{MAI-add}}^{(i)}$ as a network resource that should be shared among various neighboring terminals. Recall that j's RTS contains $N_{AW}^{(j)}$; the remaining number of access slots in the current AW. Obviously, the number of concurrent transmissions should not exceed $N_{AW}^{(j)}$. Thus, terminal j uses $N_{AW}^{(j)}$ as the number of future potential interferers in its neighborhood.

Future interference at terminal i comes from interferers within the maximum range of i and interferers outside that range. The interference margin $P_{\text{MAI-add}}^{(i)}$ has to account for both types of interferers (if $P_{\text{MAI-add}}^{(i)}$ is distributed among within-range interferers only, an increase in the interference from outside the range of i could cause at packet collision at i). Let $P_{\text{MAI-within}}^{(i)}$ and $P_{\text{MAI-other}}^{(i)}$ be the two components of $P_{\text{MAI-add}}^{(i)}$. While terminal i can predict the number of within-range interferers, it cannot do the same for outside-range interferers. To estimate $P_{\text{MAI-other}}^{(i)}$, we follow a similar approach to the one used in cellular networks for an analogous problem. In cellular networks, the base station has control over in-cell interference (using open- and closed-loop power control), but it cannot influence out-of-cell interference. This problem has been thoroughly investigated in [35], and a practical (widely adopted) solution for it is to assume that the out-of-cell interference is a certain fraction of the in-cell interference. Considering the similarity between the role of a receiver in a power-controlled MAC protocol for MANETs and the role of a base station in cellular systems, we let $P_{\text{MAI-other}}^{(i)} = \alpha P_{\text{MAI-within}}^{(i)}$, where $\alpha \approx 0.5$ for the two-ray propagation model and uniformly distributed terminals. A simple weighting factor can be used to account for other distributions [35].

Based on the above, the maximum tolerable interference $P_{\text{MTI}}^{(i)}$ that a *single* future interferer can add to terminal i is set to:

$$P_{\text{MTI}}^{(i)} = \frac{P_{\text{MAI-add}}^{(i)}}{(1 + \alpha) N_{AW}^{(j)}}. \tag{9}$$

[6] In [31] the authors derived a finite value for the interference range in the case of minimum TP. However, the thermal noise power was not taken into account in that derivation.

When responding to j's RTS, terminal i indicates in its CTS the power level $P^{(ji)}_{\text{POWMAC}}$ that j must use for the data transmission. In addition, terminal i inserts $P^{(i)}_{\text{MTI}}$ in the CTS message to inform its neighbors of the maximum power they can use such that i's reception is not disturbed. The CTS is sent at an adjustable power ($P^{(i)}_{CTS}$) whose value is included in the CTS packet, as explained in Section III-H.

Upon receiving i's CTS, terminal j replies back with a DTS packet that includes the value of $P^{(ji)}_{\text{POWMAC}}$. The DTS is needed to inform j's neighbors that may have not heard i's CTS about $P^{(ji)}_{\text{POWMAC}}$. Using $P^{(ji)}_{\text{POWMAC}}$ and the channel gain information, j's neighbors can compute the amount of *expected* MAI due to the scheduled transmission $j \rightarrow i$. The total expected MAI due to scheduled transmissions in the neighborhood of a terminal, say u, allows u to determine if it can receive a packet (data or Ack) following the current AW. If this MAI exceeds $\xi_{\max} P_{\text{thermal}}$, then u is expected to perceive high MAI, and therefore, should refrain from scheduling a reception; otherwise, u is free to receive a packet.

Similar to the CTS packet, the DTS packet contains the amount of additional interference $P^{(j)}_{\text{MTI}}$ that node j can tolerate during its Ack reception. As in [9], the DTS packet in POWMAC also provides a mechanism to announce the success of the RTS/CTS exchange between j and i to those neighbors of j who have not heard i's CTS. The IEEE 802.11 scheme uses carrier sensing for this purpose; if the neighbors of j do not sense a carrier after hearing the RTS for some time, they assume that the RTS/CTS exchange was not successful. This same mechanism, however, cannot be used in POWMAC since the data packet is transmitted at a power less than the RTS power, and thus the carrier sense range of the data packet is much smaller than that of the RTS (or CTS) packet. The DTS is also sent at an adjustable power as explained in Section III-H.

Once the RTS/CTS/DTS exchange is completed, no further negotiations are made for the corresponding data/Ack transmission. This makes TPC schemes in MANETs fundamentally different from their cellular counterparts. In cellular systems, every time a new session is started or terminated, the powers of ongoing transmissions are renegotiated. In contrast, power in MANETs is allocated only once at the start of the session, i.e., the whole data packet is transmitted at one power level, regardless of what follows the start of that packet transmission. The cellular approach requires that the entire state of the system (power used by every terminal in the network) be known whenever a new session is to be admitted, which cannot be achieved in a distributed MANET.

D.2 Slave Terminals

Slave terminals are terminals that are within the transmission range of a master terminal. In addition to the computations that master terminals perform (e.g., computing $P^{(.)}_{\text{POWMAC}}$, $P^{(.)}_{\text{min}}$, etc.), there are two "feasibility conditions" (FCs) that each slave terminal, say k, must fulfill for its activity (transmission or reception) to proceed simultaneously with each scheduled activity in k's vicinity. The FCs are:

- FC$_1$ *(Effect of terminal k's transmission on the receptions in k's neighborhood)*: Terminal k's data or Ack transmission should not disturb already scheduled receptions in k's vicinity.
- FC$_2$ *(Effect of k's neighbors' transmissions on k's reception)*: The additional interference due to already scheduled transmissions should not increase the load factor at terminal k above $\xi_{\max}$ during terminal k's data or Ack reception.

The two FCs must be satisfied with respect to *all* scheduled activities in k's vicinity that are known to terminal k. As will become clear shortly, the chances for terminals to fulfill their FCs can be improved by allowing pairs of communicating terminals to move forward the transmission times of their Ack packets. In other words, POWMAC allows for a delay lag between the reception of a data packet and the transmission of its corresponding Ack packet. Thus, a recipient, say i, of a data packet may wait for a certain period, denoted by $\tau^{(ji)}_{Ack}$ (see Figure 2), before sending the Ack to terminal j. This lag allows i to avoid overlapping its Ack transmission (reception) with other data or Ack receptions (transmissions) in i's or j's vicinities. $\tau^{(ji)}_{Ack}$ is communicated using a 1-byte field in the RTS/CTS/DTS packets. Note that it is not useful to change the transmission time of a data packet to avoid overlapping data packets since the main goal of POWMAC is for data packets to proceed simultaneously.

Delaying the transmission time of an Ack packet must be carefully coordinated between the the source and sink terminals; otherwise, conflicts may arise and may result in collisions. For example, the source may choose to delay the Ack by Δ_1 seconds, and later on the sink terminal chooses to delay the same Ack packet by $\Delta_2 < \Delta_1$ seconds, thus violating the source's FCs. Another issue is how to compute $\tau^{(ji)}_{Ack}$ when there are multiple scheduled activities in terminal j's neighborhood (each activity calls for a different value of $\tau^{(ji)}_{Ack}$). To address these issues, we establish two "viability rules" (VRs) for changing the Ack transmission time:

- VR$_1$: Each terminal that wishes to fulfill its FCs (with respect to a certain neighboring activity) is allowed to increase the present value of $\tau^{(.)}_{Ack}$, but not decrease it.
- VR$_2$: Each terminal computes $\tau^{(.)}_{Ack}$ that fulfills its FCs with respect to a given neighboring activity in such a way that if $\tau^{(.)}_{Ack}$ is later *increased* by the same terminal to accommodate another neighboring activity or is increased by the communication peer, then that terminal's FCs are not violated. An example that explains this rule will be given shortly.

The significance of the VRs is that they allow each terminal to *independently* consider its interaction with its active neighbors (i.e., fulfill its FCs by choosing an appropriate $\tau^{(.)}_{Ack}$) on a *per-terminal* basis. To illustrate, consider Figure 3, where four terminals are in the same vicinity, i.e., control packets of any terminal are heard by the other three terminals. Terminal j has already scheduled a data packet transmission to terminal i. Terminal v wishes to schedule a transmission to terminal n simultaneously with the transmission $j \longrightarrow i$. The VRs allow terminal v to evaluate its future interaction with terminal j and accordingly choose a value for $\tau^{(vn)}_{Ack}$, and independently to consider its interaction with terminal i and accordingly choose a possibly different value for $\tau^{(vn)}_{Ack}$. Furthermore, the VRs also allow the receiving terminal n to independently change the value of $\tau^{(vn)}_{Ack}$ to fulfill its own FCs without worrying that this new value could affect the FCs at terminal j. To demonstrate how terminals operate to fulfill their FCs, we examine the two scenarios shown in Figures 4 and 7 (other possible scenarios are described in [25]). In these scenarios, terminals j and i have just completed an RTS/CTS/DTS exchange. The (slave) terminal v has a data packet that it wishes to transmit to terminal n. We now examine what terminal v has to do in each scenario to fulfill the FCs.

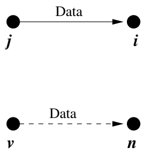


Fig. 3. Example of a network topology where POWMAC allows for two simultaneous transmissions in the same vicinity.

1) Source-Source Interaction:

The first scenario represents source-source interaction. An example of this scenario is shown in Figure 4. Here, source terminal v can potentially interfere with source terminal j, and vise versa. After v hears j's RTS and DTS messages, it uses the signal strength of the received RTS message and the value of the RTS transmission power ($P_{\max}$) to estimate the channel gain G_{vj}. The channel gain and the value of $P^{(j)}_{\text{MTI}}$ (included in the DTS message) are used to update the maximum power $P^{(v)}_{\text{MAP}}$ that v can use in its future transmissions, according to (7), *during j's Ack reception*. Terminal v also records the transmission times and the TP of the $j \rightarrow i$ data/Ack packets (recall that $P^{(ji)}_{\text{POWMAC}}$ is used for both data and Ack). This information is part of the DTS; the exact format of the control packets will be given later.

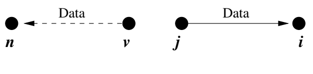


Fig. 4. Scenario that describes a source-source interaction.

In order for terminal v to fulfill its FCs, it compares its data packet length with j's data packet length. Note that terminals that contend in the same AW schedule their data transmissions to start at the same time but may complete them at different times. If v's data packet is shorter than j's data packet (see Figures 5), and the additional interference due to j's data transmission (i.e., $P^{(ji)}_{\text{POWMAC}}G_{vj}$) would *not* increase the load factor at terminal v beyond $\xi_{\max}$ during terminal v's Ack reception, then v does not do any more computations. Else, v delays the Ack transmission time until j finishes its data transmission, i.e., the Ack packet is moved from Position 1 to Position 2 in that figures. This way, terminal v satisfies FC$_2$, while FC$_1$ is also satisfied (with respect to the interaction $v \leftrightarrow j$) since v's transmission does not overlap with j's reception.

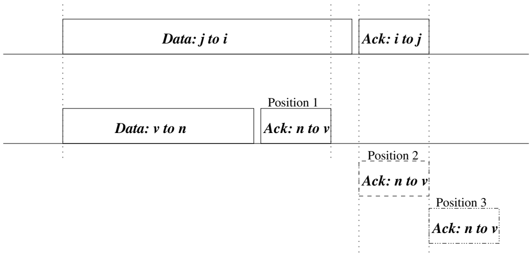


Fig. 5. Slave terminal's Ack packet transmission completes before master terminal's Ack transmission starts.

In case v's data packet is equal to j's data packet, then v does not do any more computations. If v's data packet is longer than j's data packet (see Figures 6), then the maximum TP used by v for its data transmission must not exceed the new value of $P^{(v)}_{\text{MAP}}$ updated from j's DTS message. Terminal v cannot decide in advance how much TP the communication $v \rightarrow n$ requires. Therefore, v includes the value of $P^{(v)}_{\text{MAP}}$ in its RTS message and leaves the decision of the TP determination to receiver n. This way, terminal v satisfies FC$_1$, while FC$_2$ is also satisfied (with respect to the interaction $v \leftrightarrow j$) since v's reception does not overlap with j's transmission. Note that both VR$_1$ and VR$_2$ are satisfied in all the above cases when considering the interaction $v \leftrightarrow j$.

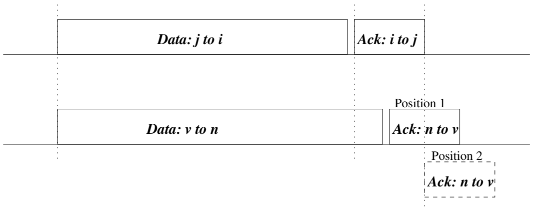


Fig. 6. Another case where slave terminal's Ack packet transmission overlaps with master terminal's Ack packet transmission.

2) Source-Sink Interaction:

The second scenario is shown in Figure 7, where the source terminal v can potentially interfere with an already scheduled reception at sink i, and vise versa. When v hears i's CTS, it uses the signal strength of the received message and the value of the CTS transmission power (included in the CTS) to estimate the channel gain G_{vi} between itself and terminal i. The channel gain and the broadcasted $P^{(i)}_{\text{MTI}}$ value are used to update the maximum power $P^{(v)}_{\text{MAP}}$ that v can use in its future transmissions, according to (7). Terminal v also records the transmission times and the TP $P^{(ji)}_{\text{POWMAC}}$ of the $j \rightarrow i$ data/Ack packets.

Data
j
i
Data
v
n

Fig. 7. Scenario that describes a source-sink interaction.

In order for v to fulfill FC$_1$, its maximum TP must not exceed the new value of $P^{(v)}_{\text{MAP}}$. Terminal v cannot decide in advance how much TP the communication $v \rightarrow n$ requires. Therefore, v includes the value of $P^{(v)}_{\text{MAP}}$ in its RTS message and leaves the decision to the receiver n.

Now, in order for v to fulfill FC$_2$, it checks whether the additional interference due to i's Ack transmission (i.e., $P^{(ji)}_{\text{POWMAC}}G_{vi}$) would increase the load factor at terminal v beyond $\xi_{\max}$. If it would not, then v does not do any more computations; else, v checks if there is an overlap between its Ack reception and i's Ack transmission. There are three possibilities to consider:

- If there is no overlap and v's Ack reception starts *after* i finishes its Ack transmission, then v does not perform any more computation to satisfy FC$_2$ with respect to the $v \leftrightarrow i$ interaction.
- If there is an overlap (see Figure 6), then terminal v delays the Ack until i finishes its Ack transmission, i.e., the Ack packet is moved from Position 1 to Position 2 in Figure 6. This way, terminal v satisfies FC$_2$.
- The last case is the one shown in Figure 5 where there is no overlap and v's Ack reception finishes *before* i starts its Ack transmission. This case requires special attention. Recall that to increase the chances for terminals to fulfill their FCs, we allow pairs of communicating terminals to move forward the transmission times of their Ack packets. This means that the receiver, terminal n in this case, may actually delay the Ack transmission time to fulfill its own FCs, which could violate v's FCs (for example, if terminal n delays the Ack transmission time such that the new schedule results in an overlap between v's Ack reception and i's Ack transmission). Therefore, terminal v delays the Ack reception time until i finishes its Ack transmission, i.e., the Ack packet is moved from Position 1 to Position 3 in Figure 5. This example shows the importance of VR$_2$.

E. Contention Resolution

For contention resolution, we follow the work in [26], which, unlike the IEEE 802.11 scheme, performs contention resolution in the persistent domain instead of the backoff domain. As shown in [26], if the access probability (x_r) of terminal r is adapted according to

$$\dot{x_r} = \alpha - \beta p_r x_r \ , \tag{10}$$

where α and β are system parameters, and p_r is the loss probability experienced by terminal r, then the system converges to an optimal point that maximizes the network throughput under a proportional fairness model.

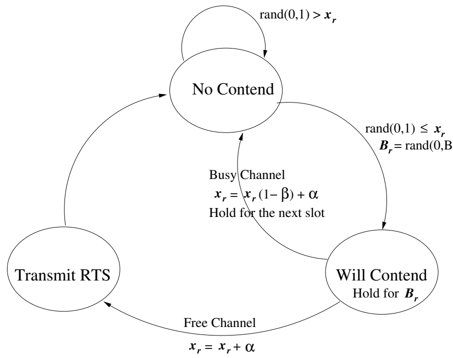


Fig. 8. State diagram of the contention resolution algorithm used in the POWMAC protocol.

If a terminal, say r, wants to transmit a data packet, it first verifies that its FCs are satisfied. If so, then with probability x_r, r contends for the channel in the next access slot of j's AW (j is a neighboring master terminal). If successful, terminal r chooses a wait time B_r that is uniformly distributed in the interval [0,B]; B is a system-wide backoff counter. After this waiting time, terminal r senses the channel. If the channel is free, terminal r transmits its RTS in the current access slot. Note that B is in the order of few microseconds while a time slot is in milliseconds, so the backoff mainly serves to prevent synchronized RTS attempts. Figure 8 shows the state diagram of the contention resolution algorithm. Note that x_r is increased by α at the end of each access slot, but decreased by β only when the contention is not successful (i.e., with probability p_r). Hence, (10) is satisfied. Note also that when using this mechanism for POWMAC, we do *not* require any synchronization. Basically, once terminal r receives j's RTS, it divides its time access into $N_{AW}^{(j)}$ slots of predetermined length, regardless of the absolute time at terminals r and j. This issue is explored further in the next section.

F. Synchronization of the Access Window

So far, we have assumed that terminals can synchronize with a neighboring master terminal. We now explain the mechanism underlaying this process. *Note that by synchronization, we do not mean that terminals have the same clock; rather, they can determine the boundaries of the AW slots.* Consider the scenario in Figure 9 where master terminal j has scheduled a data transmission to terminal i, and (slave) terminal v has synchronized with j's AW (as we will explain shortly) and has scheduled a data transmission to terminal n. Suppose now that terminal k wishes to transmit to terminal l. We now explain how k synchronizes with j's AW. Note that v is in i's but not j's vicinity, and likewise, k is in n's but not v's vicinity.

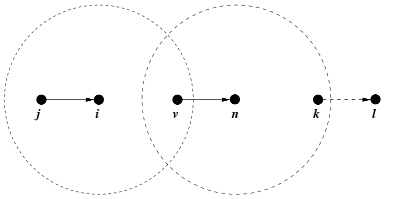


Fig. 9. Example that illustrates how slave terminals synchronize with the master terminal's AW. The two circles represent the maximum transmission ranges of terminals i and n.

First, we design the duration of the AW slot (AWS) to be fixed and common to all terminals. Specifically, an AWS consists of the sum of the transmission durations of the RTS, CTS, and DTS packets, the maximum backoff interval, plus two fixed short interframe spacing (SIFS) periods[7]. However, fixing the AWS duration is not enough for terminal k to synchronize with j's AW; the reason is that when v transmits its RTS message, it chooses a random wait time B_v that is uniformly distributed in the interval [0,B]. Since k hears only n's CTS, it is not possible for k to synchronize with j's AW. The situation is exemplified in Figure 10. The main problem is that k cannot determine the value of B_v, and so it cannot determine the end of that AWS. To remedy this situation, the value of B_v is announced in both the RTS and CTS control packets, allowing terminal k to synchronize with j's AW.

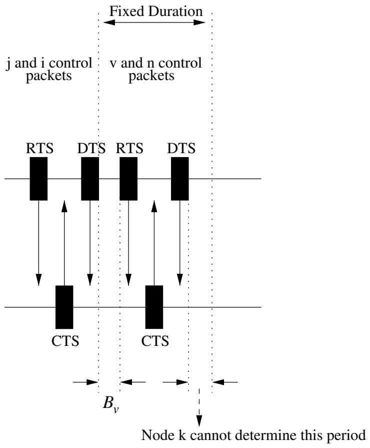


Fig. 10. Example that illustrates the challenge in synchronizing with the AW of a master terminal.

Finally, when the master terminal j sends its RTS message, it sets the value of B_j in the RTS message to the maximum backoff duration B. Thus, the following slot in the AW (i.e., the slot where v and n exchange their control messages) starts immediately after the reception of the j's DTS message, as shown in Figure 10.

G. Updating the AW Size

The AW size at a terminal, say j, is updated adaptively as a function of the load in the vicinity of j. The goal is to choose an AW size that maximizes the chances of concurrent data transmissions. To achieve that, terminal j examines two history values: the actual interference perceived by terminal j during its reception, and the number of concurrent data transmissions and receptions in j's vicinity.

At the end of the data reception at terminal j, if the actual interference perceived by terminal j is higher than a given fraction (e.g., 75%) of the planned interference $\xi_{\text{max}} P_{\text{thermal}}$, then the AW size need not be changed, since the allocated additional power to combat MAI was efficiently utilized to allow for concurrent transmissions.

On the other hand, if less than that threshold was used, then terminal j should adapt (either increase or decrease) the AW size so that the allocated power is not wasted. To this end, terminal j checks the number of concurrent transmissions that actually took place in the previous AW (based on the numbers of CTS and DTS packets). If this number is less than, say η%, of the AW size, then either the load is low or the value of the AW size is too big to the extent that $P_{\text{MTI}}^{(j)}$ is too small (see (9)), i.e., $P_{\text{MTI}}^{(j)}$ is not large enough to allow for other nearby terminals to transmit. In both of these cases, terminal j decreases its AW size. In contrast, if the number of concurrent transmissions that actually took place in that AW is greater than η% of the AW size, then there is room for increasing the number of concurrent transmissions in the vicinity of terminal j. Hence, the AW size is increased. In case of a data-packet collision, the AW size is kept constant. Note that a collision may happen if the control messages were not successfully heard by a neighboring station. Finally, to prevent unstable fluctuations in the AW values, the AW size is incremented or decremented in steps of 1.

H. Adaptive Reservation Mechanism

In the IEEE 802.11 scheme, the RTS and CTS packets are transmitted at a fixed power P_{max}. As discussed in Section I, this approach can be overly conservative. Recall that in POWMAC, a receiver, say i, sends a CTS packet that contains CAI, namely $P_{\text{MTI}}^{(i)}$, to bound the TP of potentially interfering neighbors. A terminal, say n, that hears this packet sets its $P_{\text{MAP}}^{(n)}$ according to (7). If $\xi_{\text{max}} P_{\text{max}}$[8] is less than $P_{\text{MTI}}^{(i)}/G_{ni}$, the CAI is actually irrelevant to terminal n, and the CTS packet has reached farther than necessary. In POWMAC, this issue is not harmful as in the IEEE 802.11 scheme, simply because control packets in POWMAC do not prevent neighbors from transmitting. Nonetheless, one way to further enhance the operation of POWMAC is to transmit control packets only to those terminals who can actually make use of the CAI. This has the added advantage of reduced contention among control packets, leading to an increase in the spatial reuse. POWMAC uses the following adaptive TP approach for the control packets.

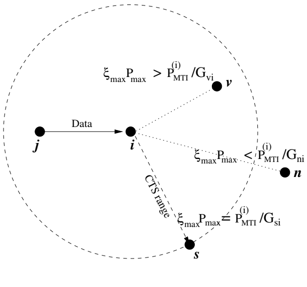

$\xi_{\text{max}} P_{\text{max}} > P_{\text{MTI}}^{(i)}/G_{vi}$
v
Data
j
i
$\xi_{\text{max}} P_{\text{max}} < P_{\text{MTI}}^{(i)}/G_{ni}$
n
CTS range
$\xi_{\text{max}} P_{\text{max}} = P_{\text{MTI}}^{(i)}/G_{si}$
s

Fig. 11. Range of the CTS message is limited to neighbors that can make use of the CAI conveyed in the CTS message.

The farthest neighbor from terminal i that can actually make use of the CAI contained in i's CTS (node s in Figure 11) is the one with channel gain of $P_{\text{MTI}}^{(i)}/(\xi_{\text{max}} P_{\text{max}})$. For any other terminal n that is more than G_{is} away from i, $\xi_{\text{max}} P_{\text{max}}$ is less than $P_{\text{MTI}}^{(i)}/G_{iv}$, and thus the CAI that is contained in i's CTS is irrelevant to terminal n. Accordingly, we set the range of the CTS of terminal i to $P_{\text{MTI}}^{(i)}/(\xi_{\text{max}} P_{\text{max}})$. Thus, the TP for the

[7] As defined in the IEEE 802.11b standard [2], a SIFS period consists of the processing delay for a received packet plus the turnaround time.

[8] Recall that the maximum TP in POWMAC is $\xi_{\text{max}} P_{\text{max}}$.

CTS packet of terminal i is:

$$P^{(i)}_{CTS} = \min\left\{\mu^* P_{\text{thermal}}\frac{\xi_{\max}P_{\max}}{P^{(i)}_{\text{MTI}}}\ ,\ \xi_{\max}P_{\max}\right\}, \tag{11}$$

where the minimum is taken because of the hardware constraints of the wireless interface. A similar computation is also applied to find the TP of the DTS packet at the transmitter. Note that the CTS (or DTS) packet may not be heard by all potential interferers (because of the hardware constraints of the wireless interface, i.e., the second term in the right hand side of (11) is less than the first). Such a limitation also exists in the IEEE 802.11 scheme, as it does not prevent nodes in the interference region from causing collisions with the data packet at the destination node (see [19] for details). Thus, this problem is not introduced by the proposed protocol. Moreover, POWMAC already takes into account future MAI due to terminals that do not hear the control packets by using $\alpha = 0.5$ in (9). Note also that in (11), we assume no interference at the CTS receiver. This is because in the design of wireless systems, the maximum range is typically calculated using only the thermal noise value [28], since there is no way of predicting all potential interferers beforehand.

Before concluding this section, we give the formats of the various control packets in POWMAC. For a source terminal j and a sink terminal i, the format of the RTS packet is:

$$\text{RTS}(j \to i) = \left\{j,\ i,\ P^{(j)}_{\text{MAP}},\ N^{(j)}_{AW},\ T^{(ji)}_{\text{data}},\ \tau^{(ji)}_{Ack},\ B_j\right\}. \tag{12}$$

The format of the CTS packet is:

$$\text{CTS}(i \to j) = \left\{i,\ j,\ P^{(ji)}_{\text{POWMAC}},\ P^{(i)}_{\text{MTI}},\ P^{(i)}_{CTS},\ N^{(j)}_{AW},\ T^{(ji)}_{\text{data}}, B_j,\ \tau^{(ji)}_{Ack}\right\}. \tag{13}$$

Finally, the format of the DTS packet is:

$$\text{DTS}(j \to i) = \left\{j,\ i,\ P^{(ji)}_{\text{POWMAC}},\ P^{(j)}_{\text{MTI}},\ \tau^{(ji)}_{Ack}\right\}. \tag{14}$$

I. POWMAC Limitations

In this section, we discuss some of the limitations of POWMAC and outline possible remedies for them. Specifically, we present two scenarios where concurrent transmissions in the same vicinity are, in principle, possible but may not be allowed under POWMAC.

So far, we have assumed that slave terminals are in the transmission range of only one master terminal. However, this may not be true; the example shown in Figure 12 presents a scenario where slave terminal v is within the transmission range of the two (unsynchronized) master terminals j and l. According to POWMAC, terminal v may send its RTS packet (or respond with a CTS to terminal n) only if the two master terminals' AWs are misaligned by less than the maximum backoff window B, since otherwise, the control/data packets sent by terminal v will not be synchronized with at least one of its masters.

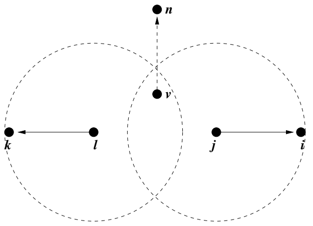


Fig. 12. Example of a slave terminal v that falls in the transmission ranges of two (unsynchronized) master terminals j and l.

A second scenario is shown in Figure 13, where terminal n has synchronized with the master j as a result of hearing j's RTS packet, while terminal v is out of j's transmission range, and is thus unaware of the j's AW. According to POWMAC, if n receives an RTS packet from v, then it responds with a CTS only if v's proposed AW is misaligned with j's AW by less than the maximum backoff window B.

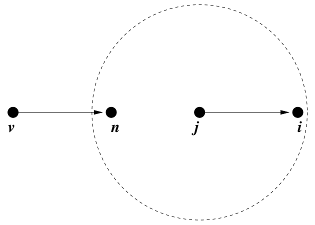


Fig. 13. Terminal n, which is in j's vicinity, receives an RTS message from terminal v, which is not in j's vicinity. Terminal n may or may not be able to respond to v's RTS message.

A close look at the above two scenarios reveals that they both occur when a terminal, say v, that is two hops away (see Figure 13) from a master terminal (j in that figure), is unaware of j's AW slots alignment. If v starts its own AW, then there is a good chance that the AWs of j and v are not synchronized. One possible approach that can reduce the chances of such scenarios to occur is to allow terminals that *overhear* any RTS/CTS/DTS messages (e.g., terminal n), to send their own RTS messages before terminals that are outside j's range (e.g., terminal v) send theirs. The idea here is to allow more terminals to synchronize with the same master. We cannot actually guarantee that n sends its RTS before v, because of the randomness in the contention resolution mechanism; however, what we can do is to increase the access probability x_n of terminal n (see (10)) beyond that of x_v, thus reducing the probability that the above two scenarios will occur.

J. Protocol Recovery

In [11] the authors observed that when the transmission and propagation times of control packets are long, the likelihood of a collision between a CTS packet and an RTS packet of another contending terminal increases dramatically; the vulnerable period being twice the transmission duration of a control packet. At high loads, such a collision can lead to collisions with data packets, as illustrated in Figure 14. In this figure, terminal D starts sending a RTS to terminal C while C is receiving B's CTS that is intended to A. A collision happens at C, hence C is unaware of B's subsequent data reception. Afterwards, if C receives a retransmitted RTS packet from node D and decides to reply back with a CTS, it may destroy B's reception.

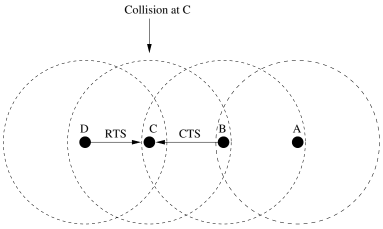


Fig. 14. Example of a collision between control packets that eventually leads to a collision with a data packet.

Another problem is if the interference goes above the planned interference tolerance $P^{(\cdot)}_{\text{MAI-add}}$. In POWMAC, we rely on two mechanisms to solve the above two problems. First, we require the carrier-sense range to be at least twice the maximum transmission range[9]. This makes the vulnerable period twice the propagation delay (less than 1 microsecond) instead of twice the transmission duration of a control packet (in the order of 100s of microseconds) and thus, the chances of control packets collisions will decrease significantly in the case of no channel shadowing effect. The second mechanism is to send a control packet preventing a potential interferer from commencing its transmission. In other word, suppose that while waiting in an AW to receive a data packet, terminal i hears an RTS message (destined to any terminal) that contains an allowable power $P^{(\cdot)}_{\text{map}}$ value that if used could cause an unacceptable interference with i's scheduled reception. Then terminal i shall respond *immediately* with a special CTS, preventing the RTS sender from commencing its transmission. This

[9] In fact, typical values for the carrier-sense range are more than twice the transmission range [19].

method is similar to the use of the Object-to-Send (OTS) control packet proposed in [43], [42]. To see how this solution helps in reducing the likelihood of collisions with data packets, consider the situation in Figure 14. Suppose that terminal A sends a RTS to terminal B, and B responds back with a CTS that collides at C with a RTS from D. Now, C does not know about B's ongoing reception. Two scenarios can happen. In the first, terminal C may later wish to send a packet to, say, terminal D. It sends a RTS, which will be heard by terminal B. B responds back with a special CTS. Note that there is a good chance that B's special CTS will collide with the CTS reply from D; however, this is desirable since C will fail to recover D's CTS packet, and will therefore defer its transmission and invoke its backoff procedure. In essence, B's special CTS acts as a jamming signal to prevent C from proceedings with its transmission.

Note that in POWMAC we try to avoid likely collision scenarios such as the one mentioned in [11]. However, there are still few complicated (and definitely much less probable) scenarios where data packets may collide; recovery from such collisions is left to the upper layers.

K. *Mobility and POWMAC*

To determine the TP for data packets, POWMAC relies on the assumption that the channel gain determined at the time of the RTS/CTS/DTS exchange is stationary for the duration of the current AW and the ensuing data packets. The channel gain can change as a result of mobility. However, as we now explain, such a change has no impact on the assumptions used in POWMAC.

For *large-scale* channel variations (e.g., mean channel gain), mobility has negligible impact on POWMAC since packet transmission times occur on the scale of few milliseconds while mobility occurs on the time scale of seconds. So the time between a control packet and an ensuing data packet is small enough to make the estimation sufficiently accurate. As for *small-scale* channel variations, although their impact can be mitigated through diversity techniques at the physical layer (e.g., RAKE receivers [34]), we argue that even if such techniques are not available, the channel stationarity assumption in POWMAC is still valid. Consider a multipath environment, where multiple versions of the transmitted signal arrive at the receiver at slightly different times and combine to give a resultant signal that can vary widely in amplitude and phase. The spectral broadening caused by this variation is measured by the Doppler spread, which is a function of the relative velocity (v) of the mobile and the angle between the direction of motion and the directions of arrival of the multipath waves [34]. This variation can be equivalently measured in the time domain using the *coherence time* (T_c), which is basically a statistical measure of the time duration over which the channel can be assumed time invariant. As a rule of thumb in modern communication system, $T_c \approx 0.423/f_m$, where $f_m = v/\lambda$ is the maximum Doppler shift and λ is the wavelength of the carrier signal.

Now, at a mobile speed of $v = 1$ meter/sec and 2.4 GHz carrier frequency, $T_c \approx 52.89$ msec. This time reduces to 10.56 msec when $v = 5$ meters/sec. For the channel stationarity assumption in POWMAC to be valid, the access window and the data packet duration must not exceed T_c. At a channel rate of 2 Mbps, it takes 4 msec to transmit a 1000-byte packet. This duration of time becomes even less at higher data rates. The propagation delay and the turnaround time (time it takes a terminal to switch from a receiving mode to a transmitting mode) are in the order of microseconds, and so they can be safely ignored. Thus, the assumption about channel stationarity is valid for moderate values of mobility (e.g., pedestrian speeds). The IEEE 802.11 was designed for such mobility scenarios [7]. In cases when terminals move faster, the packet size can be shortened so that the stationarity assumption still holds.

L. *POWMAC in Rate-Controlled Environments*

In this section, we explain how rate control can be combined with the POWMAC protocol. The IEEE 802.11b specifications provide a physical-layer multi-rate capability. All control packets are transmitted at the lowest rate (1 Mbps) to achieve the maximum range, while data packets can be transmitted at rates 1, 2, 5.5, and 11 Mbps. These different rates are achieved using multiple modulation schemes; BPSK, QPSK, and two variants of CCK. The higher is the rate, the higher is the SINR threshold (i.e., μ^*) that is needed to achieve the target BER.

Several schemes have been proposed for rate adaptation (e.g., [17]). The main idea in such schemes is to use the measured SINR of the received RTS packets to set the transmission rate for each data packet according to the highest feasible value allowed by the channel condition[10]. These schemes use a fixed TP, and a higher rate if the measured SINR is more than μ^*, i.e., these approaches utilize the additional available power in the received signal to allow for a higher rate. POWMAC, on the other hand, utilizes that additional signal power to allow for interference-limited transmissions in the neighborhood of a receiver. This, however, does not mean that a TP scheme and a rate control scheme cannot be combined together. In fact, it was shown in [14] that adapting the transmit power, data rate, and coding scheme achieves maximum spectral efficiency. For example, one way to integrate the protocol proposed in [17] with POWMAC is as follows. First, the maximum feasible rate is chosen according to the scheme in [17]. Second, the POWMAC protocol is used with the required μ^* for that chosen rate being used in (3). The message here is that POWMAC and rate-control schemes are complementary for maximizing network throughput. Please refer to [25] for more details.

M. *Protocol Overhead*

We now explore, using a simplified analysis, the potential throughput improvement of a multi-rate POWMAC protocol over a multi-rate 802.11 scheme. Let L_c be the total length (in bits) of the IEEE 802.11 RTS plus CTS packets. The total length of the POWMAC RTS, CTS, plus DTS packets is $\approx 1.68L_c$. Hence, the length of the AW slot is $1.68L_c + B$ (recall that B is the maximum backoff duration). Let L_d be the average data packet length. Let R_c and R_d be the transmission rates of control and data packets, respectively. Suppose that there are N feasible simultaneous in the same vicinity. The duration of time it takes to send N data packets according to POWMAC is $T_{POWMAC} = N\left(\frac{1.68L_c}{R_c} + B\right) + \frac{L_d}{R_d}$. The duration of time it takes to send the same N packet according to the IEEE 802.11 is $T_{802.11} = N\left(\frac{L_c}{R_c} + \frac{L_d}{R_d}\right)$ [11]. Computing T_{POWMAC} and $T_{802.11}$ in this way is quite optimistic since we are assuming that for POWMAC, all AW slots result in successful RTS/CTS/DTS exchanges, and that for the 802.11 scheme, an RTS/CTS exchange follows immediately the transmission of the previous data packet[12].

For POWMAC to outperform the 802.11 scheme, we must have $T_{POWMAC} < T_{802.11}$. With some manipulations, this condition can be written as $(0.68)\frac{R_d}{R_c} + \frac{BR_d}{L_c} < \frac{N-1}{N}\frac{L_d}{L_c}$. Clearly, the larger the ratio $\frac{R_d}{R_c}$, the lesser is the improvement of POWMAC over the 802.11. Furthermore, the greater the value of N, the more is the improvement of POWMAC over the 802.11. For example, according to the IEEE 802.11b specifications, the maximum value of R_d/R_c is 11 ($R_d = 11$ Mbps). Furthermore, L_d is typically in the order of tens of L_c. For example, for 2-KB data packets, $L_d/L_c \approx 59$. Using these values, it can be shown that as long as $N > 1$, POWMAC will outperform the 802.11 scheme. Even for N as small as 2, T_{POWMAC} is only 73% of $T_{802.11}$.

IV. PERFORMANCE EVALUATION

A. *Simulation Setup*

We now evaluate the performance of the POWMAC protocol and contrast it with the IEEE 802.11 scheme. Note that we do not compare POWMAC to energy-oriented protocols (e.g., [15], [19], [20], [30]), since at best these protocols give comparable throughput to that of the 802.11 scheme. Furthermore, since POWMAC uses a single-channel, single-transceiver design, it is unfair to compare it with two-channels, two-transceivers based protocols (e.g., [23], [24], [40]). Our results are based on simulation experiments conducted using CSIM programs (CSIM is a C-based process-oriented discrete-event simulation package [3]). For simplicity, data packets are assumed to be of a fixed size. The routing overhead is ignored since the goal here is to evaluate the performance improvements due to the MAC protocol. Furthermore, because the interference margin is chosen so that the maximum transmission range under the POWMAC and 802.11 protocols is the same, it is safe to assume that both protocols achieve the same forward progress per hop. Consequently,

[10] Note that in the above schemes, the RTS and CTS packets are still transmitted at the lowest rate so that neighboring terminals can overhear these packets and are informed of the ensuing data transmission

[11] For simplicity, the Ack packet overhead is not considered.

[12] The IEEE 802.11 scheme requires terminals to backoff after the end of a data transmission even if the channel is idle.

we can focus on the one hop throughput, i.e., the packet destination is restricted to one hop from the source. The two-ray propagation model is used, and the capture model is similar to the one in [38]. Other parameters used in the simulations are given in Table I. These parameters correspond to realistic hardware settings [4]. According to these parameters, each node has, on average, ten neighbors.

Data packet size	2 KB
Data rate	1 Mbps
SINR threshold	6 dB
Transmission range	750 meters
Carrier-sense range	1500 meters
Path loss factor	4
ξ_{max}	7 dB

TABLE I

PARAMETERS USED IN THE SIMULATIONS.

B. Macroscopic Results

We first simulate a set of basic scenarios for the purpose of highlighting the advantages and operational details of POWMAC. Consider the line topology in Figure 15. The distances between the terminals are also shown in the figure. Terminal A is transmitting to node B, and node C is transmitting to node D. Persistent load is used in this experiment, i.e., terminals A and C always have packets to send. The transmissions from A and C interfere with the data reception at D and B, respectively. However, the interference from A to D is much smaller than the one from C to B, and so in the following discussion, we focus on the latter one.

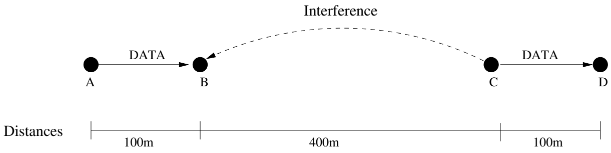


Fig. 15. Toy topology where the two interfering transmissions $A \rightarrow B$ and $C \rightarrow D$ can proceed simultaneously if A's and C's transmission powers are appropriately chosen.

In the first scenario, node B starts moving in the direction of node C at speed of 5 m/s. Figure 16(a) depicts the throughput of the network as a function of time. According to the 802.11 scheme, only one transmission can proceed at a time since all terminals are within the carrier-sense range of each other. However, according to POWMAC, for the first 12 seconds, the two transmissions $A \rightarrow B$ and $C \rightarrow D$ can proceed simultaneously, resulting in about 84% improvement in network throughput. For the next 40 seconds, as node B gets closer to node C, the channel gain G_{BC} increases and so $P^{(C)}_{\text{MAP}}$ decreases until it becomes less than the one required by node D to achieve its SINR threshold. Therefore, once node A exchanges RTS/CTS/DTS with B, node C cannot transmit to D[13]. On the other hand, if node C exchanges RTS/CTS/DTS packets with D before A does that with B, then node A increases its TP to overcome the interference induced from C at node B. Hence, the two transmissions $A \rightarrow B$ and $C \rightarrow D$ can proceed simultaneously. Roughly, half of the time A starts before C and half of the time C starts before A, so the throughput enhancement is about 34% during the period between 12 and 52 seconds. After 52 seconds, the interference at B due to C becomes larger than the one allowed by the planned loading, so either $A \rightarrow B$ or $C \rightarrow D$ can proceed, but not both. The small degradation in throughput after 52 seconds is attributed to the overhead of the AW when no simultaneous transmissions are taking place.

In the second scenario, terminal C moves in the direction of B at a speed of 5 m/s, while all other terminals are stationary. Figure 16(b) shows the throughput of the network as a function of time. The difference between this scenario and the previous one is that this time, not only is $P^{(C)}_{\text{MAP}}$ decreasing (as a result of G_{BC} increasing), but $P^{(CD)}_{\text{POWMAC}}$ is also increasing as a result of the decrease in G_{CD}. In the first 12 seconds, the two transmissions $A \rightarrow B$ and $C \rightarrow D$ can proceed simultaneously. Between 12 and 22 seconds, the throughput enhancement is 34% for the same reason given in the previous scenario. After that, only one transmission proceeds, and the throughput becomes comparable to that of the 802.11 scheme.

C. Random Grid Topologies

We now study the performance under more realistic network topologies. First, we consider a *random-grid* topology, where 25 mobile terminals are placed within a square area of length 1500 meters. The square is split into 25 smaller squares, one for each terminal. The location of a mobile terminal within the small square is selected randomly. For each generated packet, the destination terminal is selected randomly from the one-hop neighbors. Each terminal generates packets according to a Poisson process with rate λ (same for all terminals). The Random Waypoint model [10] is used for mobility, with a terminal speed that is uniformly distributed between 0 and 2 meters/sec.

The performance is demonstrated in Figure 17. Part (a) of the figure depicts the throughput versus λ. It can be shown that at high loads, POWMAC achieves about 50% increase in throughput over the IEEE 802.11 scheme. This increase is attributed to the increase in the number of simultaneous transmissions. Part (b) of Figure 17 depicts the energy consumption versus λ. This is the total energy used to *successfully transmit* a packet. It includes the energy used to transmit control packets and the lost energy in retransmitting data and control packets in case of collisions[14]. For all cases, POWMAC requires roughly the same energy required by the 802.11 scheme. These results are in line with the analysis in Section III-C, where the interference margin was chosen so that both protocols consume the same energy per bit.

D. Clustered Topologies

Next, we study the performance of POWMAC under *clustered* topologies. In such topologies, a terminal communicates mostly with terminals within its own cluster, and rarely with neighboring clusters. These topologies are common in practice (e.g., a historical site where users of wireless devices move in groups). To generate a clustered topology, we consider an area of dimensions 600×600 (in meters). Sixteen terminals are split into 4 equal groups, each occupying a 100×100 square in one of the corners of the complete area. For a given source terminal, the destination is selected from the same cluster with probability $1 - p$ or from a different cluster with probability p. In each case, the selection from within the given cluster(s) is done randomly.

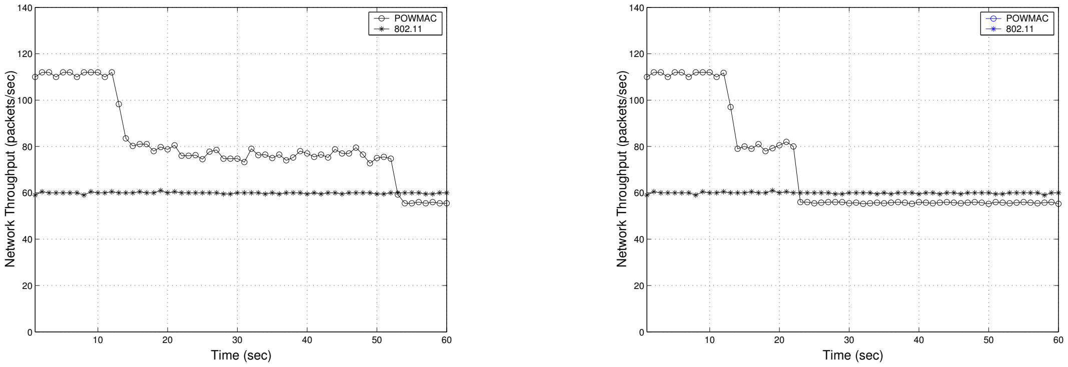
Part (a) of Figure 18 depicts the network throughput versus λ for $p = 0.25$ and $p = 0.5$. According to the 802.11 scheme, only one transmission can proceed at a time since all terminals are within the carrier-sense range of each other. Furthermore, its throughout performance is approximately the same regardless of the value of p. In other words, the 802.11 scheme does not benefit from the locality of the traffic. On the other hand, according to POWMAC, two to three transmissions can proceed simultaneously, resulting in a significant improvement in network throughput. Moreover, it is clear that POWMAC utilizes traffic locality to increase network throughput; its performance is better for smaller values of p.

Part (b) of the figure shows that POWMAC saves a significant amount of energy relative to the 802.11 scheme. Since a terminal communicates mostly with terminals within its own cluster, the destination terminal is within 100 meters of the source terminal, thus requiring much less TP than $P_{\max}$. This is the reason why the figure shows a huge advantage of POWMAC over the 802.11 scheme in terms of energy consumption. So, although POWMAC was designed with the goal of increasing throughput, significant energy may be saved as a consequence of reducing the TP. Understandably, energy saving increases as traffic becomes more localized (i.e., when $p = 0.25$).

Next, we show the strong parallelism that is achieved by POWMAC. To this end, we study the percentage of time during which N transmissions

[13] When C sends an RTS to D, D replies with a negative CTS since $P^{(C)}_{\text{MAP}}$ is less than $P^{(CD)}_{\text{POWMAC}}$ as computed by node D.

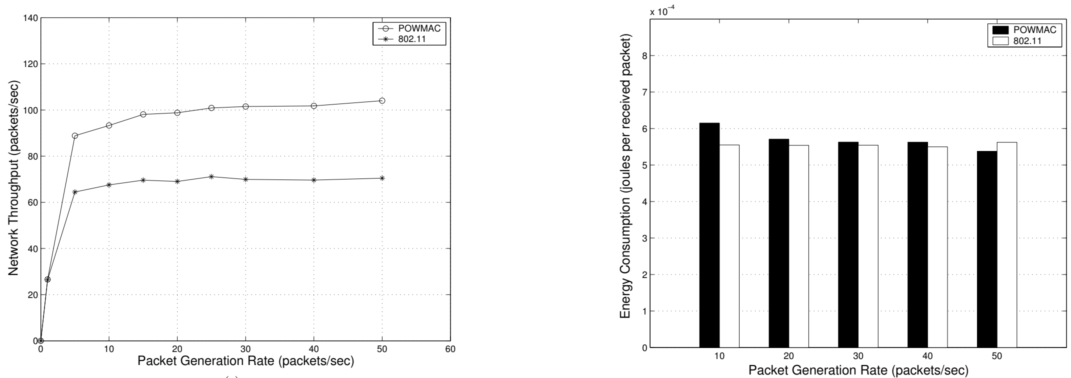
[14] The *processing* power consumption in the transmitter and the receiver circuitry is the same for both POWMAC and the IEEE 802.11 scheme. Furthermore, this power depends on the wireless card used, unlike the TP consumption. Our energy model accounts only for the TP.



(a) Scenario 1: B is moving towards C.

(b) Scenario 2: C is moving towards B.

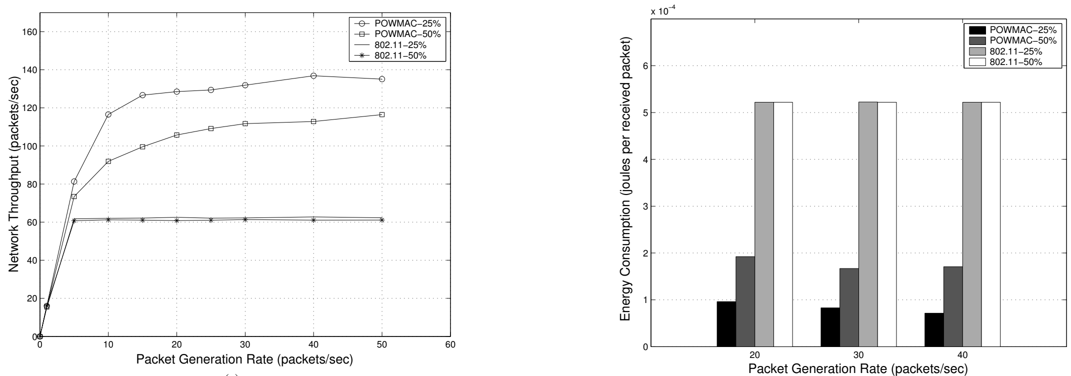
Fig. 16. Performance of the POWMAC and the 802.11 protocols (line topologies).



(a)

(b)

Fig. 17. Performance of the POWMAC and 802.11 protocols as a function of λ (random-grid topologies).



(a)

(b)

Fig. 18. Performance of the POWMAC and 802.11 protocols as a function of λ (clustered topologies).

take place simultaneously in the *same* neighborhood. Figure 19 depicts this measure of performance. The 802.11 scheme allows for only one transmission in a neighborhood, and so for both cluster and grid topologies, $N = 1$ for all the time. In contrast, POWMAC allows for up to 5 and 7 simultaneous transmissions in the same neighborhood in the random-grid and clustered topologies, respectively.

Figure 20 shows the time evolution of the AW for a typical terminal. To produce this figure, we look at the AWS of a terminal that is located roughly in the center of the random-grid topology. The initial value of the AW was 4. The terminal changes its AWS every time a data packet has been received by monitoring the measured interference during that packet reception versus $\xi_{\max}$, as explained in Section III-G. Over several runs, *the average size of the AW was found to be approximately 3.*

Table II shows the percentage of data collisions under different load conditions for both POWMAC and the IEEE 802.11 scheme. Because of its conservative design, the 802.11 scheme results in fewer collisions. The price, however, is loss in throughput. On the other hand, POWMAC takes an avant-garde approach of allowing concurrent interference-limited transmissions. Although POWMAC results in more collisions, it is able to significantly improve the overall network performance (i.e., throughput).

λ (packets/sec)	802.11 (%)	POWMAC (%)
1	1	1.5
5	4.7	12.5
10	6.2	17.4
15	9.2	19.1
20	9.0	17.9

TABLE II

PERCENTAGE OF DATA-PACKET COLLISIONS AS A FUNCTION OF λ.

Next, recall that in Section III-H, we pointed out that both POWMAC and the 802.11 scheme cannot completely eliminate collisions due to interference. The reason is that the interference range is typically larger than

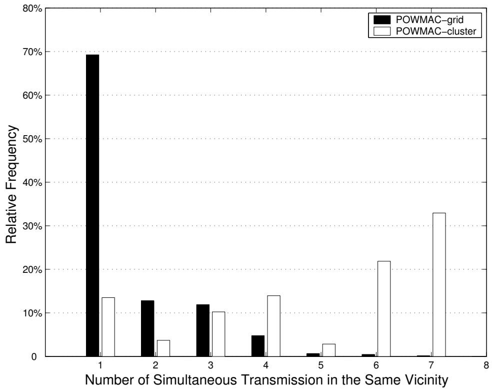


Fig. 19. Relative frequency with which N transmissions take place simultaneously in the same neighborhood.

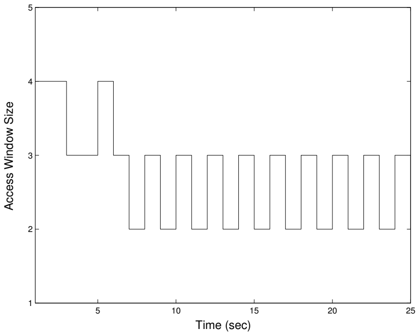


Fig. 20. Time evolution of the AW for a typical terminal.

the transmission range. To address this limitation in the 802.11 scheme, the authors in [41] proposed to limit the communication range to a value that is below the maximum one. In other words, if the minimum power required to receive a packet at a terminal, say i, is P_{th}, then i responds with a CTS packet only if the reception power is at least γP_{th}, where $\gamma > 1$. While it does not solve the problem completely, this solution reduces the severity of it; the price being decreasing the transmission range and affecting the topological properties of the network. For similar reasons, POWMAC would also benefit from this solution. Figure 21 depicts the throughput of POWMAC and the 802.11 scheme versus the transmission range. The experiment was conducted for a random-grid topology with each terminal generating packets according to a Poisson process at a rate of 20 packets/sec. It is shown that the throughput performance of both POWMAC and the 802.11 scheme increases by up to 26% and 22% respectively, as the transmission range is decreased from 100% to 70% of the maximum range.

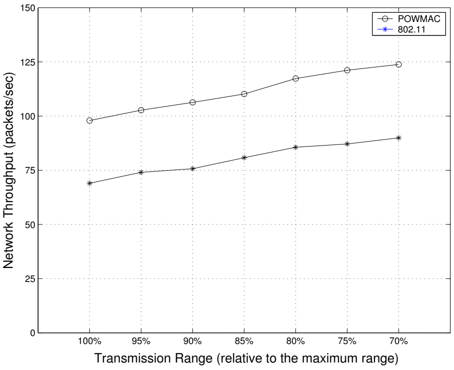


Fig. 21. Performance of the POWMAC and the 802.11 protocols as a function of the transmission range.

To study the impact of α in (9) on the performance of POWMAC, we run simulations for the random-grid topology using different values of α. The throughput is shown in Table III. It is clear that the throughput does not vary much with α, when α is between 0.1 and 0.9. However, as α increases to 3.0, the throughput decreases noticeably, since receivers' interference tolerances become too small for the receivers' neighbors to start their own transmissions.

α	0.1	0.5	0.9	5.0
Throughput (packets/sec)	104	106	102	90

TABLE III

PERFORMANCE OF POWMAC AS A FUNCTION OF α.

Finally, we study the impact of the data packet size on the performance. We run simulations for the random-grid topology, where each terminal generates packets according to a Poisson process with $\lambda = 20$ packets per second. The performance versus the packet size is shown in Figure 22. Part (a) of the figure shows that the throughput enhancement of the proposed protocol is lesser for shorter packets, which agrees with the analysis given in Section III-M. Part (b) of that figure shows that the energy consumption decreases as we increase data packets' sizes. This is again not surprising, since the fraction of energy consumed on control packet compared to data packets is smaller when data packets are larger.

V. CONCLUSIONS AND FUTURE WORK

In this paper, we proposed a power controlled MAC protocol for MANETs, known as POWMAC. Similar to the 802.11 scheme, POWMAC is based on a single-transceiver circuitry, and it operates over a single channel for data and control packets. POWMAC adjusts the transmission powers of data packets to allow for some interference margin at the receiver. Multiple interference-limited transmissions in the vicinity of a receiver are allowed to overlap in time, provided that their MAI effects do not lead to collisions at nearby receivers.

We compared the performance of POWMAC with that of the IEEE 802.11 scheme. Our simulation results showed that POWMAC can improve the network throughput by up to 50% in random-grid topologies and much more than that in clustered topologies. Furthermore, POWMAC can achieve some reduction in the energy consumed to successfully deliver a packet from the source to the destination. To the best of our knowledge, POWMAC is the first single-channel protocol that utilizes TPC to increase network throughput while preserving the collision avoidance property of the 802.11 scheme.

Besides tuning the parameters of POWMAC and investigating its performance under various scenarios and topologies, our future work will address other techniques for capacity improvement in POWMAC. Because of its demonstrated superior performance (compared to TDMA and FDMA), CDMA has been chosen as the access technology of choice in cellular systems, including the recently adopted 3G systems. It is, therefore, natural to explore the potential of integrating CDMA into the design of the POWMAC protocol.

REFERENCES

[1] http://www.ietf.org/html.charters/manet-charter.html.

[2] International Standard ISO/IEC 8802-11; ANSI/IEEE Std 802.11, 1999 Edn. Part 11: wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications.

[3] Mesquite Software Incorporation, http://www.mesquite.com.

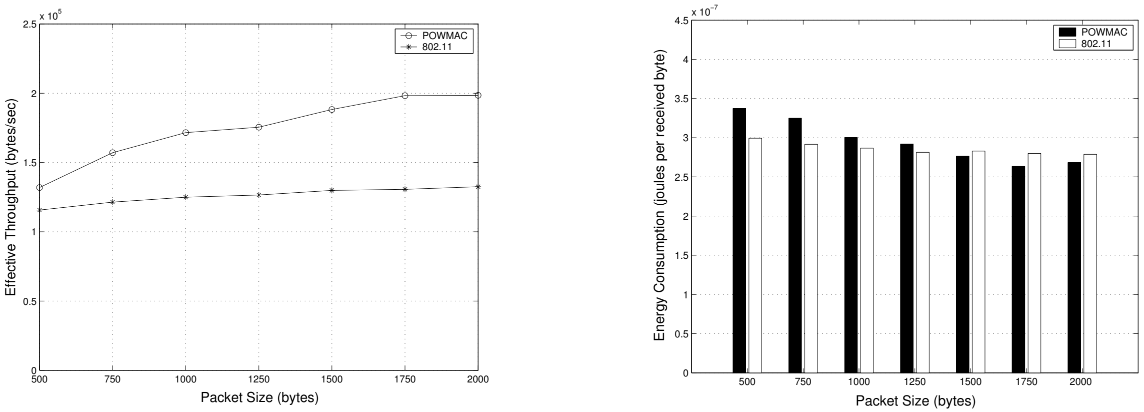
[4] The Cisco Aironet 350 Series of wireless LAN, http://www.cisco.com/warp/public/cc/pd/witc/ao350ap.

[5] A. Acharya, A. Misra, and S. Bansal. MACA-P : a MAC for concurrent transmissions in multi-hop wireless networks. In *Proceedings of the First IEEE PerCom 2003 Conference*, pages 505–508, Mar. 2003.

[6] S. Agarwal, R. H. Katz, S. V. Krishnamurthy, and S. k. Dao. Distributed power control in ad-hoc wireless networks. In *IEEE International Symposium on Personal, Indoor and Mobile Radio Communications*, volume 2, pages 59–66, Oct. 2001.

[7] K. Aretz, M. Haardt, W. Konhauser, and W. Mohr. The future of wireless communications beyond the third generation. *Computer Networks*, 37:83–92, 2001.

[8] N. Bambos. Toward power-sensitive networks architecture in wireless communications: Concepts, issues, and design aspects. In *IEEE Personal Communications Magazine*, volume 5, pages 50–59, June 1998.



(a)

(b)

Fig. 22. Performance of the POWMAC and 802.11 protocols as a function of the data packet size (random-grid topologies).

[9] V. Bharghavan, A. Demers, S. Shenker, and L. Zhang. MACAW: A media access protocol for wireless LANs. In *Proceedings of the ACM SIGCOMM Conference*, volume 24, pages 212–225, 1994.

[10] J. Broch, D. A. Maltz, D. B. Johnson, Y.-C. Hu, and J. Jetcheva. A performance comparison of multi-hop wireless ad hoc network routing protocols. In *Proceedings of the ACM MobiCom Conference*, pages 85–97, 1998.

[11] J. Deng and Z. Haas. Dual busy tone multiple access (DBTMA): A new medium access control for packet radio networks. In *Proceedings of the IEEE ICUPC Conference*, pages 973–977, 1998.

[12] T. ElBatt and A. Ephremides. Joint scheduling and power control for wireless ad-hoc networks. In *Proceedings of the IEEE INFOCOM Conference*, pages 976–984, 2002.

[13] T. A. ElBatt, S. V. Krishnamurthy, D. Connors, and S. Dao. Power management for throughput enhancement in wireless ad-hoc networks. In *Proceedings of the IEEE ICC Conference*, pages 1506–1513, 2000.

[14] A. Goldsmith and P. Varaiya. Increasing spectral efficiency through power control. In *Proceedings of the IEEE ICC Conference*, pages 600 – 604, 1993.

[15] J. Gomez, A. T. Campbell, M. Naghshineh, and C. Bisdikian. PARO: supporting dynamic power controlled routing in wireless ad hoc networks. *ACM/Kluwer Journal on Wireless Networks*, 9(5):443–460, 2003.

[16] P. Gupta and P. R. Kumar. The capacity of wireless networks. *IEEE Transactions on Information Theory*, 46(2):388–404, Mar. 2000.

[17] G. Holland, N. Vaidya, and P. Bahl. A rate-adaptive mac protocol for multihop wireless networks. In *Proceedings of the ACM MobiCom Conference*, pages 236–251, 2001.

[18] P. Johansson, T. Larsson, N. Hedman, B. Mielczarek, and M. Degermark. Scenario-based performance analysis of routing protocols for mobile ad-hoc networks. In *Proceedings of the ACM MobiCom Conference*, pages 195–206, 1999.

[19] E.-S. Jung and N. H. Vaidya. A power control MAC protocol for ad hoc networks. In *Proceedings of the ACM MobiCom Conference*, pages 36–47, 2002.

[20] P. Karn. MACA - a new channel access method for packet radio. In *Proceedings of the 9th ARRL Computer Networking Conference*, pages 134–140, 1990.

[21] V. Kawadia and P. R. Kumar. Power control and clustering in ad hoc networks. In *Proceedings of the IEEE INFOCOM Conference*, pages 459–469, 2003.

[22] T. J. Kwon and M. Gerla. Clustering with power control. In *Proceedings of the IEEE MILCOM Conference*, pages 1424–1428, 1999.

[23] J. Monks, V. Bharghavan, and W.-M. Hwu. A power controlled multiple access protocol for wireless packet networks. In *Proceedings of the IEEE INFOCOM Conference*, pages 219–228, 2001.

[24] A. Muqattash and M. Krunz. Power controlled dual channel (PCDC) medium access protocol for wireless ad hoc networks. In *Proceedings of the IEEE INFOCOM Conference*, pages 470–480, 2003.

[25] A. Muqattash and M. Krunz. POWMAC: A single-channel power-control protocol for throughput enhancement in wireless ad hoc networks. Technical report CENG-TR-245, University of Arizona, Dept. of Electrical & Computer Engineering, Sept. 2004. http://www.ece.arizona.edu/∼alaa/powmac-tech-report.pdf.

[26] T. Nandagopal, T.-E. Kim, X. Gao, and V. Bharghavan. Achieving mac layer fairness in wireless packet networks. In *Proceedings of the ACM MobiCom Conference*, pages 87–98, 2000.

[27] S. Narayanaswamy, V. Kawadia, R. S. Sreenivas, and P. R. Kumar. Power control in ad-hoc networks: Theory, architecture, algorithm and implementation of the COMPOW protocol. In *Proceedings of the European Wireless Conference*, pages 156–162, 2002.

[28] T. Ojanperä and R. Prasad. *Wideband CDMA For Third Generation Mobile Communications*. Artech House, Incorporated, 1998.

[29] D. R. Pauluzzi and N. C. Beaulieu. A comparison of SNR estimation techniques for the AWGN channel. *IEEE Transactions on Communications*, 48:Pages:1681–1691, 2000.

[30] M. B. Pursley, H. B. Russell, and J. S. Wysocarski. Energy-efficient transmission and routing protocols for wireless multiple-hop networks and spread spectrum radios. In *Proceedings of the EUROCOMM Conference*, pages 1–5, 2000.

[31] D. Qiao, S. Choi, A. Jain, and K. G. Shin. Miser: an optimal low-energy transmission strategy for ieee 802.11a/h. In *Proceedings of the 9th Annual International Conference on Mobile Computing and Networking*, pages 161–175, 2003.

[32] R. Ramanathan and J. Redi. A brief overview of ad hoc networks: challenges and directions. *IEEE Communications Magazine*, 40(5):20–22, May 2002.

[33] R. Ramanathan and R. Rosales-Hain. Topology control of multihop wireless networks using transmit power adjustment. In *Proceedings of the IEEE INFOCOM Conference*, pages 404–413, 2000.

[34] T. Rappaport. *Wireless Communications: Principles and Practice*. Prentice Hall, 2002.

[35] T. S. Rappaport and L. B. Milstein. Effects of radio propagation path loss on DS-CDMA cellular frequency reuse efficiency for the reverse channel. *IEEE Transactions on Vehicular Technology*, 41:231–242, Aug. 1992.

[36] V. Rodoplu and T. Meng. Minimum energy mobile wireless networks. *IEEE Journal on Selected Areas in Communications*, 17(8):1333–1344, 1999.

[37] 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 Transactions on Communications*, 23(12):1417–1433, 1975.

[38] C. Ware, J. Chicharo, and T. Wysocki. Simulation of capture behaviour in IEEE 802.11 radio modems. In *Proceedings of the IEEE Vehicular Tech. Conference*, volume 3, pages 1393–1397, Fall 2001.

[39] R. Wattenhofer, L. Li, P. Bahl, and Y.-M. Wang. Distributed topology control for power efficient operation in multihop wireless ad hoc networks. In *Proceedings of the IEEE INFOCOM Conference*, pages 1388–1397, 2001.

[40] S.-L. Wu, Y.-C. Tseng, and J.-P. Sheu. Intelligent medium access for mobile ad hoc networks with busy tones and power control. *IEEE Journal on Selected Areas in Communications*, 18(9):1647–1657, 2000.

[41] K. Xu, M. Gerla, and S. Bae. How effective is the IEEE 802.11 RTS/CTS handshake in ad hoc networks? In *Proceedings of the IEEE GLOBECOM Conference*, volume 1, pages 72–76, Nov. 2002.

[42] C.-H. Yeh. ROAD: a variable-radius MAC protocol for ad hoc wireless networks. In *Proceedings of the IEEE Vehicular Tech. Conference*, volume 1, pages 399–403, 2002.

[43] H. Zhou, C.-H. Yeh, and H. Mouftah. A solution for multiple access in variable-radius mobile ad hoc networks. In *Proceedings of the IEEE International Conference on Communications, Circuits and Systems*, volume 1, pages 150–154, 2002.