Collision-Free Asynchronous Multi-Channel Access in Ad Hoc Networks

Duy Nguyen[†]
[†]Computer Engineering Department
University of California, Santa Cruz
Santa Cruz, CA 95064, USA
Email: duy@soe.ucsc.edu

J.J. Garcia-Luna-Aceves[†,*]
[*]Palo Alto Research Center
3333 Coyote Hill Road
Palo Alto, CA 94304, USA
Email: jj@soe.ucsc.edu

Katia Obraczka[†]
[†]Computer Engineering Department
University of California, Santa Cruz
Santa Cruz, CA 95064, USA
Email: katia@soe.ucsc.edu

***Abstract*—In this paper, we present a collision-free asynchronous multi-channel access protocol for Ad Hoc wireless networks using a single transceiver. Our protocol, dubbed AM-MAC for Asynchronous Multi-channel Medium Access Control, targets low-cost and low-power deployments where nodes are equipped with a single transceiver. Other distinguishing features of AM-MAC include its simplicity and the fact that it does not require temporal synchronization among nodes. This is accomplished through an asynchronous split phase together with an observation phase as well as an unique handshake. Nodes observe the control channel for a period of time before asynchronously switching to the negotiated channel. Protocol correctness and collision-freedom in a multi-channel environment are verified. We also provide an analytical throughput assessment for our multi-channel approach. Simulation results show that AM-MAC improves performance significantly when compared to IEEE 802.11 and exhibits comparable performance to MMAC, one of the well-known multi-channel medium access control protocols, without the need for temporal synchronization.**

I. Introduction

Bandwidth demand in wireless networks continues to rise as existing and emerging applications become increasingly popular, including multimedia streaming, emergency response and disaster rescue operations, smart environments, and others. As new physical layer technology (e.g., single-chip radios featuring advanced coding and modulation techniques) is able to sustain higher data rates, the fundamental limits challenging development and deployment of high data-rate, QoS-sensitive applications have been shifting from PHY to the medium access control (MAC) layer. Consequently, it is imperative to design efficient MAC techniques that will "expose" the underlying PHY's data rates to the applications, yet keeping cost low. Low cost is especially critical in dense deployments in "extreme", remote environments where nodes need to be disposable. Energy efficiency is another critical requirement in such environment.

Since Gupta and Kumar[5] established the scalability limitations of wireless ad-hoc networks (MANETs) in terms of their capacity to carry point-to-point traffic, numerous research efforts have focused on proposing techniques to improve this fundamental limitation. A notable example is the work by Glossglauser and Tse[4] that shows that with mobility, constant throughput is achievable as the number of nodes increases. More recently, multi-packet reception and transmission techniques have been proven to reverse the Gupta-Kumar scaling laws and effectively allow MANET capacity to increase as the number of nodes increases[3]. However, such techniques require multiple and more sophisticated radios, increasing cost and energy consumption considerably.

An alternative that is quite attractive given its cost-performance benefits is to make use of low-cost, low-power, widely available multi-channel radios. Indeed, while IEEE 802.11's Distributed Coordinate Function (DCF) has been originally designed for one common channel, the IEEE 802.11 PHY offers multiple channels transmission capability. More recently, a number of multi-channel MAC protocols have been proposed as a way to increase performance and network utilization. However, transmission over multiple channels raise a number of challenges including hidden terminals over multiple channels and node synchronization. In fact, as described in Section II, which overviews related work in multi-channel MAC, many proposed multi-channel MAC protocols rely heavily on temporal synchronization.

In this paper, we introduce the Asynchronous Multi-Channel Medium Access Control protocol, or AM-MAC for short, an asynchronous collision-free multi-channel medium-access control protocol. AM-MAC's main features include its efficient utilization of the medium and energy efficiency without the need for time synchronization. Additionally, AM-MAC's simplicity and ease of implementation makes it well suited for low-cost, limited-capability devices.

The rest of the paper is organized as follows. Section II reviews related work. Section III describes our protocol in detail. Section IV presents our simulation results while Section V presents our analytical throughput results. Finally, Section VI concludes the paper.

II. Background And Related Work

Multi-channel MAC approaches can be classified into the following categories: dedicated control channel, split phase, common hopping, and parallel rendezvous. We will briefly describe each approach; for more details, we refer the reader to [8].

In the dedicated control channel approach, a channel is reserved exclusively for exchanging control information. Usually, a node is equipped with 2 radios, in which one is solely used for control information exchange. For example, the DCA protocol [15] maintains a dedicated radio for control messages and the other for data transmission. Since each node has two transceivers, it can always monitor the control channel. The multi-channel hidden terminal problem simply does not exist in this case. However, an obvious drawback is the low spectrum utilization since one radio is reserved exclusively for control data exchange. Other drawbacks of the dedicated control channel approach include higher cost and energy consumption associated with equipping and operating nodes with two radios.

Split phase approaches avoid the need for multiple radios by having a dedicated common control channel and having nodes alternate between channel negotiation phases on the common channel and data transfer phases on the negotiated channels. This approach, however, requires tight time synchronization among all nodes in order to agree on the switching time between the control and data phases. Additionally, during the control phase, other channels become under-utilized. Another drawback is that if a node missed the negotiation period or did not succeed contending during this period, it would have to wait until the next negotiation phase to contend again. A notable example of the split phase approach is the MMAC protocol [10]. Wiflex [6] is another multi-channel MAC for OFDM-like PHY that uses split phase, but does not require nodes to be synchronized.

Protocols that follow the common hopping approach have all idle nodes follow a common hopping sequence. A pair of nodes will stop hopping as soon as they agree on a hopping sequence to communicate; they will go back to hopping at the end of their exchange. CHMA[13] is an example of the common hopping approach. It eliminates the need for carrier sensing and code assignment by allowing the source-destination pair to agree and remain in the same hopping frequency in order to communicate. The main drawbacks are frequent channel switching and tight synchronization requirement.

Finally, following the parallel rendezvous approach, each node publishes its own channel hopping schedule. Senders learn their receiver's current hopping sequence via a seed broadcast mechanism. Senders, then, must adopt the published sequence if they want to increase the time spend on the channel with the receiver. The advantage is that there is no single channel bottleneck. This approach ,however, does require tight synchronization. Examples include [8], [1].

From the previous discussion on the state-of-the-art of multi-channel MAC protocols, we note that most existing approaches require tight temporal synchronization among nodes and when that requirement is relaxed, collision-freedom cannot be guaranteed. AM-MAC provides an elegant and simple solution to achieving collision freedom without the need for temporal synchronization. The following section describes AM-MAC in detail.

III. AM-MAC PROTOCOL

We start by listing our design assumptions:

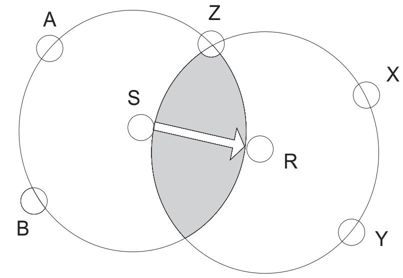


Fig. 1: Sender S and receiver R are transmitting the ATS packets. The shaded region shows the anticipated noise

- Each node is equipped with a single transceiver such that it can either transmit or receive, but not both simultaneously.
- A node's transceiver has N orthogonal channels of equal bandwidth; channel orthogonality means that simultaneous transmissions do not interfere with one another.

A. AM-MAC Handshake

For collision freedom (see Section 2), nodes must observe the control channel for a specified period of time. Any nodes interested in data transmissions must perform the handshake exchange accordingly.

We borrow the basic CSMA handshake mechanism and carrier sensing for channel negotiation. However, the regular RTS/CTS handshake is insufficient to provide collision-free data transmission. For example, consider the scenario in Figure 2. R sends a CTS to S after having received a RTS request; however X, which is hidden from S, begins to send a RTS to Y at the same time. X has no knowledge of R's channel selection. Thus a collision in the data channel may occur.

In order to guarantee collision-freedom without the need for nodes to be synchronized, we introduce the ATS (for Announce To Send) control frame. ATS is used to inform the sender's and receiver's neighbors of the ongoing data transmission or channel selection and, at the same time, serves as a jamming signal to prevent possible interference and to prevent hidden terminal problems.

Both sender and receiver need to broadcast their ATS packets which act like a busy-tone in order to broadcast channel selection information and prevent any potential interfering neighbors. This strategy is similar to the work done in [12] using a separate busy-tone channel. As for neighbors, they must adhere the to back-off rules stipulated by the control frame exchange. This is illustrated by an example in Figure 1 which shows the ATS packet transmissions by both S and R. Some neighbors of S, such as A and B, can receive the ATS packet; the same applies to neighbors X and Y of R. However, the mutual neighbor Z of S and R (in the shaded region) will only hear noise. Depending on the proximity of S and R's transmission, this overlapping region may shrink or expand. On the sender side, ATS informs neighbors who have not heard the CTS to be aware of the channel selection. Simply, the sender announce its intention to transmit twice, once through

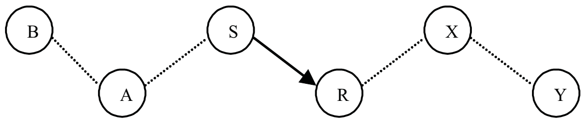


Fig. 2: Sender S and receiver R together with potential interferences from the neighbors

an RTS and once through an ATS. On the receiver side, the ATS acts like an extended CTS to announce the channel selection and jam any potential interfering neighbors. In the following section, we will show that this unique handshake ensures collision-free data transmission with some additional conditions.

The AM-MAC's control frames are:

- RTS (Request to Send): is used when a sender has data to send. We modified the RTS to contain some additional fields such as the available channel list, and data transfer time.
- CTS (Clear To Send): is used to acknowledge reception of the RTS. We introduce some additional fields, i.e., selected channel, and data transfer time.
- ATS (Announce To Send): is used to inform neighbors of the sender/receiver of the ongoing transmission and the data channel selection. It also acts as a jamming signal or a busy tone. Besides the standard RTS/CTS fields, some additional fields of ATS are: selected channel and data transfer time.

B. Collision free conditions for AM-MAC

Figure 2 illustrates possible cases of hidden terminal involving the source-destination pair S and R. Node A represents any neighbor of S that is hidden from R while node B represents any neighbor of A hidden from S but can cause collision at A and prevents A from following the conversation between S and R. Similarly, node X is a neighbor of R but is hidden from S and may cause collision at R. Node Y represents any neighbor of X that is hidden from R and can prevent X from following the conversation between S and R

Let us also define the following notations:

- The maximum end-to-end propagation time in the channel is $\tau < \infty$.
- The transmission times of the RTS, CTS, and ATS are $\gamma, \gamma^{'}, \gamma^{''}$, respectively; the transmission time of a data packet is δ; the channel switching delay is ε, and $\gamma, \gamma^{'}, \gamma^{''} < \delta < \infty$

Theorem 1: AM-MAC provides correct data channel acquisition in the presence of hidden terminals, provided that $\gamma > \tau$ and $\gamma + 2\tau + \varepsilon < \gamma^{'}, \gamma^{''} < \infty$

Proof: Consider the illustration in Figure 2. For S to send data to R, S must receive a CTS from R confirming the request. Without loss of generality, assume that at time t_0 S sends an RTS to R. Because the channel has minimum propagation delay, any neighbor of S must begin receiving the RTS at time $t_0^A > t_0$. If the RTS arrives at A with no errors, A must back off for a period larger than $2\tau + \gamma^{'} + \gamma''$ after receiving the RTS, or for a total time of $3\tau + \gamma + \gamma^{'} + \gamma^{''}$ after t_0. If the RTS arrives at A in error (e.g., because of possible interference from B), A must also back off for a period larger than $2\tau + \gamma^{'} + \gamma^{''}$ after receiving the RTS. It follows that the RTS sent by S at time t_0 forces any neighbor of S other than R to back off until time $t_1 > t_0 + \gamma + \gamma^{'} + \gamma^{''} + 3\tau$.

If R receives the RTS from S in error, R will simply ignore or drop it. Assume that the RTS is received correctly at time t_2. R begins to reply with a CTS to S at time $t_2 \leq t_0 + \gamma + \tau$. Within one propagation delay, S receives the CTS from R at time $t_3 \leq t_2 + \gamma^{'} + \tau = t_0 + \gamma + \gamma^{'} + 2\tau$.

As for the receiver side, any neighbor of R must begin receiving CTS at time $t_2^X \leq t_2 + \gamma^{'} + \tau$. If CTS from R arrives at X with no errors, X then must back off for a period of time greater than $\tau + \gamma^{''}$ after t_2^X. If R's CTS arrives at X in error, X must also back off for a period greater than $\tau + \gamma^{''}$ after t_2^X. It follows that CTS sent by R at time t_2 forces X and any neighbor of R other than S to back off until $t_4 > t_2^X + \tau + \gamma^{''} = t_0 + \gamma + \gamma^{'} + \gamma^{''} + 3\tau$.

S begins its ATS broadcast at time $t_3 \leq t_2 + \gamma^{'} + \tau \leq t_0 + \gamma + \gamma^{'} + 2\tau$. Similarly, R begins its ATS broadcast at time $t_2^X \leq t_2 + \gamma^{'} + \tau \leq t_0 + \gamma + \gamma^{'} + 2\tau$. Because $t_1 > t_3$ and $t_4 > t_2^X$, any potential interfering neighbors of S must back off long enough to allow the transmission of ATS packets. Otherwise, ATS transmissions will jam any potential interfering neighbors and let them know of the channel selection since ATS packet is longer than RTS packet.

Neighbors should hear at least a CTS packet or a ATS packet. This is important because ATS and CTS carry the data channel selection information. Thus, any neighbor in the vicinity of S and R must have heard either CTS or ATS because they are both longer than RTS. ∎

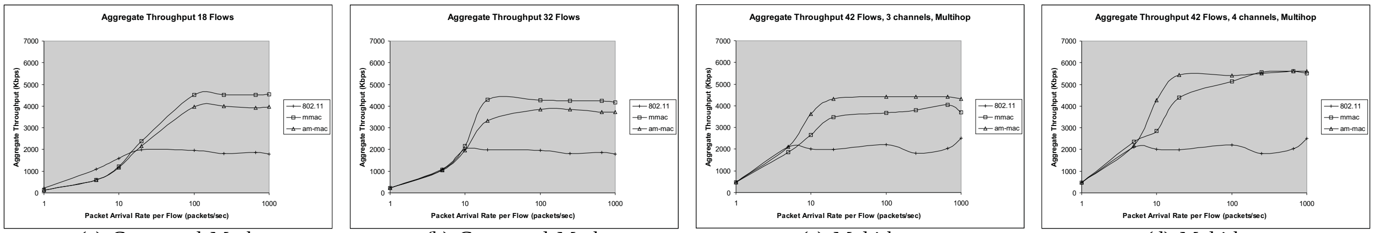
Having complete knowledge of channel selection is crucial to provide a collision free protocol. Nodes which just completed data transmission will often have no knowledge of the channel status. Thus, nodes must observe for the maximum data transmission time before initiating any request for transmission.

Let T_0 be be the observation period before a node initiates a request in the common channel and T_{MAX} be the maximum data access allowed.

Theorem 2: Assuming a single collision domain network. If $T_0 \geq T_{MAX}$,

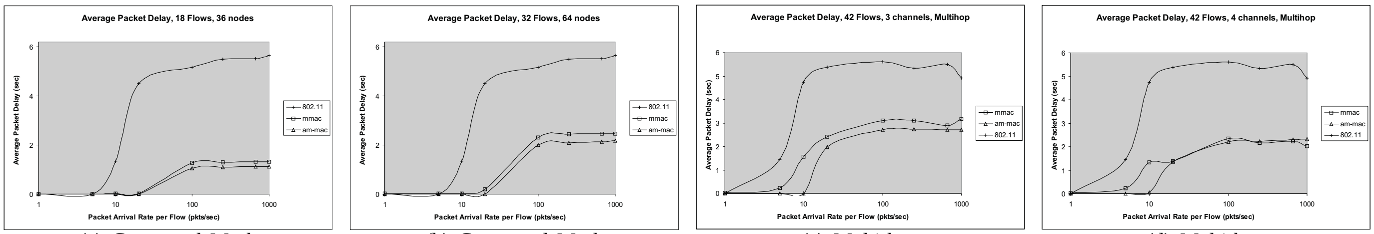
Then AM-MAC is collision free.

Proof: Consider a node A just coming back from the data channel, and a node B switching to the data channel simultaneously. Obviously, A and B are not aware of each others' activities. If node A has observed the common channel for T_0 observation period, node A will have complete knowledge of the channel selected by B since B's data access is upper bounded by T_0. If there is a collision, two things can happen. Either node A did not observe for a sufficient time T_0 or node B held onto the data channel for period of $T_B > T_0$. This contradict our assumptions that $T_B \leq T_{MAX} \leq T_0$. ∎



(a) Connected Mesh (b) Connected Mesh (c) Multi-hop (d) Multi-hop

Fig. 3: Aggregate Throughput vs Packet Arrival Rate.



(a) Connected Mesh (b) Connected Mesh (c) Multi-hop (d) Multi-hop

Fig. 4: Average Packet Delay vs Packet Arrival Rate.

C. AM-MAC Operation

Having described all the basic mechanisms that form the core of AM-MAC, its complete operation is presented here. Suppose A has data destined for B and A gets access to the common channel, the procedures used by A and B is described as follows:

1) A transmits RTS to B, assuming A had already listened in the common channel for the required observation time T_0.
2) B replies by sending a CTS to A. B starts a timer for the CTS so that upon its expiration, it sends the ATS.
3) On receiving the CTS, A updates its channel list and sends ATS packet .
4) After the transmission of the ATS packets, nodes switch to the negotiated channel.
5) A begins sending data to B.
6) After data transmission has been completed, A and B will go back to observing the channel for a full waiting period T_0 before transmitting or responding to a request.

As for any neighbors of A or B, they will observe the control channel and update their channel usage list accordingly.

IV. Performance Evaluation

We evaluate AM-MAC's performance through extensive simulations using the ns-2[14] network simulator with CMU's wireless extension[11]. We evaluate AM-MAC against IEEE 802.11 and MMAC. As performance metrics, we used aggregate throughput and average packet delay.

Throughput is calculated as the number of successful packets received multiply by the packet length and divided by the total simulation time. Packet delay is calculated as the difference between the time a packet arrives at the queue and the time a packet gets transmitted successfully.

For our simulations, we used two network scenarios, namely a fully connected mesh and multi-hop topologies. In the fully connected mesh scenario, every node is within range of one another and is able to reach any destination in one hop. In the multi-hop scenarios, a packet may travel several hops before it reaches the destination.

A. Simulation Setup

Our simulation setup is based on MMAC's setup model because we consider our protocol a variant of MMAC protocol. The difference is that each node follows its own observation phase instead having a single control phase for all nodes. In our simulations, the bit rate for each channel is 3Mbps and the transmission range is set to 250m. Each source generates and transmits constant-bit rate traffic. We use 3 or 4 available channels depending on the scenarios, packet size of 512 bytes, drop-tail queues with maximum queue length of 50 packets, omni-directional antenna, and TwoRayGround propagation model. Each data point in the graph is an average of 10 different simulation runs.

In the fully connected mesh scenario, every node is within each other's range. We simulate two scenarios, one with 36 nodes and 18 concurrent flows and another with 64 nodes and 32 concurrent flows, in a 400x400m area. Nodes move according to the random way-point model with speeds varying between 0 and 10m/s and with no pause. In each scenario, approximately half of the nodes are sources and half are destinations. All nodes begin transmission at the same time.

In the multi-hop network scenario, 121 nodes are placed randomly in a 1000x1000m area. Sources and destinations are randomly selected such that a node may be the source for multiple destinations and a node maybe a destination for multiple sources. At any given point in time, 42 concurrent flows are active. Nodes move according to the random way-point model with speeds varying between 0 and 10m/s with no pause time. For these simulations, we use 3 and 4 available channels.

B. Simulation Results

Figure 3(a) shows the aggregate throughput for the fully connected mesh scenarios. When the network load is low, all protocols have similar performance. When the network approaches saturation, AM-MAC performs significantly better than 802.11 and yields a slightly lower performance compared to MMAC. Figure 3(b) shows the result for the same mesh scenario but with great node density(64 nodes) and more flows(32 concurrent flows). The gain in performance over IEEE 802.11 is mainly due to multi-channel transmission vs a single channel transmission. We observe that, in the fully-connected mesh environments, AM-MAC performs slightly lower than MMAC. This is because when a pair of nodes is negotiating for a channel, all nodes in the networks must delay their requests for data transmissions anyway. As a result, perfect synchronization seems to give MMAC a slight advantage in this environment.

Figure 3(c) and Figure 3(d) show the simulation results for the multi-hop scenario. With 3 available channels, we vary the constant-bit rate traffic from low to very high. For low traffic loads, all protocols have similar performance. However, under moderate to high traffic load conditions, AM-MAC performs better than 802.11 and slightly better than MMAC in terms of aggregate throughput. We repeat the same experiment with 4 available channels, and notice a similar performance trend.

Figure 4 shows the average packet delay in the fully connected mesh and multi-hop environment. When the number of contending nodes is moderate, AM-MAC has significantly better average packet delay than 802.11 and has similar performance to MMAC. The same result applies when we increase the number of contending nodes, flows, and packet arrival rate.

In the fully connected mesh scenario, AM-MAC achieves a slightly slower throughput than MMAC's but gains an advantage in terms of average packet delay. We conjecture that the perfect synchronization gives MMAC an advantage in throughput but suffers in average packet delay because of the control phase wait time. As a result, on average, AM-MAC has better average packet delay than MMAC. However, in the multi-hop scenario, our protocol AM-MAC achieves better or comparable performance in both throughput and average packet delay.

V. Approximate Throughput Analysis

In this section, we present the approximate throughput analysis for our protocol using Markov chain. To make the analysis tractable, we made the following assumptions about the network.

- A finite population of N nodes with identical traffic loads among nodes.
- Arrival of RTS is Poisson distributed with λ.
- The network is fully connected and each node has the same number of neighbors.
- Destination for each packet is chosen from a random uniform distribution.
- Packet lengths are independent and geometrically distributed with probability q.
- The arrival of RTS packets and the completion of data occurs as a single event.

Let k be the number of channels currently in use. Then, we have a total of $2k$ nodes in concurrent transmissions and $(N-2k)$ idle nodes. The value of k is bounded by 0 and $\lfloor \frac{N}{2} \rfloor$

The arrival of the RTS packet at any given state k is given by

$$\lambda_k = \lambda(N - 2k) \tag{1}$$

We are interested in calculating the probability that an RTS packet will become successful at any given state k in our Markov process. This is simply equal to the probability that no other RTS arrives during the vulnerability period and the intended destination node is currently listening on the common channel. For networks using carrier sensing with propagation delay τ, the probability of successful RTS packet at any given state k is:

$$P_{S_k} = e^{-(\lambda_k)\tau} \left(\frac{(N-1) - 2k}{(N-1)} \right) \tag{2}$$

Thus, the probability that the arrival RTS packets will be successful and result in a transition to the next state is:

$$P_{T_k} = \lambda_k \cdot P_{S_k} \tag{3}$$

Then, transition probability for our Markov model can be expressed as follows:

$$P_{ij} = \begin{cases} 0, & \text{if } j > i+1; \\ q^0(1-q)^i P_{T_i}, & \text{if } j = i+1; \\ \\ q^{i-j+1}(1-q)^{j-1} P_{T_i} \\ + q^{i-j}(1-q)^j(1-P_{T_i}), & \text{if } 0 < j \le i; \end{cases} \tag{4}$$

The first condition makes sure that no more than one agreement can be made at any given time in the common channel. The second condition shows that a new data agreement has been made. It is the probability that none of the current data transmissions is completed, and there is a new data transmission agreement. The third condition captures the probability that the Markov state k may remain in the same state or go back to other previous states. This is equivalent to no new data agreements or some data transmissions in other channels are completed.

A system of k equations and k unknowns can be easily constructed and solved, knowing $\sum_{i=0}^{k} \pi_i = 1$. The average data utilization is the average number of channels sending data:

$$E[k] = \sum_{i=0}^{k} i \cdot \pi_i \tag{5}$$

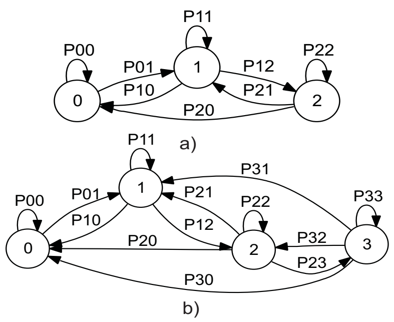


Fig. 5: The Markov chain state for (a)3 channels (b)4 channels. PXY denotes the transition probability P from X to Y.

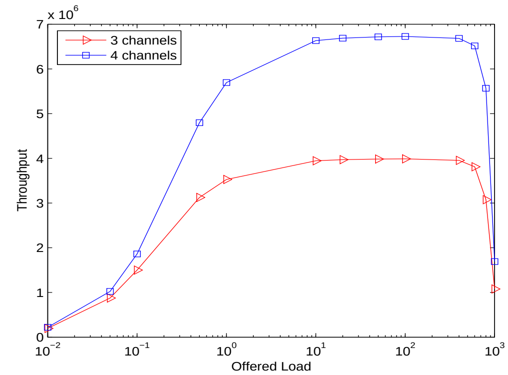


Fig. 6: Analytical Results

Since we assume that the traffic is uniformly distributed across all stations, the average capacity per node is simply $E[k]/N$. And the average utilization per channel is $E[k]/k$.

As for a fixed packet length, we use the well-known birth and death Markov process to simplify our Markov model above. Basically, completion of data transmission is sequential, not simultaneous. At any given instant, only one data packet may depart. We assume the data service time has the inter-arrival times of Poisson arrivals, or exponentially distributed.

The rate of arrival of successful RTS packets that results in the transition forward to a new state is simply P_{T_k}, see (3).

The rate of departure for data packets at a given state(1/average service time)

$$\mu_k = k \cdot 1/\delta$$

Let us define the ratio of arrival and departure to be $r_k = P_{T_{k-1}}/\mu_k$ and $R_k = r_k r_{k-1} \cdots r_1$ and $R_0 = 1$. Then π_0 can easily be found from:

$$1 = \left(\sum_{j=0}^{k} R_j \right) \pi_0$$

Then, the stationary probability [7] is given by

$$\pi_i = \frac{R_i}{\sum_{i=0}^{k} R_j} \tag{6}$$

Having found the probability of being in a particular state of the Markov process, we can proceed to find E[k] and throughput of the system. The analytical results for a simplified version of AM-MAC are shown in Figure 6 with 3Mbps per-channel capacity and propagation delay set to 1/1000 of the average data packet length.

VI. Conclusion

In this paper we presented AM-MAC a novel solution to multi-channel medium access for single-transceiver nodes. AM-MAC employs a simple, yet efficient approach to collision-free data transmission over multiple channels without the need of temporal synchronization among nodes. Our simulation results show that AM-MAC significantly improves performance when compared against IEEE 802.11, and yields comparable performance to MMAC [10], which requires tight synchronization.

References

[1] P. Bahl, R. Chandra, and J. Dunagan. SSCH: Slotted Seeded Channel hopping for capacity improvement in ieee 802.11 adhoc wireless networks. In *In Proc. of ACM MobiCom*, 2004.

[2] C. L. Fullmer. Phd thesis: Collision avoidance techniques for packet-radio networks.

[3] J. Garcia-Luna-Aceves, H. R. Sadjadpour, and Z. Wang. Challenges: Towards truly scalable ad hoc networks. In *In Proc. ACM MobiCom*, 2007.

[4] M. Grossglauser and D. Tse. Mobility Increases the Capacity of Ad Hoc Wireless Networks. *IEEE/ACM Transactions on Networking*, 10(4):477–486, August 2002.

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

[6] J. Lee, J. Mo, T. M. Trung, J. Walrand, and H.-S. So. WiFlex: Multi-Channel Cooperative Protocols for Heterogeneous Wireless Devices. In *Wireless Communications and Networking Conferences(WCNC)*, pages 1489–1494, March 31 - April 3 2008.

[7] A. Leon-Garcia. *Probability and Random Processes for Electrical Engineering*. Addison-Wesley Publishing Company.

[8] J. Mo, H.-S. W. So, and J. Walrand. Comparison of multi-channel MAC protocols. In *8-th ACM/IEEE International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, Montreal, Qc., Canada, October 2005.

[9] H.-S. W. So and J. Walrand. Technical report: Design of a multi-channel mac protocol for ad hoc wireless networks.

[10] J. So and N. Vaidya. Multi-channel Mac For Ad Hoc Networks: Handling Multi-channel Hidden Terminals using a single transceiver. In *In Proc. ACM MobiHoc*, 2004.

[11] The CMU Monarch Project. Wireless and mobility extension to ns.

[12] F. A. Tobagi and L. Kleinrock. Packet switching in radio channels: Part II-The hidden terminal problem in carrier sense multiple-access and the busy-tone solution. *IEEE Trans. Commun.*, COM-23:1417–1433, Dec 1975.

[13] A. Tzamaloukas and J.J. Garcia-Luna-Aceves. Channel-Hopping Multiple Access. In *Proc. IEEE ICC*, New Orleans, Louisiana, June 2000.

[14] VINT GROUP. Ucb/lbnl/vint network simulator(version 2).

[15] S. Wu, C. Lin, Y. Tseng, and J. Sheu. A New Multi-Channel MAC Protocol with On-Demand Channel Assignment for Multi-Hop Mobile Ad Hoc Networks. In *In'l Symposium on Parallel Architectures, Algorithms and Netwroks(I-SPAN)*, 2000.