Open access • Proceedings Article • DOI:10.1109/OCEANSAP.2006.4393832

Slotted FAMA: a MAC protocol for underwater acoustic networks — Source link

Marcal Molins, Milica Stojanovic

Institutions: Massachusetts Institute of Technology

Published on: 16 May 2006

Topics: Underwater acoustic communication, Network packet, Node (networking), Wireless ad hoc network and Underwater acoustics

Slotted FAMA: a MAC protocol for underwater acoustic networks

Marçal Molins
Sea Grant College Program
Massachusetts Institute of Technology
Email: marcalmolins@gmail.com

Milica Stojanovic
Sea Grant College Program
Massachusetts Institute of Technology
Email: militsa@mit.edu

***Abstract*— Long propagation delays and low bit rates of underwater acoustic networks make these systems fundamentally different from the packet radio networks. As a consequence, many of the network protocols designed for radio channels are either not applicable, or have extremely low efficiency over underwater acoustic channels. These facts necessitate a dedicated design of protocols for an underwater acoustic network.**

A medium access control (MAC) protocol suitable for an underwater acoustic network is proposed and analyzed. The protocol is based on a channel access discipline called floor acquisition multiple access (FAMA) which combines both carrier sensing (CS) and a dialogue between the source and receiver prior to data transmission. During the initial dialogue, control packets are exchanged between the source node and the intended destination node to avoid multiple transmissions at the same time. Special attention is paid to the networks that are not fully connected, in which nodes can be hidden from each other. The new protocol uses time slotting and is thus called *Slotted FAMA*. Time slotting eliminates the need for excessively long control packets, thus providing savings in energy. Protocol performance in throughput and delay is assessed through simulation of a mobile ad hoc underwater network, showing the existence of optimal power level to be used for a given user density.

I. Introduction

Underwater acoustic (UWA) networks have many characteristics that make them different from packet radio networks. The major distinguishing characteristics of an underwater acoustic channel are its low bandwidth and long propagation delay caused by the low speed of sound [1]. A medium access control (MAC) protocol allows the nodes in a network to share the common broadcast channel. The main task of a MAC protocol is to prevent simultaneous transmissions that lead to packet collisions. Selection of a suitable MAC protocol has a great impact on the system efficiency, and is especially important for channels with low quality and high latency, such as the underwater acoustic channel.

Many MAC protocols have been proposed since the first Aloha [2] protocol. Carrier sensing multiple access (CSMA) [3] and its variations have been widely used to prevent collisions between two or more stations transmitting at the same time. These protocols require the stations to "listen" to the channel before starting to transmit to avoid possible collisions with other ongoing transmissions. CSMA is efficient when used in fully connected networks with propagation delays that are small compared to the packet duration. As the delay increases, the efficiency is rapidly lost. In addition, the problem of *hidden terminals* and *exposed terminals*arises in ad-hoc networks due to the lack of connectivity between certain nodes.

A situation of a *hidden terminal* occurs when one station cannot sense one or more nodes that can interfere with its transmission. A situation of an *exposed terminal* occurs when a station delays transmission because of another overheard transmission that would not collide with it. Figure 1 illustrates a situation in which nodes A and C are hidden from each other. A situation of exposed terminals would occur if C were transmitting to D and B wanted to start a transmission to A. B would listen to the channel and it would defer its transmission although packets from C to D would not collide with packets sent from B to A. CSMA protocols degrade in the presence of such problems, both of which are very likely to occur in UWA networks.

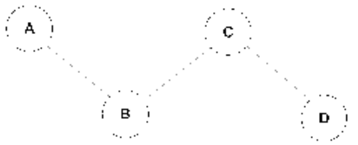


Fig. 1. Network with hidden terminals.

To solve the problems of CSMA, a handshake can be used prior to transmission of data packets. Karn [4] proposed a protocol called MACA (Multiple Access Collision Avoidance). When a node wants to transmit a packet, it sends an RTS (*Request To Send*) control packet addressed to the intended receiver. This receiver terminal responds with a CTS (*Clear To Send*) control packet. This second packet warns all the nodes in its neighborhood (all those whose transmission could collide with the announced one) that a data packet is about to be transmitted, and tells the source node that it has permission to start the transmission. A complete handshake is illustrated in Figure 2

Bharghavan [5] suggested some modifications to the original MACA protocol, which resulted in a new protocol called MACAW (MACA-Wireless). The main features of this protocol are an adaptive backoff algorithm and the inclusion of an

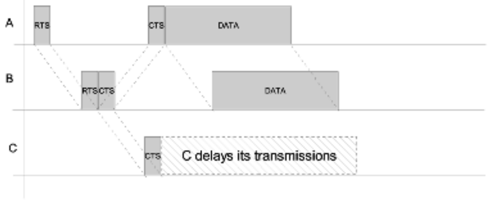


Fig. 2. A handshake between A and B.

ARQ technique that allows retransmission of erroneous DATA packets.

Fullmer and Garcia-Luna-Aceves [6] identified the conditions needed to ensure that data packets in a MACA scheme are sent without collisions. A collision in this protocol may occurs due to different packet delays, e.g., when a terminal has one neighbor very close and another very far. Figure 3 illustrates failure of the MACA protocol to ensure collision avoidance. Node B is close to A and node C is hidden from it. We see how successful handshake between A and B can be completed before C can hear B's CTS, but the data packet is corrupted by C's RTS.

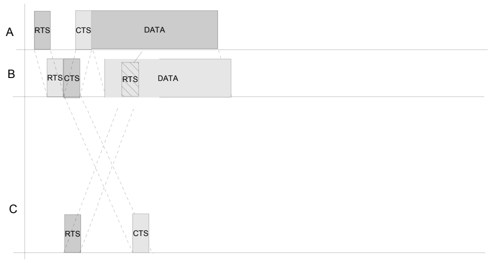


Fig. 3. RTS from C collides with data packet from A.

To overcome this problem, the carrier sensing capability, which had been disabled in MACA and MACAW, was recovered, resulting in the protocol known as FAMA (Floor Acquisition Multiple Access) . It was shown that using a carrier sensing protocol, collision avoidance is guaranteed if the following conditions hold: a) RTS length should be greater than the maximum propagation delay, and b) CTS length should be greater than the RTS length plus twice the maximum propagation delay plus the hardware transmit-to-receive transition time. These conditions are the basis of the FAMA protocol.

Although it ensures the absence of collisions in the channel, the length of control packets becomes excessive on an underwater acoustic channel. The required packet lengths depend directly on the propagation time, which is very high in an underwater acoustic network (a second for 1.5 km) . This leads to an unacceptable waste of energy.

II. Slotted FAMA

As we discussed, using the original FAMA protocol in underwater acoustic networks would not be efficient due to the required length of the RTS and CTS packets. At the same time, violating these conditions would lead to data collisions as shown in Figure 3.

To overcome this problem, a protocol should prevent the nodes from sending packets when data is being sent. In other words, nodes should *know* if sending a packet could collide with a concurrent transmission in its neighborhood. To do that without meeting the conditions of the FAMA protocol, a restriction must be imposed on the times when packets can be sent. This can be accomplished by slotting the time to eliminate the asynchronous nature of the protocols. Each packet (RTS, CTS, DATA or ACK) has to be transmitted at the beginning of one slot. The slot length has to be determined in a manner that ensures absence of data packet collisions. Namely, it has to allow all the nodes to receive the information required, so that they will know whether transmitting at the beginning of the following slot will interfere with an ongoing transmission. This can be achieved with a slot length of $\tau + \gamma$, where τ is the maximum propagation delay and γ is the transmission time of a CTS packet. In this manner it is guaranteed that an RTS or a CTS packet transmitted at the beginning of a slot is received by all the nodes within transmission range over the duration of one slot. An ARQ protocol has also been included by sending ACK or NACK packets to acknowledge the data reception. We will call this new protocol *Slotted FAMA*.

To account for any clock drift that may be present in the system, a guard time can be inserted to slightly increase the slot duration.

A. Algorithm definition

When a node wants to send a packet it waits until the next slot and transmits an RTS packet. This packet is received by the destination node and all the terminals in the neighborhood of the source node within the slot time. The destination node then sends a CTS packet at the beginning of the next slot. This packet is also received within the slot time by the source node and all the terminals in the range of the destination node. When the source terminal has received the CTS it knows that it has permission to transmit, so it waits until the beginning of the next slot and then starts sending the data packet. When the receiver has the entire data packet it sends an ACK packet to indicate that the transmission has been successful. A successful handshake is illustrated in Figure 4.

As the FAMA protocol, slotted FAMA is based on carrier sensing. This means that terminals are constantly listening to the channel. Terminals stay in *Idle* state until they sense the carrier in the channel or until they have a packet ready to transmit. If a packet is ready to be transmitted at the beginning of a slot and no carrier has been detected, terminal sends an RTS and waits two slots (current slot and the next one) to receive a CTS packet. If no CTS is received during this time, a collision is assumed and the terminal goes to *Backoff* state for a random number of slots. After that, the RTS packet is

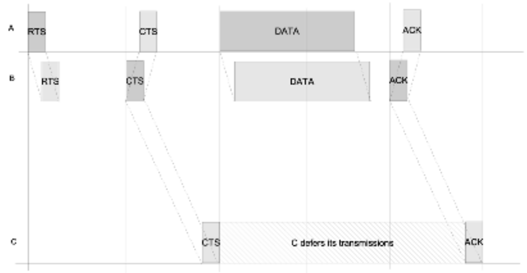


Fig. 4. A successful handshake between terminals A and B in slotted FAMA.

re-sent if no carrier has been sensed during the backoff time. When CTS is successfully received, the terminal will start sending the data packet in the next slot.

When a terminal detects carrier on the channel it goes to a *Receiving* state while it is receiving the packet. The type of packet received will determine the receiver actions as listed below.

- After receiving an RTS packet intended for another station (*xRTS* packet) the terminal must wait two slots (long enough for the receiver to send a CTS and the sender to start transmitting data). If after this time no carrier is sensed, the terminal returns to the Idle state. This two slot wait is necessary because of the ARQ protocol. It re-introduces the problem of the *exposed terminal*, which will be discussed later.
- After receiving a CTS packet intended for another station (*xCTS* packet) a terminal must wait long enough to allow the other station to transmit the entire data packet and receive the corresponding ACK. Since the terminal has received the CTS packet, it will also receive the ACK packet and will thus know that data transmission has ended successfully.
- After receiving a Data packet intended for another station (*xDATA* packet) a terminal must wait long enough to allow the reception of the subsequent ACK or NACK packet. Since it is possible that the terminal cannot hear the ACK or NACK packet, it must wait an additional slot to detect whether the data packet has been re-sent (meaning that a NACK was sent) or not.
- After hearing an ACK packet intended for another station (*xACK* packet) a terminal only has to wait until the end of the slot since the data transmission has successfully ended.
- After hearing a NACK packet intended for another station (*xNACK* packet) a terminal must wait long enough to allow for a complete data packet to be transmitted and a new ACK or NACK to be sent (the same time as if it had received an xCTS packet).
- If a terminal senses interference in the channel, a collision is assumed. Since it doesn't know which packets have collided, the worst assumption is made, and it acts as if it had received an xCTS packet which is the situation that requires a longer wait.

B. ARQ and the exposed terminal problem

A simple ARQ has been added to the original FAMA protocol to improve its performance in high BER scenarios. After a packet has been received, it is checked for errors using a standard cyclic redundancy check (CRC) procedure, and an ACK is transmitted if the data packet is found correct. Otherwise, a NACK packet is sent and the source node re-transmits the data packet.

The inclusion of an ARQ protocol into FAMA, re-introduces the exposed terminal problem. If we look back at Figure 1, an exposed node situation occurs if C defers the transmission of a packet to D because it hears that B is already sending a packet to A. Transmissions from C would not collide with transmissions directed to A since they are hidden from each other. However, if B is to receive an ACK packet from A, then C should not transmit, and must wait for the entire transaction to be completed between A and B.

When no ACK/NACK packets are used, C must wait only one slot after hearing an xRTS packet and thus allowing just the subsequent CTS packet to arrive. After one slot without hearing this xCTS packet, C would assume that the receiver is out of its range and that a transmission can begin without causing any collision.

The inclusion of the ACK/NACK packets causes the terminal C to wait until the xDATA packet is sent because the sender B (which is in its range) has to receive this ACK or NACK packet from A. A transmission, even without colliding with the data packet at the receiving node C, could collide with the ACK packet at the source node B. Thus, there is a trade-off between using this ARQ protocol and a different one in an upper layer (e.g., an end-to-end retransmission protocol). For a channel with multiple hops and high BER, a data-link ARQ is more suitable than and end-to-end one which would cause a large reduction in the throughput and an increase in delay.

C. Backoff algorithm

In the original FAMA protocol, when a station does not receive a CTS in response to its previously sent RTS packet, it goes to the Backoff state. The backoff time is set randomly between a minimum and a maximum time, and after the backoff, the RTS packet is re-sent. If carrier is sensed on the channel while a station is in the Backoff state, the terminal goes to the Receiving state and performs all the operations that the received packet requires. Once all these operations are performed, the station returns to the Backoff state and re-sets a new backoff time.

In high traffic environments (i.e. a node with lots of neighbors) the time to acquire the channel is very long when the node is entering the Backoff state. This is caused by the fact that when a node leaves the Backoff state due to an incoming packet, it resets the backoff time. Hence, if the backoff time is high and the terminal hears petitions very often, it is likely to detect carrier on the channel within the backoff time, thus having to restart the Backoff procedure over and over. To alleviate this problem, the backoff time could be reduced, but this would involve another problem: with reduced backoff times, the probability of two or more neighbors choosing the same time, and thus re-sending their RTS at the same instant, would increase highly.

The proposed solution is to avoid the resetting of the backoff time every time the terminal is returning to the Backoff state. This means that the backoff time is chosen when there is no response to a sent RTS. At this time, the timer is set with the random backoff time. If during this backoff time, the station has to leave the Backoff state due to an incoming transmission, the timer will continue running. When the station returns to the Backoff state, the backoff time is not reset. When the timer expires, RTS packet is resent. This allows the protocol to improve the performance in high traffic situations without having to decrease the backoff time.

D. *Transmission priority*

To improve the circulation of packets through the network and the fairness of the protocol, a transmission priority has been given to the nodes that have just received a packet. Usually, when a node hears a transmission it waits to prevent a collision and, after that, if it has a packet ready to transmit it goes to the Backoff state. This is done to avoid many nodes sending an RTS packet at the same time after the end of a transmission.

Now, after a node receives a packet, if it has any packet ready to transmit, it will proceed to send the RTS without going to the Backoff state. Thus, packets that involve several hops in the network will travel more *smoothly* through it. It also aids nodes with a high number of incoming packets to transmit these incoming packets to the next hop with higher priority.

E. *Trains of packets*

To increase the efficiency of the protocol, the use of trains of packets is considered. Each station has a local queue (that we will suppose to be infinite) where it stores the packets waiting to be sent. The priority is given to the oldest packet in the queue thus favoring a lower end-to-end delay. When a station establishes communication with another station, it will send the packet it was trying to send plus all the packets in its queue that have to be sent to the same station. Thus, multiple packets are sent with a unique handshake. In each packet sent, a flag will tell the receiving node if the sender is transmitting more packets in the same train.

Using this modification, one could also change the ARQ protocol acknowledging all the packets at the end of the train, as proposed by Morris in [7]. Then the transmitting node would only resend the erroneous packets. This would increase the efficiency of the protocol, but the main problem is that FAMA (and Slotted FAMA too) needs to set a maximum packet time (maximum transmission time of a DATA packet) because this is the time that stations overhearing a CTS packet have to wait in order to avoid collisions with a concurrent transmission. So the inclusion of DATA packets within a train have to be acknowledged one at a time. The ACK/NACK packet sent will also have the same flag indicating that more packets are going to be received. A station receiving an ACK packet with this flag set to '1' will have to defer its transmissions as if it had heard a CTS packet.

F. *Slot Time and Packet Priority*

If the slot length is defined to be much longer than the RTS/CTS packet length (and this may be the case in an underwater channel with long delays) there is a possibility that more than one complete control packet is successfully received within one time slot at one terminal.

A convenient order of priorities has to be set within all packet types to avoid possible collisions and obtain a better throughput. Since data reliability is our principal goal, CTS and xCTS (CTS's which are meant for another node) packets must have the highest priority because they are setting the beginning of a data packet transmission. From this point of view, packets involving remote transmissions should have priority to favor fairness of the protocol.

III. Throughput Analysis

Let us assume the network layout shown in Figure 5. In this figure, node ω has a total of N neighbors (in this case N=6, numbered 1 through 6 in the figure). Each of them has Q neighbors which are hidden from ω (in the figure, neighborhood of node 1 is shown, light gray nodes are hidden from ω, so Q=3). Each node has a packet ready to send every $1/\lambda$ seconds on the average (the arrivals follow a Poisson distribution with average λ packets per second, i.e. exponentially distributed inter-arrival time). The packets are distributed evenly among the neighbors, i.e. λ/N directed to each of the neighbor nodes.

Throughput per node (S) can be defined as:

$$S = \frac{\bar{U}}{\bar{B} + \bar{I}} \tag{1}$$

where $\bar{U}$ is the average time while useful data is being sent, $\bar{B}$ is the average time while channel is being used (busy period) and $\bar{I}$ is the average time between two busy periods (Idle time).

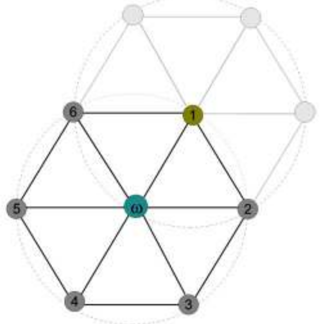


Fig. 5. Network layout.

Let us define P_s as the probability of success (no collisions) on the channel. The probability of no collisions is the probability that no neighbors transmit within a time slot used by a given node ω. These transmissions can be the neighbors' RTS's, or CTS's whose corresponding RTS's have not been heard. This second situation would correspond to RTS's sent by nodes hidden from ω to one of ω's neighbors during the previous slot. For example, if ω sends an RTS during slot *n* and one of the light gray nodes has successfully sent an RTS to node 1 during slot *n-1*, ω's RTS and 1's CTS will collide. For each ω's neighbor, the number of neighbors hidden from it (light gray nodes for node 1) equals Q. Since each of these Q nodes send RTS's to each ω's neighbors at a rate λ/N the probability of no collisions is:

$$P_s = \prod_1^N e^{-\lambda T_{slot}} \cdot \prod_1^N \left(\prod_1^Q e^{-\frac{\lambda}{N} Tslot} \right) = e^{-\lambda(N+Q)T_{slot}}$$

Actually, this is a lower bound on the probability of no collisions, since we are assuming that the RTS sent by a node hidden from ω has not collided and that a CTS will be sent.

Given the BER, the probability of error in a data packet containing L bits, assuming independent errors, is:

$$P_e = 1 - (1 - BER)^L \approx L \cdot BER$$

A busy period can be the time during which data is being successfully sent ($\bar{T}_{success}$), a period of collisions on the channel ($\bar{T}_{fail}$), or the time during which we cannot transmit due to transmissions from other nodes ($\bar{T}_{defer}$).

The duration of a failed period is two slots (the slot in which the RTS packet is sent, and the next slot spent in waiting for the CTS that won't arrive). The probability that a given RTS was transmitted by ω is $\frac{1}{N+1}$ because all the $N+1$ nodes transmit at the same rate, so:

$$\bar{T}_{fail} = \frac{2T_{slot} \cdot (1 - P_s)}{N+1}$$

The duration of a successful period includes the RTS, CTS, DATA (plus all the retransmission due to errors in the packet), all the sent NACKs (if there's any one) plus the final ACK. RTS, CTS and ACKs/NACKs only need one slot to be transmitted. A data packet needs more slots; let us call T_{data} the duration of all the slots needed by a DATA packet. If by T we denote the time between the start of a DATA packet and the time of successful reception of the ACK packet, then:

$$T = \sum_{n=1}^{\infty} n(T_{data} + T_{slot}) \cdot P_e^{n-1}(1 - P_e) = \frac{T_{data} + T_{slot}}{1 - P_e}$$

The duration of a successful transmission is now RTS plus CTS plus T, i.e., $T_{Tot} = 2T_{slot} + T = 2T_{slot} + \frac{T_{data}+T_{slot}}{1-P_e}$. Hence,

$$\bar{T}_{success} = P_s \cdot T_{Tot}$$

Deferral periods are those while ω is deferring its transmissions because the channel has been acquired by another node or because it has heard interference (collision) on the channel. The first situation will occur when ω overhears a CTS from one of its neighbors. The second situation will occur when there's a collision overheard by ω.

The probability of overhearing a CTS is:

$$Prob(CTSoverheard) = \frac{Q\frac{\lambda}{N}NP_s}{(N+1)\lambda} = \frac{Q}{N+1}P_s$$

In this situation, the deferral time equals $T = \frac{T_{data}+T_{slot}}{1-P_e}$ The probability of hearing noise in the channel is:

$$Prob(collision) = \frac{N}{N+1}(1 - P_s)$$

In this situation, the deferral time equals $T_{data} + T_{slot}$, so the average deferral time is:

$$\bar{T}_{defer} = (T_{data}+T_{slot}) \left(\frac{QP_s}{(N+1)(1-P_e)} + \frac{N}{N+1}(1 - P_s) \right)$$

The average idle time on the channel is:

$$\bar{I} = \frac{1}{(N+1)\lambda}$$

Denoting by δ the transmission time of DATA packet, the average time during which useful data is sent from ω is obtained as

$$\bar{U} = \frac{\delta}{N+1}P_s$$

Linking all the parts, we obtain the final result which is given in Equation (2). This equation measures what we will call *Throughput per node* defined as the fraction of time during which a certain node is transmitting correct data. This equation is valid for a static single-hop network.

IV. Simulation results

To asses the performance of a dynamic network consisting of autonomous underwater vehicles (AUVs), a simulation analysis has been carried out. A complete network simulator has been implemented within the Matlab/Simulink environment. The simulated 25 km^2 network area is divided into 16 cells and one AUV is placed in a random manner in each cell as shown in Figure 6. Each AUV acts independently from the others and sends petitions to the channel following a Poisson distribution with an average of 1 packet per 300 seconds. AUV's move at a constant velocity of 5 knots (2.5m/s). An AUV moves within its cell changing the direction in random intervals of time uniformly distributed between 1 and 5 minutes. Every so often, a node may decide to go outside of its cell, move into a random location within the network for a random time between 5 and 10 minutes, and return back to where it was.

The DATA packet size is set to 3000 bits, and all the other control packets are 100 bits long. Bit rate is set to 1000 bits per second.

Since routing algorithm and network maintenance have not been taken into consideration, some assumptions have to be made. The optimal routing path is chosen as the one with the fewest number of hops. Since nodes are moving, network maintenance information is assumed to be conveyed within the DATA packets. Hence, routing tables are updated every time a node receives, or overhears, a DATA packet.

$$S = \frac{\delta P_s}{(N+1)P_sT_{Tot} + 2T_{slot}(1-P_s) + (T_{data}+T_{slot})(Q\frac{P_s}{1-P_e} + N(1-P_s)) + \frac{1}{\lambda}} \tag{2}$$

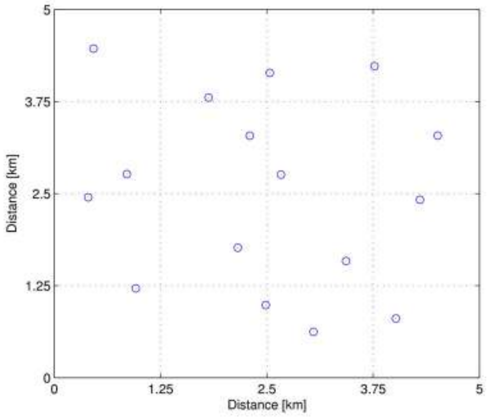


Fig. 6. Example network.

A. Transmission range

The example network has been simulated for different transmission ranges from 1.5 km to 3 km. Low transmission ranges involve low competition to acquire the channel, but a higher number of hops through the network, and lower connectivity. This means that for low transmission ranges many nodes can get disconnected from the network. A situation of isolated nodes occurs when one or more pairs of nodes cannot be connected through any path. On the other hand, high transmission ranges, although achieving a high connectivity and a lower number of hops, will involve an increase of the traffic that a node hears, due to the increase in the number of neighbors that it has. An excessive connectivity will make it difficult to acquire the channel due to a large amount of RTS collisions. In Figure 7 the relation between the transmission range and the average number of neighboring nodes is plotted.

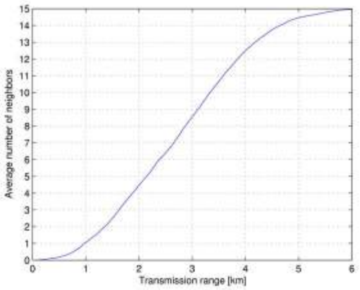
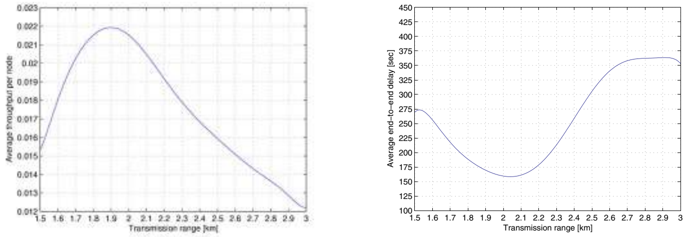


Fig. 7. Number of neighboring nodes vs Transmission Range.

The throughput per node was previously defined as the percentage of time that a single node is successfully transmitting data. Figure 8(a) shows that a maximum throughput is achieved at a certain transmission range, around 2 km for the particular example considered. For lower transmission ranges, the greater number of hops and the lower connectivity degrade the performance. For higher transmission ranges, the degradation is caused by the increase of neighbors which involve an increase of the *overheard traffic*.

End-to-end delay is defined as the time tht elapses since a packet is generated until it is successfully received by the destination node. Figure 8(b) shows the end-to-end delay, indicating the existence of a transmission ranges (also about 2 km) at which the lowest end-to-end delay is achieved.



(a) Average Throughput per Node

(b) Average end-to-end delay

Fig. 8. Performance as a function of the transmission range: throughput and delay.

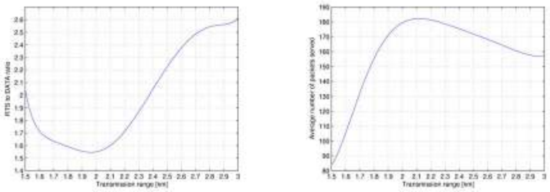
Another measure that was used to evaluate the performance of the protocol is called *RTS to DATA ratio*. This is the ratio between the RTS and DATA packets sent. This ratio gives us a measure of the average number of RTS packets sent for each DATA packet. It shows the degree of difficulty with which the nodes acquire the channel. Figure 9(a) clearly shows how this ratio increases with the transmission range. The increase in the number of neighbors makes the acquisition of the channel more difficult. For low transmission ranges, the RTS/DATA ratio also increases due to the isolation of the nodes (nodes send RTS's that nobody hears).

Figure 9(b) shows that the total number of packets served in one hour is also greatest for the optimal transmission range. Lower values imply a large number of packets in the network that have not yet arrived to their destination. This effect causes a saturation of the network (new packets are created while old packets have not been served) which may lead to an unstable behavior.

Looking at these results it is clearly seen that there exists an optimal transmission range which overcomes the problems of isolation (low transmission ranges) and excessive traffic (high transmission ranges).

V. Conclusion

In this paper, a new MAC protocol for UWA networks has been presented and studied. *Slotted FAMA* avoids DATA packet collisions without requirements on the packet size.



(a) RTS to DATA ratio

(b) Total packets served in one hour

Fig. 9. Performance as a function of the transmission range: RTS to DATA ratio and total packets served

Simulation results show that for a given network topology, the transmission range can be chosen so as to maximize the network performance in terms of throughput and delay.

Future work should concentrate on the development of algorithms for network maintenance and for updating the routing tables of the mobile nodes. Another point that the future work should address is the backoff algorithm. This first version of Slotted FAMA uses a simple backoff procedure. The use of adaptive backoff times (as used in MACAW) should be studied to provide a better performance of the protocol and to favor fairness, mainly in high traffic situations.

In long slot situations, the use of simultaneous bidirectional communications should be studied. This means that in situations where the propagation delay is high, transmitter and receiver could send a packet to each other simultaneously without leading to a collision. This could be used to acknowledge trains of packets, because they have to be acknowledged one by one.

Power control is another important issue that should be studied. We have shown the existence of optimal transmission power, which in practice must be achieved through adaptive power control.

The use of Slotted FAMA in radio networks could also be a research point, specially in networks where the propagation delays are not negligible.

References

[1] L. Berkhovskikh and Y. Lysanov, *Fundamentals of Ocean Acoustics*, 1982.

[2] N. Abramson, "The aloha system," in *Computer-Communication Networks*, N. Abramson and F. Kuo, Eds. Englewood Cliffs, New Jersey: Prentice-Hall, 1973, pp. 501–518.

[3] L. Kleinrock and F. A. Tobagi, "Packet switching in radio channels: part I carrier sense multiple-access modes and their throughput-delay characteristics," *IEEE trans. on commun.*, vol. COM-23, pp. 1400–1416, 1975.

[4] P. Karn, "MACA - A new channel access method for packet radio," *ARRL/CRRL Amateur Radio 9th Computer Networking Conference*, 1990.

[5] V. Bharghavan, A. Demers, S. Shenker, and L. Zhang, "MACAW: A media access protocol for wireless LANs," in *Proceedings, 1994 SIGCOMM Conference*, London, UK, Aug. 31st - Sept. 2nd 1994, pp. 212–225.

[6] C. L. Fullmer and J. J. Garcia-Luna-Aceves, "Floor acquisition multiple access (FAMA) for packet-radio networks," June 07 1995. [Online]. Available: http://citeseer.ist.psu.edu/180615.html; ftp://ftp.cse.ucsc.edu/pub/reinas/papers/chane.sigcomm.ps.gz

[7] J. M. Morris, "Optimal blocklengths for ARQ error control schemes," *IEEE trans. on commun.*, vol. COM-27, pp. 488–493, 1979.

[8] C. L. Fullmer and J. J. Garcia-Luna-Aceves, "Performance of floor acquisition multiple access in ad-hoc networks," in *SCC '98. Proceedings. Third IEEE Symposium on Computers and Communications*, June 30th - July 2nd 1998, pp. 63–68.