

# NANOMAC: A DISTRIBUTED MAC PROTOCOL FOR WIRELESS AD HOC SENSOR NETWORKS

Jussi Haapola  
Centre for Wireless Communications  
P.O. Box 4500, 90014 University of Oulu, Finland  
jhaapola@ee.oulu.fi

## ABSTRACT

Sensor networks have ad hoc topologies with low, but highly variable traffic rates. Because devices are often battery operated, power consumption becomes an important issue. Minimising the power consumption of the electronics alone is not enough, for especially media access control (MAC) protocols have to be optimised. The availability of different MAC protocols is good, but efficient ones suitable for low bit-rate, low-power devices are rare. An efficient MAC protocol has a strong effect on power consumption because it reduces retransmissions and makes better use of the channel. In this paper, a carrier sense multiple access with collision avoidance (CSMA/CA) based MAC protocol, called nanoMAC, suitable for low bit-rate, low-power wireless devices with high efficiency is proposed, a general analysis is given, and results are compared with other MAC protocols found in the literature.

## I. INTRODUCTION

To allow for easy distribution and relocation, wireless sensors and systems are often battery operated. However, battery power causes some nontrivial complications as sensor devices are small, which leaves limited space for a battery. For a long period of operation very low power consumption is necessary. The devices may also be in a place where changing the battery constantly is impossible. These requirements call for an efficient wireless sensor solution with low power consumption. Minimising the power consumption of the components of the sensor is not enough as also the MAC protocol has to work efficiently to optimise the amount of bits sent. As the traffic offered to the network can be highly dynamic, the MAC

protocol must be stable even with high loads, therefore ALOHA-like throughput and stability [1] is not acceptable.

Section II briefly reviews pure ALOHA, nonpersistent CSMA, and multiple access with collision avoidance (MACA) protocols as a comparison for the nanoMAC. The nanoMAC analysis is presented in section III, and its throughput and delay results are presented in section IV. Conclusions are drawn in section V.

## II. DISTRIBUTED MAC PROTOCOLS

We consider a single-hop ad hoc system with an infinite population of devices where the frame generation is assumed to be a Poisson process with rate  $g$  frames/sec. The channel is considered error-free without Capture. Since the population is infinite, each frame can be considered to belong to a different user. Therefore, every new frame can be assigned to an idle device, which does not have a frame to retransmit. The throughput,  $S$ , is defined as the average number of successful frame transmissions per time interval  $T_p$ . We let  $G$  be the traffic intensity or normalised traffic offered to the channel.  $G$  is defined as  $G = gT_p$ , and it includes new frames as well as retransmissions of old frames. Throughput,  $S$ , is a function of  $G$ , and from the Poisson distribution, we know that it has the mean of  $G$  frames per frame time, and has the probability,  $P(g) = G^g e^{-G} / g!$ .

In pure ALOHA the method of access is to send a frame, whenever there is data to be sent. With the ALOHA protocol, there will be a lot of collisions because they occur whenever packet transmissions overlap any amount of time. A user waits a length of time equal to the round-trip delay for an acknowledgement (ACK) from the receiver. If no ACK is re-

ceived, the frame is assumed lost in a collision and is retransmitted with a randomly selected delay to avoid repeated collisions.  $S$  is defined as the fraction of time the channel carries useful information, hence [2]  $S = Ge^{-2G}$ .

Nonpersistent CSMA provides a better use of channel resources by listening for a clear channel before attempting transmission, and the method is called carrier sensing (CS). In nonpersistent CSMA, a device with a frame to send, first senses the channel, and sends if the channel is detected vacant. Otherwise, it waits a random interval before repeating the procedure. This mode reduces collision, but collisions still occur, and there might be longer delay times [3]. The throughput of nonpersistent CSMA with ACK on the same channel can be presented as:

$$S = \frac{Ge^{-aG}}{G(1+2a+c) + e^{-aG}}, \quad (1)$$

where  $a$  is a normalised propagation time  $a = \tau/T_p$ ,  $c$  is the normalised ACK delay, and  $\tau$  is the propagation delay. The delay,  $D$ , for nonpersistent CSMA is presented in [3] and when ACK is on the same channel it is:

$$D = \left(\frac{G}{S} - 1\right)(1+2a+T_{CS}+c+\delta)+1+a, \quad (2)$$

where  $T_{CS}$  is time needed for carrier sensing,  $c$  is the acknowledgement frame, and  $\delta$  is the normalised, average retransmission delay.  $S$  can be found from Eq. (1).

Karn [4] first introduced the MACA protocol. The motivation for MACA is that the CS procedure leads into a 'hidden terminal' problem, and the CS provides information about potential collisions at the sender, not at the receiver where it matters. The normalised throughput of MACA is described in [5] and is:

$$S = \frac{1}{e^{G(2b+a)} \left(b+a + \frac{1}{G} + F'\right) + \frac{e^{Gb} \left(b + \frac{a}{2} + P'(a-F')\right) + 1 + \frac{3a}{2} + F' + P'(a-F')}{}, \quad (3)$$

where:

$$F' = \left[ \frac{e^{Gb} - 1 - Gb}{Gb(1 - e^{-Gb})} \right]; P' = \left[ \frac{e^{-Gb} - e^{-G(b+a)}}{1 - e^{-G(b+a)}} \right]. \quad (4)$$

### III. SENSOR MAC ARCHITECTURE

The presented nanoMAC protocol [6] is of CSMA/CA type and is  $p$ -nonpersistent, i.e., with probability  $p$ , the protocol will behave like nonpersistent CSMA. The methods of saving power and resources are implemented on many levels: devices will sleep whenever another device is transferring data, the frame sizes are short to prevent spurious errors, and the overhead is minimised to compensate for small frames. A frame train ideology is introduced which means that several frames are sent with each collision avoidance (CA) reservation and a common ACK frame is received as a reply. So, if only one small frame is corrupted only that has to be retransmitted. Most of the control information is sent in the CA frames, which enables other devices to calculate a proper sleep time and the data frames have only a small overhead. The protocol has to be stable even at high traffic loads.

The theoretical throughput of the nanoMAC is as follows: the fore-mentioned conditions and those described in [3] apply. The MAC's transmission periods are depicted in Fig. 1 and illustrate how two different devices communicate, and how the channel is used.

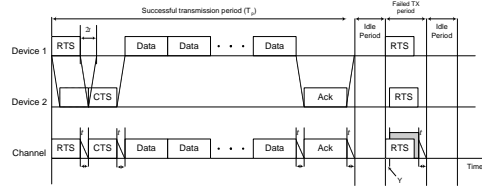


Figure 1: Transmission periods of nanoMAC.

The idle period is the duration between the end of a frame transmission and the arrival of the next frame. The scheduling is memoryless (no retransmissions, if RTS successfully sent), and works as in [3] in  $p$ -persistent CSMA:

$$I = \frac{\tau}{1 - e^{-g\tau}}. \quad (5)$$

The average utilisation  $U$  is the average amount of time that useful data is sent during the course of one successful busy period:  $U = (\gamma + T_p) \times P_{suc}$ , where  $P_{suc}$  denotes the

probability of the transmitted frame being successful. Therefore we get:

$$U = (\gamma + T_p) (1 - P_{ers} + P_{ers}e^{-g\tau}) , \quad (6)$$

where  $\gamma$  is the time an RTS frame takes to transmit. The average transmission period  $B$  can be calculated using Fig. 1 to intuitively count all the different elements in the channel. We get:

$$\tilde{B} = 2\gamma + 4\tau + T_p + \epsilon + \tilde{Y} , \quad (7)$$

where  $\tilde{Y}$  is the probability of no arrivals coming in the interval,  $(\tau - Y)$ , and is  $P[\text{Not}]$  or  $P[\text{Scheduled}]$  and  $P[x > P_{ers}]$ ,  $0 \leq y \leq \tau$ :

$$F_Y(y) = P[\text{Not}] + P[\text{Scheduled}] \times P[x > P_{ers}] \\ = 1 - P_{ers} + P_{ers}e^{-g\tau}e^{gy} . \quad (8)$$

Next, we differentiate and get:

$$F'_Y(y) = P_{ers}e^{-g\tau}\delta(y) + gP_{ers}e^{-g(\tau-y)} , \quad (9)$$

from which we can calculate the expected value of  $\tilde{Y}$  as:

$$E[\tilde{Y}] = P_{ers} \left( \tau - \frac{1 - e^{-g\tau}}{g} \right) . \quad (10)$$

Now, from Eq. (7) we can calculate  $B$ , and finally, we combine Eqs. (5) and (6) with  $B$  and get  $S$ . This is done after normalising  $T_p$  to 1, and introducing the following variables:

$$a = \frac{\tau}{T_p}, \quad b = \frac{\gamma}{T_p}, \quad c = \frac{\epsilon}{T_p}, \quad \text{and } G = gT_p ,$$

And the throughput  $S$ :

$$S = \frac{U}{B + I} \\ = \frac{G(b+1)(1 - P_{ers} + P_{ers}e^{-aG})}{G(1 + (4 + P_{ers})a + 2b + c + \frac{a}{1 - e^{-aG}})} \quad (12) \\ \frac{-P_{ers}(1 - e^{-aG})}{-P_{ers}(1 - e^{-aG})}$$

For the delay of nanoMAC, a delay calculus method presented in [5] has been used. Fig. 2 shows the different delays occurring in a single device, which is a part of a Markov chain. Let the same assumptions as in the calculation of the throughput apply. To determine the average delay, we consider the transitions a device

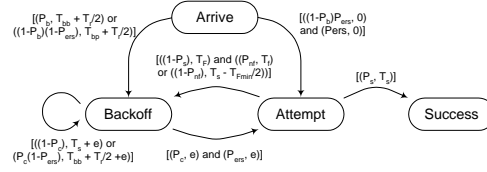


Figure 2: The delay model for a single device in the Markov chain using the presented MAC protocol.

makes upon receiving data to send, the probability of such transitions, and the related average delays until the data is successfully delivered. Let  $E(A)$  equal the expected delay incurred on each visit by the device to the *Attempt* state, and let  $E(B)$  equal the expected delay incurred on each visit to the *Backoff* state. From Fig. 2 we obtain:

$$D = P_b \left[ T_{bb} + \frac{T_r}{2} + E(B) \right] + \\ (1 - P_b)(1 - P_{ers}) \left[ T_{pb} + \frac{T_r}{2} + E(B) \right] + \\ (1 - P_b)P_{ers} [0 + E(A)] . \quad (13)$$

Now,  $E(A)$  can be presented as a function of  $E(B)$  and  $E(B)$  can be solved. The probability,  $P_b$ , of finding the channel busy, is  $P_b = B/(B + I)$ .

#### IV. THROUGHPUT — DELAY PERFORMANCE

In Fig. 3, 0.01, 0.1, and 1-nonpersistent nanoMAC throughput is compared with those of ALOHA, nonpersistent CSMA and MACA. The data rate is 12.8 kbps and communications range is 60 m. NanoMAC clearly outperforms all the other protocols and the performance of MACA with small frames is surprisingly poor. The size of a single frame is 41 bytes, but nanoMAC can send 10 frames (each 41 bytes) in a row with one reservation equalling 410 bytes of data. The  $p$ -nonpersistence does not have an effect for the throughput in the expected traffic range. Instead, with lower values of  $p$  the throughput does not degrade even with extremely high traffic rates, so the lower  $p$  values make the protocol more stable.

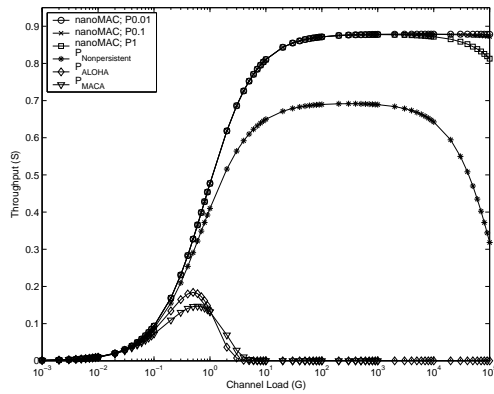


Figure 3: Throughput vs. normalised traffic with data rate 12.8 kbps, range 60 m, and frame sizes of 41 bytes. (nanoMAC  $10 \times 41$  bytes)

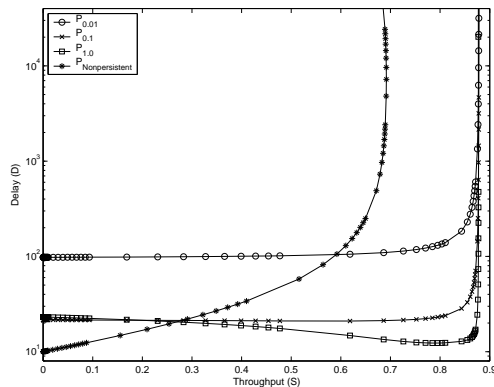


Figure 4: Theoretical, expected, average delays for the nanoMAC and nonpersistent CSMA.

Fig. 4 presents the average delays, in transmission periods, of 0.01, 0.1, and 1-nonpersistent nanoMAC and nonpersistent CSMA. Various parameters, like CS time, average backoff delays, and random backoff times are introduced into the delay calculations. The return channel for ACK is the same as communications channel and CSMA has been normalised to the same parameters as nanoMAC, meaning that data of  $10 \times 41$  bytes is sent. We can see that from an  $S$  of 0.3 and above nanoMAC has superior performance over CSMA and there is no sign of degradation. The latency increases rapidly though, when the maximum throughput is reached.

## V. CONCLUSIONS

NanoMAC is designed for a fully distributed sensor network. It has been designed for saving power in several different ways which include frame optimisation by header minimisation and frame train structure, sleep periods when other devices are transmitting, and "smart" ACK concatenation. Mathematical throughput and delay calculations are presented as well as comparison with a few commonly known distributed MAC protocols. Future work includes the verification of nanoMAC performance with a microcontroller and an ISM radio transceiver, and an analysis of node and overall power efficiency.

## REFERENCES

- [1] Raphael Rom and Moshe Sidi, editors. *Multiple Access Protocols: Performance and Analysis*. Springer-Verlag, first edition, 1990.
- [2] N. Abramson. The aloha system—another alternative for computer communications. In *1970 Fall Joint Computer Conference*, volume 37, pages 281–285, 1970.
- [3] L Kleinrock and F. A. Tobagi. Packet radio in radio channels, part 1: Carrier sense multiple access modes and their throughput-delay characteristics. *IEEE Transactions on Communications*, pages 23(12): 1400–1416, December 1975.
- [4] Phil Karn. A new channel access method for packet radio. In *ARRL/CRRL Amateur Radio 9th Computer Networking Conference*, 22 September 1990.
- [5] Chane Lee Fullmer. *Collision Avoidance Techniques for Packet-Radio Networks*. PhD thesis, University of California Santa Cruz, June 1998.
- [6] Jussi Haapola. *Low-Power Wireless Measurement System for Physics Sensors*. PhD thesis, University of Oulu, Finland, June 2002. [www.ee.oulu.fi/~jhaapola/M\\_Sc\\_thesis.pdf](http://www.ee.oulu.fi/~jhaapola/M_Sc_thesis.pdf).