MAC Protocols in Simple Sensor Networks

- -
5031877

- -
5300126

Matthias Franz
6298796

***Abstract*—This report discusses experiments conducted on a MAC protocol simulator. The simulator implements ALOHA and a simple RTS CTS protocol. Using the simulator the case of many nodes trying to communicate with a central data sink is explored, as is common in sensor networks. The effect of message timing, using a sending interval with nodes choosing random transmit times within the interval, and data length get investigated. The superior performance of RTS CTS in case of larger data sizes and the collision reducing effect of the sending interval are displayed and discussed.**

I. INTRODUCTION

Sensor networks often face the challenge of many nodes trying to transmit to the same central sink. In case of a dynamic sensor network, the sensor nodes cannot be configured with a global schedule that would prevent collisions. Receiver initiated protocols can also prove difficult if nodes are highly mobile or join and leave the network at whim. This report explores the effect of the MAC protocol on a simple setup in which every node wants to transmit to the sink, but has no information on the rest of the network.

A. ALOHA

ALOHA is a very simple MAC protocol. Nodes send messages as soon as they are ready. After sending, a node waits for an Acknowledgment (ACK). In case of a collision at the receiver, the transmitter will not receive an ACK packet and will back-off. After the random binary exponential back-off, the transmitter will try to transmit again.

B. RTS CTS

We want to compare ALOHA to another protocol that uses request-to-send (RTS) and clear-to-send (CTS) packets to avoid collisions. On receiving a RTS or CTS not meant for itself, a node will back-off long enough to allow the transmitter of the RTS or CTS to finish its communication. The communication ends in an ACK.

II. SIMULATOR ARCHITECTURE

The simulator is implemented in python using matplotlib for visualization. The simulator works on discrete time, with one central clock being used by every node for event management. Both protocols have been implemented as a state machine. The simulator takes propagation times and the time it takes to receive and send a packet into account.

Propagation time, the time it takes for a message to be received, is measured in clock ticks. When both sending and receiving a message, it takes a node the same number of clock ticks to send or receive a message as the message length. In this way the simulation takes into account that it takes longer to both send and receive longer messages. A message also takes the same number of clock ticks as the distance between the nodes to arrive. This allows the simulation to reflect the fact that messages take time to arrive and need to travel over the distance between nodes.

The pathloss model used is a simple geometric model because as long as the receiving node is within the radius of the transmission range of the sender, the message will be received. The probability of a link within the circle representing the range of a node is 1.

III. EXPERIMENTS

This section explains the setup of the two conducted experiments, data sink and data sink with random schedule, and the measurements taken. The topology for both setups can be found in Fig. 1. Every simulation for every set of parameters has been run 25 times.

A. Data sink

In this experiment 29 nodes try to transmit to a central data sink simultaneously. Every node will send their message at the same time. This will lead to an initial collision. To evaluate the efficiency of the MAC protocols the number of collisions and the time taken (T) until every node successfully sends its message to the sink get measured. This scenario represents a typical sensor network. The effects of an increase in data size get evaluated. Sensor networks might not send a data reading as soon as they have one, but might accumulate a lot of readings and send them in bulk.

B. Data sink with random schedule

This experiment uses the same topology as the data sink experiment, but not every node transmits simultaneously. A random interval in which the nodes can start their transmission is given. The effect of the size of the random interval in combination with the transmitted data size is explored.

IV. RESULTS

The results of the experiments can be seen in Table I for the data sink experiment and Table II for the data sink with random schedule experiment.

V. DISCUSSION

This section discusses the results and expectations of the simulations.

TABLE I

TRANSMITTING TO DATA SINK WITH EVERY NODE STARTING SIMULTANEOUSLY

	ALOHA				RTS CTS			
Data size	$\overline{\#Collisions}$	$\overline{T}$	σ of # Collisions	σ of T	$\overline{\#Collisions}$	$\overline{T}$	σ of # Collisions	σ of T
5	2023	1942	795	244	389	2762	22	345
25	14884	6139	1187	1022	444	4028	20	303
50	80723	21697	14905	3336	472	5542	26	496
75	399948	73839	30213	1826	510	6861	30	565
100	-	-	-	-	536	8197	29	666
125	-	-	-	-	547	9265	31	769
150	-	-	-	-	575	11268	50	1060
175	-	-	-	-	578	12358	38	936
200	-	-	-	-	632	13368	42	865

TABLE II

ALOHA DATA SINK WITH RANDOM SCHEDULE

		ALOHA				RTS CTS			
Data size	Random interval	$\overline{\#Collisions}$	$\overline{T}$	σ # Collisions	σ time taken	$\overline{\#Collisions}$	$\overline{T}$	σ # Collisions	σ time taken
5	25	2142	2042	71	343	301	2794	22	321
10	25	5158	3131	207	397	308	3194	26	331
15	25	8389	4009	507	512	328	3536	26	266
20	25	12578	5397	753	777	336	3711	25	320
25	25	17355	6451	1034	812	360	4154	25	328
5	50	2114	2017	73	348	299	2763	25	286
10	50	5307	3030	174	400	309	3117	28	316
15	50	8879	4199	429	570	328	3511	30	325
20	50	12752	5401	750	700	345	3777	21	287
25	50	17781	6366	1418	832	367	4047	28	324
5	75	2050	2034	71	281	295	2820	18	409
10	75	5248	3090	201	459	315	3124	23	453
15	75	9092	4086	345	372	329	3503	25	285
20	75	13218	5163	609	522	348	3787	28	321
25	75	18402	6607	1466	891	371	4186	29	373
5	100	2039	1992	59	222	287	2668	23	366
10	100	5224	3182	217	459	300	2991	19	346
15	100	9090	4028	410	511	326	3425	23	376
20	100	13530	5151	935	818	350	3751	27	333
25	100	18490	6469	1251	977	365	4248	15	426

A. Data sink

Table I and Fig. 2 show the results of the data sink experiment. For the smallest data size ALOHA is more efficient than a RTS CTS protocol, but already at a data size of 25, RTS CTS is 1.5 times quicker in transmitting all messages to the sink than ALOHA. The number of collisions in ALOHA increase strongly with an increase in data size, RTS CTS on the other hand scales well, since the RTS CTS exchange takes care that no collision happens while transmitting long data packets. For a data size of 75 ALOHA already experiences 784 times more collisions and is 10 times slower than RTS CTS. ALOHA simulations using a larger data size than 75 have been aborted since they took too much time.

B. Data sink with random schedule

Table II and Fig. 3 show the results of the data sink with random schedule experiment. The largest random interval with the smallest data size leads to the lowest T and number of occurred collisions in both ALOHA and RTS CTS. The effect of the random interval is strongest, if it is large in comparison to the data size.

The number of collisions in RTS CTS is always lower with random transmit times than with simultaneous transmission. This is because the first initial collision gets avoided, and with a larger random interval even more collisions get avoided, since less nodes transmit their RTS signals at the same time. Even with fewer collisions, T is lower than for simultaneous transmission.

The random interval had a negative effect on ALOHA for both T and the number of collisions. This might be because the random interval is too small for the number of nodes. When using a data size of 5 with a random interval of 500, T is 1615 and the number of collisions at the sink is 157, still RTS CTS

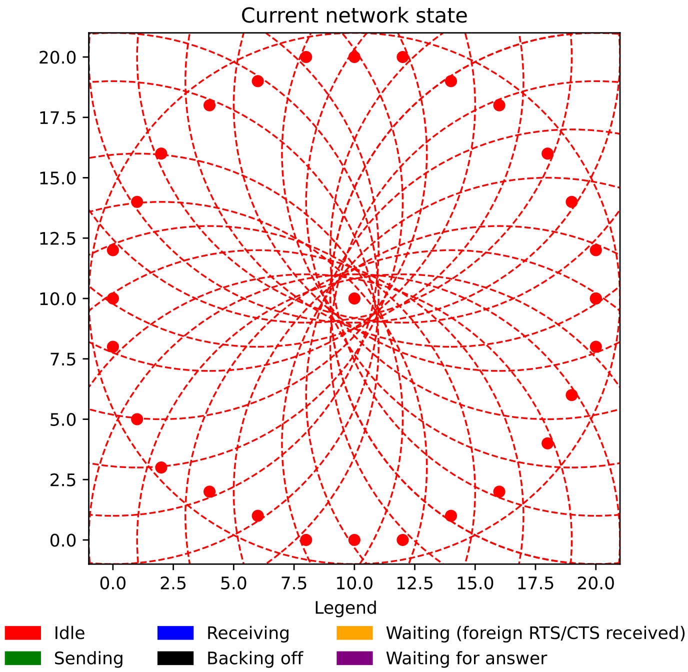

Fig. 1. **Network topology of data sink experiment.** The dots show the location of the node, the red lined circles show the transceive range of the node. The color of the node corresponds the the nodes state. X and Y axis show the distance between nodes

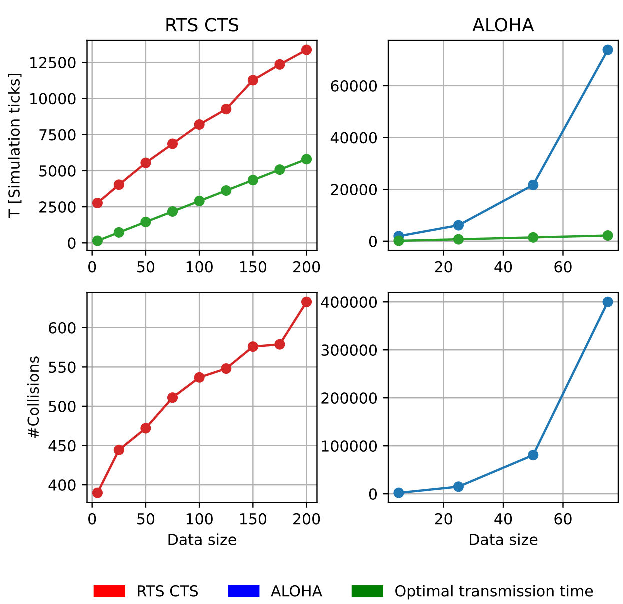

Fig. 2. **Average collisions and T plotted over data length.** This plot shows how T and number of collisions scale with the data length. ALOHA only has data till a data size of 75, since the simulations with longer data lengths took too much time.

is more efficient, taking 1581 simulation ticks and having only

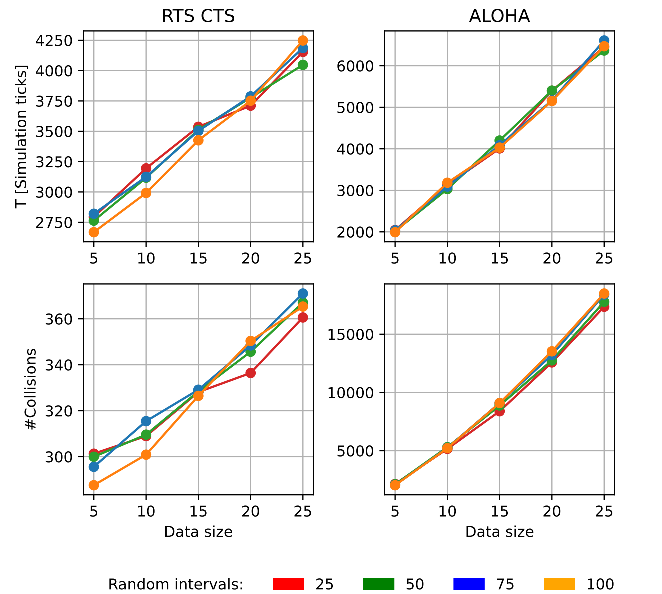

Fig. 3. **Average collisions and T plotted over data length and random transmit time interval.** This plot shows how T and number of collisions scale with the data length and the random transmit time interval.

17 collisions at the sink. An optimized ALOHA schedule, with every node transmitting as soon as the previous one finished (sending every 26 ticks), no collisions happen and it takes 1016 simulation ticks for every node to transmit to the sink. Using the knowledge of the number of nodes and the time it takes for every node to transmit to the sink, a optimal random interval of 988 can be chosen, leading to 20 collisions and a T of 1187. ALOHA with an optimal schedule will be more efficient than RTS CTS because it needs to send less packets.

VI. Conclusion

ALOHA benefits from small data sizes and large sending intervals. With an increase in data size RTS CTS quickly experiences fewer collisions and becomes faster than ALOHA. In case a static schedule can be implemented, ALOHA will be more efficient. Ad Hoc sensor networks will mostly not be able to implement static schedules, making RTS CTS a more efficient choice, even when choosing a large sending interval.

In real life applications the probability of nodes receiving two RTS' signals at the same time is very low, meaning a small sending interval is already built into the networks. An easy and cheap way to reduce collisions and therefore also energy wastage is to let the nodes decide when to send using a large sending interval. This also simplifies adding new nodes to the network, since they do not need to sync to a schedule nor do the other nodes in the network need to do anything, all that's needed is to give the new node a sending interval. This leads to a higher T but more energy efficiency which is often more important in sensor networks.