

Robert G. Gallager
Massachusetts Institute of Technology
Department of Electrical Engineering
and Computer Science
77 Massachusetts Avenue
Cambridge, Massachusetts 02139

Summary

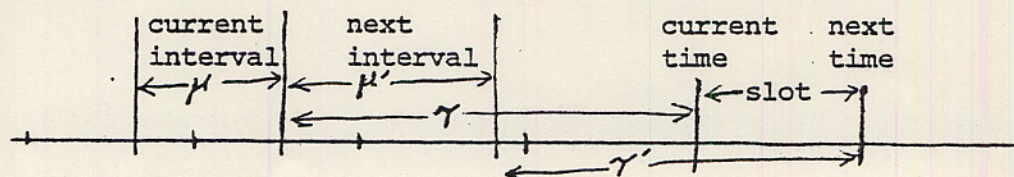
Consider a random access broadcast medium such as that in a slotted Aloha system. In particular assume a very large, effectively infinite, population of sources each sending messages with Poisson arrivals, to a common receiver. Each message has a fixed duration, taken as one time unit, and the sources are synchronized in the sense that each transmitted message arrives at the receiver over a unit interval $(i, i+1)$ for some integer i . These unit intervals are called slots. Thus any two messages either overlap completely, arriving in the same slot of time, or do not overlap at all. We assume that if only one message is transmitted in a given slot, it is received without error, whereas if two or more messages are in a common slot, the messages conflict and no information about any of the messages is received. We assume finally that at the end of a slot, each source detects whether zero, one, or more than one messages were transmitted in that slot. There are, moreover, no "side channels" of information through which the sources can schedule the use of the slots. The problem is to design an algorithm for each source so that each conflicting message can be eventually rebroadcast without conflict with the best tradeoff between expected delay and overall arrival rate.

The original slotted Aloha system, due to Abramson and Roberts, operates with each source transmitting each message in the first slot after its arrival; if a message is involved in a conflict it is retransmitted after a random interval selected from some given probability distribution. The maximum throughput for such a system is $1/e$ messages per time unit, but, as shown subsequently by several authors, the system is unstable. Many variations to this basic type of system have been proposed using various techniques such as reservations, carrier sensing, the FM capture effect, etc. to reduce conflicts. Such techniques achieve much higher throughputs than $1/e$ and are clearly desirable in practice, but we have intentionally excluded them from our model here since we want to study the underlying problem of conflict resolution in its simplest context.

The work reported here stems from some earlier results by J. Capetenakis, "The Multiple Access Broadcast Channel: Protocol and Capacity Considerations", Report ESL-R-806, Electronic Systems Laboratory, M.I.T., March 1978. Capetenakis devised a tree search algorithm at the sources for resolving conflicts and showed that the system was stable with a maximum throughput of 0.43 messages per slot. The new algorithm reported here is somewhat simpler and has a maximum stable throughput of .487 messages per slot. The new algorithm also has the peculiar property of delivering messages to the common receiver in a strictly first come first served order. Each source keeps track of two common parameters, the current delay τ and the current transmission interval μ . Each source with a message

PROCEEDINGS AFOSR WORKSHOP IN COMM. THEORY & APPL.
SEPT. 17-20, 1978, PROVINCETON, MA

to transmit also keeps track of the time t since that message arrived. The source transmits that message in a given slot if, at the beginning of the slot, $\gamma \leq t \leq \gamma + \mu$. If, at the end of that slot, no conflicts have occurred, then γ is decreased by $\mu - 1$; the term μ corresponds to the fact that all messages that arrived in the given time interval of μ are successfully transmitted, and the term 1 corresponds to the elapse of 1 time interval. The figure below illustrates this behavior, using γ' and μ' for the values of γ and μ at the beginning of the next slot.



If a conflict occurs, then additional slots must be devoted to resolving the conflicts in the current transmission interval. This is done by retransmitting the messages in the first half of the interval μ . This is accomplished by choosing the new value of γ to be $\gamma' = \gamma + 1 + \mu/2$ and choosing the new value of μ to be $\mu' = \mu/2$. If this new slot results in the successful transmission of one message, then the second half interval is subsequently transmitted, choosing the new value of γ to be $\gamma'' = \gamma' + 1 - \mu'$ and choosing the new value of μ as $\mu'' = \mu'$. On the other hand, if the new slot after a conflict results in no messages being transmitted, then it is known that a conflict must exist in the second half interval, which is then subdivided again for the following transmission, using $\gamma'' = \gamma' + 1 - \mu'/2$ and $\mu'' = \mu'/2$. The idea of using this subdivision in this case is due to J. Massey. Finally, if the new slot after a conflict results in yet another conflict, we further subdivide the given interval μ' , using $\gamma''' = \gamma'' + 1 - \mu''/2$ and $\mu''' = \mu''/2$. With a little thought, it can be seen that the conditional distribution, in this case, of the number of messages in the second interval of size μ' is Poisson, and that this interval is statistically homogeneous with the waiting interval for which the messages have never been transmitted.

Because of the above homogeneity, we define an epoque of conflict resolution to be completed whenever a conflict is followed by two successful transmissions of a single message each. At the end of an epoques, μ is chosen for the next slot as some predetermined value, μ_0 . After the analysis to be outlined, it turns out that stable throughput is maximized at $\mu_0 = 2.52$, but choosing $\mu_0 = 2$ loses only about 1% of the maximum throughput and reduces the delay for more moderate loads. For completeness, we now present the entire algorithm, as exercised at each source node. Initially, we take $\gamma = 0$ and $\mu = 0$, and start at step 1, which corresponds both to the beginning of an epoques and also to the second half interval after a conflict. Step 2 corresponds to the first half interval after a conflict.

In order to analyze the algorithm, we observe that the sequence of delays, γ , at the beginnings of successive epoques, forms a random walk with a single barrier at $\gamma = 0$. γ will have a finite expected value, and in fact moments of all orders, if the increment in γ from one epoques to the next has a negative expected value and moments of all orders. The increment in γ , however, can be analyzed by a Markov chain corresponding to the steps of the algorithm. The existence of a moment generating function

for the increments is almost obvious from the Markov chain, and the negative expectation of the increments for an input rate less than .487 was established numerically with an HP25 hand calculator.

Algorithm

- 1) If there is a message whose delay is between γ and $\gamma+\mu$, transmit it;
 At the end of the slot,
 if a conflict has occurred then
 $\gamma \leftarrow \gamma + 1 + \mu/2$;
 $\mu \leftarrow \mu/2$;
 goto step 2;
 else
 $\mu \leftarrow \min(\mu_0, \gamma+1)$;
 $\gamma \leftarrow \max(0, \gamma+1-\mu_0)$;
 goto step 1;
- 2) If there is a message with delay between γ and $\gamma+\mu$, transmit it;
 At the end of the slot,
 if a conflict has occurred then
 $\gamma \leftarrow \gamma + 1 + \mu/2$;
 $\mu \leftarrow \mu/2$;
 goto step 2;
 else if no messages in slot then
 $\gamma \leftarrow \gamma + 1 - \mu/2$;
 $\mu \leftarrow \mu/2$;
 goto step 2;
 else
 $\gamma \leftarrow \gamma + 1 - \mu$;
 goto step 1;

The results here can be extended to the case of a finite number of sources, and with a slight modification of the algorithm, the system makes a smooth transition into self scheduled time division multiplexing under input rates approaching 1 message per slot. The algorithm can also be extended to packet radio networks, but so far we have been unable to analyze its performance in this case.