A NON-TOKEN-BASED-DISTRIBUTED MUTUAL EXCLUSION ALGORITHM FOR SINGLE-HOP MOBILE AD HOC NETWORKS

Romain Mellier and Jean-Frederic Myoupo
LaRIA, CNRS FRE 2733, Université de Picardie Jules Verne,
5, rue du Moulin Neuf 80000 Amiens, France
{mellier, myoupo}@laria.u-picardie.fr

Vlady Ravelomanana
LIPN, CNRS, UMR 7030, Université Paris 13,
Av. J-B Clément 93430 Villetaneuse, France
vlady.ravelomanana@lipn.univ-paris13.fr

Abstract This paper presents a simple mutual exclusion algorithm for ad hoc mobile networks. Our algorithm does not use the token circulation technique. A station which requests a Critical Section (CS) competes in order to be alone to use the unique channel dedicated to this CS. To reach this goal, we derive a Markov process which guarantees that each station will enter the CS. More precisely, we show that, in presence of collision detection, n/ln2 broadcast rounds are necessary in the average case to satisfy n (n unknown) stations wishing to enter the same CS.

Keywords: Mutual Exclusion , Ad Hoc Mobile Networks, Wireless Networks.

1. Introduction

Nowadays, the research in Mobile Ad hoc NETworks (MANET for short) is attractive and desirable, due to the development of wireless networks and personal communication ([Bose et al., 1999], [Chlebus et al., 2002], [Garg and Wilkes, 1996], [Hayashi et al., 2000], [Lakshdisi et al., 2001], [Lin and Stojmenovic, 2001], [Malpani et al., 2000], [Malpani et al., 2001], [Myoupo, 2003], [Myoupo et al., 2003], [Vaidya et al., 2001]) . Mobile ad hoc networks are formed by a collection of mobile wireless nodes. Communication links form and disappear as nodes come into and go out of each other communication range. Such networks have many practical applications, including home networking, search-and-rescue, and military operations. We assume that critical sections are disseminated in the network : some stations can be dedicated or specialized to collect or to give some information concerning the characteristics of the network. We can consider a rescue ad hoc network for example. One of its station can be dedicated to centralize the material damages collected by all stations. Another one can be specialized in the collection of medical information and so on. A *unique channel* is assigned to each dedicated station. It is used by the other stations to send or to receive the needed information. So the access to a dedicated station can only be done in mutual exclusion way. More precisely, a station which desires to send or to receive an information from a specialized station, through a channel k_0, may compete in order to be alone to broadcast in k_0. Therefore, the access to one of these dedicated or specialized stations is considered as a *Critical Section (CS)* in the mobile network.

1.1 Related Works

Intuitively, one can believe that a mutual exclusion algorithm for ad hoc networks can be obtained by a simple adaptation of a mutual exclusion algorithm for wired networks. But it is not obvious due to the permanent change of the topology of the network in a MANET. Token-based mutual exclusion algorithms provide access to CS through the maintenance of a single token that cannot simultaneously be present at more that one station in the network. Only the station holding the token can enter the CS. From this token consideration, we can quote first the work of Raymond ([Raymond, 1989]) in which the requests are sent over a static spanning tree of the network, toward the token holder. Chang et al ([Chang et al., 1990]) extend the previous algorithm by inducing a token oriented Directed Acyclic Graph (DAG) for the search of the token holder. Next, Dhamdhere and Kulkarni ([Dhamdhere and Kulkarni, 1994]) showed that the algorithm of Chang et al ([Chang et al., 1990]) can suffer from deadlock and brought a solution to this problem by assigning a dynamic changing sequence number to each node yielding a total ordering of stations in the network. Recently J.E. Walter et al ([Vaidya et al., 2001]) proposed a token-based mutual exclusion algorithm, for a MANET, which combines the ideas from ([Dhamdhere and Kulkarni, 1994]) adapted to a mobile environment. They use the partial reversal technique introduced in [Bertsekas and Gafni, 1981] to maintain a token oriented DAG with a dynamic destination. Their approach induces a logical DAG on the network, dynamically modifying the logical structure to adapt to the changing physical topology in the ad hoc environment.

1.2 Our contribution

This paper presents a distributed mutual exclusion protocol obtained from a Markov process. Our approach does not use a token circulation technique. It guarantees that each candidate for a CS will be satisfied. We give the average number of broadcast rounds necessary for n stations, n unknown, to enter the same CS. Our idea is to construct a splitting process which yields a random binary tree. With the help of the probabilistic divide-and-conquer technique, we derive a single hop protocol which requires $n/\ln 2$ broadcast rounds in average case. Where *ln* is the logarithmic function in basis e and n is the number of stations wishing to enter the same CS.

The rest of this work is organized as follows : the environment considered in this work is presented in section 2. In section 3 a simple single-hop protocol is presented, and its performance is obtained with the help of average case analysis. Section 4 contains brief comments on experimental results. Concluding remarks end the paper

2. Basic definitions

An Ad Hoc Network is a set S of N radio transceivers or stations which can transmit and/or receive messages through a set C of k channels (a MANET(N, k) for short). The time is assumed to be slotted and all stations have a local clock that keeps synchronous time. In any time slot, a station can tune into one channel and/ or broadcast on at most one channel. A broadcast operation involves a data packet whose length is such that the broadcast operation can be completed within one time slot. So, in the MANET with *collision detection* (CD for short), the status of an n-station MANET channel is :

- **NULL** : if no station broadcasts on the channel in the current slot,
- **SINGLE** : if exactly one station broadcasts on the channel and
- **COLLISION** : if two or more stations broadcast on the channel in the current time slot.

Also, all communications are performed at time slot boundaries, i.e. the duration of broadcast operations is assumed to be sufficiently short.

1 Let us consider N stations which communicate by message passing over a wireless network through k distinct communication channels. Each station runs an application process and a mutual exclusion process (to get a resource) that communicate with each other to ensure that the station cycles between its REMAINDER section (not interested in the CS), its WAITING section (waiting for access to CS), and its CRITICAL section. **Only the station which broadcast yields a single status of the channel executes the CS**. When leaving the CS, it broadcasts in the unique channel to inform the other stations that they can compete to enter the CS.

2 We suppose that n is the number of candidates for entering the same critical section. It is clear that $n \leq N$

3 The system is a **single-hop** network.

We assume that critical sections are disseminated in the network : some stations can be dedicated or specialized to collect or to give some information concerning the characteristics of the network. For example we can consider a rescue ad hoc network . One of its stations can be dedicated to centralize the material damages collected by other stations. Another one can be specialized in the collection of medical information and so on. *A unique channel* is assigned to each dedicated station. It is used by the other stations to send or receive the needed information. So the access to a dedicated station can only be done in a mutual exclusion way. Therefore these specialized stations are considered as *resources* by the others. More precisely, a station which wants to send or to receive an information from a *resource* station through a channel, say k_0, may compete to be alone to broadcast in k_0. Therefore, communications with all these dedicated or specialized stations are *Critical Sections* (CS).

3. A single-hop mutual exclusion algorithm

Here, S is the set of stations which require access to the same critical section, say CS_0. First, we give a procedure that can split randomly the set S of stations into two non-empty subsets, say S_1 , S_2. Each station is assumed to have the computing power of small laptop such that they can generate random bits and store few data. The time is slotted. The following procedure is an implementation of *Bernoulli* process where the protocol *Single-Hop Ex-Mut* below randomly partitions a given set S (for example the initial set of stations), into two subsets S_1 and S_2. The process is repeated until there is two non-null parts. Each station has a counter initialized to zero and runs the following protocol in order to enter in CS_0.

3.1 Processing an example

We begin this section by presenting an example of MANET(5, 1) in figure 1 will help to understand the basic idea our approach. In this figure, we suppose that the protocol works first with the most left branch until its leaves are reached. Then, it goes backward working again in the most left branch with nodes containing more than one station, and so on. This process is recursively done until the leaves of all branches are considered.
In figure 1, the number of stations that have chosen the same bit are indicated in the nodes. The stations which have chosen bit 1 are on nodes of the right sub-tree. Those which chosen bit 0 are in the nodes of the left sub-tree. The numbers in bold around the node means a step of our example. Before talking our example, we first give a brief description of our algorithm. The set of stations is recursively partitioned into two subsets S_1 and S_2. Each station owns a counter. All counters are initialized to 0.

1 A station has the right to broadcast in the channel only if it is in S_1 and if its counter shows 0.

2 A station in S_2 can broadcast only if the status of channel is either Single or Collision and if its counter shows 0 and if S_1 is empty.

3 If S_1 is empty and S_2 is not empty then the stations of S_2 which counter show 0 move in S_1.

4 After the random choice of a bit from set {0, 1}, if S_1 (if collision status) and S_2 (if collision status) are not empty, then the stations in S_2 increase their counters by 1.

5 If there are two consecutive single status of the channel, then each station in S_2 decreases its counter by 1.

6 If the status of the channel is single, then the station which has broadcasted executes its CS.

We are now in a position to process our example :

- **Step 1** : Initially the five stations broadcast in the unique channel, obviously with collision. Then, each of them chooses a bit in $\{0, 1\}$ at random. Three of them choose bit 1 (set S_1) and two bit 0 (set S_2). Each station of S_2 increments its counter by one.

- **Step 2** : The three stations in S_1 broadcast in the unique channel and obviously with collision again. Each of them, again chooses a bit in $\{0, 1\}$ at random. Two of them choose 1 (S_1) and one chooses bit 0 (S_2). The unique station in S_2 executes its CS.

- **Step 3** : None of the two stations in S_1 chooses bit 1. Hence $S_1 = \emptyset$.

- **Step 4** : Since $S_1 = \emptyset$ then $S_1 \leftarrow S_2$.

- **Step 5** : One station of S_1 chooses 1 and the other bit 0. Therefore the status of the channel is single. Then only station in S_1 executes its CS.

- **Step 6** : The unique station in S_2 which counter shows 0 moves in S_1 and broadcasts with single status . Then it executes its CS.

- **Step 7** : We then have two consecutive single status. Therefore each station in S_2 decreases its counter by one. Consequently the counters of the two stations which went in S_2 in step 1 show 0 each. So, they move in S_1. One of them chooses 1 and the other bit 0.

- **Step 8** : The unique station in S_1 executes in its CS.

- **Step 9** : Since $S_1 = \emptyset$, $S_1 \leftarrow S_2$. The unique and last station in S_1 finally also executes in its CS.

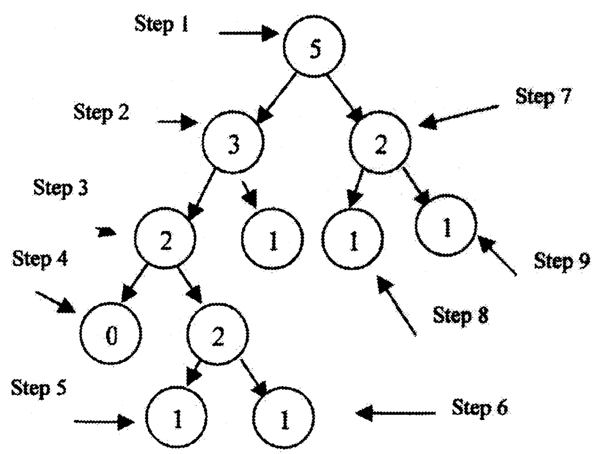

Figure 1. Mutual Exclusion on a MANET(5, 1)

3.2 The Algorithm

Now that we have cleared the ideas which guide our motivation, we now turn to give the procedure that each station must run in order to perform the mutual exclusion. More precisely a station runs the algorithm which follows :

Procedure Single-Hop-Ex-Mut (INPUT : S, OUTPUT : S_1, S_2)
1. $S_1 \leftarrow S$; $S_2 \leftarrow \emptyset$;
2. **WHILE two consecutive status of the channel are not NULL (i.e. $S_1 \neq \emptyset$ AND $S_2 \neq \emptyset$) DO**
3. Each station listens to the unique channel while the protocol is running
4. **REPEAT**
   "WAITING Section"
5. Each station in S_1 broadcasts in the unique channel assigned to CS_0.
6. IF the status of the channel is SINGLE then
   BEGIN
      the unique station in S_1 executes CS_0
      the stations of S_2 which counters show zero move in S_1
   END
7. IF the status of the channel is COLLISION then
   BEGIN
      Each station in S_2 broadcasts in the unique channel.
      IF the status of the channel is SINGLE then the unique station in S_2 executes CS_0
      IF the status of the channel is COLLISION then each station in S_2 increments its counter by one
   END

8. IF the status of the channel is NULL then
   BEGIN
      IF $S_2 \neq \emptyset$ then all stations of S_2, which counters
         show zero move in S_1
   END
9. Each station in S_1 chooses a bit in the set $\{0, 1\}$ at random.
10. Those which have chosen 1 stay in S_1 and those which have chosen 0 move in S_2.

UNTIL the unique channel has two consecutive SINGLE status.
The stations of S_2 decrease their counters by one. Those which counters show zero move in S_1.

"REMAINDER section"

11. The station which is in the CS_0 broadcasts a signal on the unique channel to mean that it is leaving the CS_0.

The effect of this message is to inform the stations in S_1 that they can now request the CS_0.
END WHILE
END Single-Hop Ex-Mut

Consider a subgroup of m stations. The probability of failure of the splitting process applied on this subset is then given by : $Pr[failure] = 2 * \frac{1}{2^m}$. So with probability equals to $1 - \frac{1}{2^m - 1}$, the procedure *Single-hop Ex-Mut* subdivides the initial set of stations into two non-empty subsets of stations. At the end of the *repeat-until* loop, each station can determine if whether

1 it is the only one that selected the bit 1, in this case the status of the channel is **SINGLE** or

2 there are at least two stations that selected bit 1, in this case the status of the corresponding channel is **COLLISION**. A station which broadcast yields a single status has the right to enter its CS. Since the power of stations allow us to store the status of the channel, only one broadcast per turn of the "repeat-until" loop in the *Single-hop Ex-Mut* procedure is needed to perform the protocol. In other words, our protocol randomly generates a binary-tree-like structure (see figure 1). Concretely, our principal problem is then to compute the number of passes through all he internal nodes (nodes of degree > 1), including repetitions which are exactly the number of broadcast rounds.

3 Note that when the channel is of single status, all stations of S_2 wait till they receive a signal from a station leaving CS_0.

3.3 The use of a counter in each station

The management of the counter is one of the key points of our algorithm. The consistency of our approach is guaranteed by the following considerations

1 each station monitors the channel, and so it knows the status of the channel at every broadcast round.

2 root of sub-tree : in the WAITING section, at level 7 of the procedure two consecutive COLLISION status (one yielded by the stations of S_1 and the second by the stations of S_2) show that after the random choice of bits 0 or 1, we have $S_1 \neq \emptyset$ and $S_2 \neq \emptyset$. Therefore the node considered has two non empty sub-trees. The stations of the right sub-tree will enter the CS_0 after those of the left sub-tree. Incrementing their counters indicate the backward processing of our algorithm.

3 decreasing the counter : according to 2. above, decreasing a counter means that all stations of the left sub-tree have been the CS_0. It is taken into account in Level 11 of the procedure.

4 It can happen that no station chooses bit 1. Then $S_1 \leftarrow S_2$ with the condition that S_2 must not be empty (see level 8 of the procedure).

3.4 Evaluation of the number of broadcast rounds necessary for n stations to enter the same CS

In this paragraph, Our goal is to introduce basic methods that are useful to analyse performance of such protocols. As we said earlier, our approach generates a binary tree and it is shown in [Flajolet and Sedgewick, 1996] that such process always ends. However we next show that the average number of broadcast rounds is given by an absolute convergent Fourier series. Then proving that the process of splitting always ends will be done.

Theorem 3.1. *Let us consider a MANET(N, 1) with CD. Suppose n stations, $n \leq N$, request the same CS. The protocol Single-Hop Ex-Mut terminates in approximately $\frac{n}{\ln 2}$ broadcast rounds on the average.*

Proof. The goal of this proof is to show that the process of splitting in the Single-Hop Ex-Mut always ends as stated earlier. It is easy to see that protocol Single-Hop Ex-Mut generates a random complete binary tree and it terminates when reaching all the nodes of degree 1. At each step of the protocol, the probability of splitting a given set of size m can be depicted as in the figure 2. Our idea is similar to the one in [Myoupo et al., 2003]. In the figure 2, each edge is weighted with $\binom{m}{p}/2^m$ which is the probability that a set of m stations will split into exactly 2 non-empty subsets of respective sizes p and $m-p$. So, the probabilistic model is here a *Markov chain* (see [Feller, 1957]) where reaching a node of the tree of degree 1 corresponds, now to an absorbing state (self-loop of the state with a probability equals to 1). If we denote by α_i the *average waiting time* to reach an absorbing state (partition recursively n stations until getting n parts), the computing is classical (cf. [Feller, 1957]) by means of linear formulae

$$\alpha_i = 1 + \sum_{r+s=i, r\geq 0, s\geq 0} \Pr[i \xrightarrow{splits} r, s]\alpha_{r,s} \qquad (1)$$

where $\Pr[i \xrightarrow{splits} r, s]$ is the probability of a transition from the "state" i to the "state" (r, s). Since in our case we have to split *sequentially*, in their turn, the two non-empty subsets (containing respectively r and s stations), i.e. to terminate all subdivisions must be done one by one.

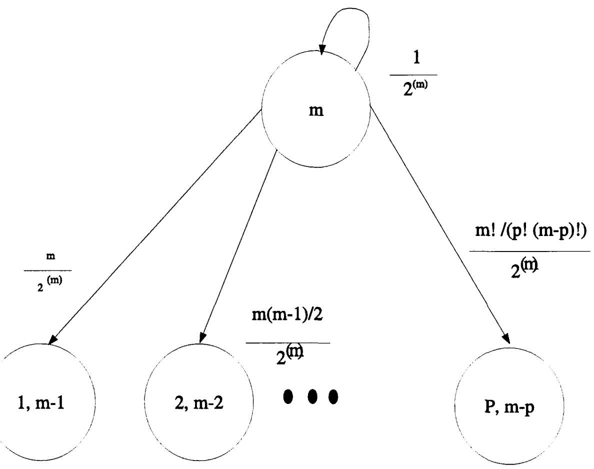

Figure 2. Splitting randomly a given number m into two parts

So, we have simply

$$\alpha_{r,s} = \alpha_r + \alpha_s \tag{2}$$

Here, $\alpha_0 = \alpha_1 = 0$ and α_n, $n \geq 2$ verifies

$$\alpha_n = 1 + \frac{1}{2^{n-1}}\alpha_n + \sum_{p=0}^{n=1} \frac{\binom{n}{p}}{2^{n-1}}\alpha_p \tag{3}$$

Now, let us introduce the exponential generating function (see for example [Comtet, 1974] for more details)

$$\alpha(x) = \sum_{n=2}^{\infty} \alpha_n \frac{x^n}{n!} \tag{4}$$

In formula (4), the average number of broadcast rounds, α_n is given as the coefficient of $\frac{x^n}{n!}$ of the power series $\alpha(x)$. Recall that if $P(x)$ is a polynomial, the notation $[x^n]P(x)$ gives the coefficient of $P(x)$. This notation applies to our function $\alpha(x)$ and the average number of broadcast rounds is then

$$\alpha_n = n![x^n]\alpha(x) \tag{5}$$

Replacing α_n in (4) by its expression from the formula (3), we obtain the following linear functionnal equation :

$$\alpha(x) = e^x - x - 1 + 2e^{\frac{x}{2}}\alpha(\frac{x}{2}) \tag{6}$$

First terms are given below

$$\alpha(x) = 2x^2 + \frac{10}{3}x^3 + \frac{100}{21}x^4 + \frac{652}{105}x^5 + \frac{24922}{3255}x^6 + \frac{9874}{1085}x^7 + \cdots \quad (7)$$

Thus, the coefficient $\frac{652}{105}$ is then the number of average broadcast rounds needed to initialize 5 stations. Successive iterations of (6) lead to

$$\begin{aligned}
\alpha(x) &= e^x - x - 1 + 2e^{\frac{x}{2}}\alpha(\frac{x}{2}) \\
&= e^x - x - 1 + 2e^{\frac{x}{2}}(e^{\frac{x}{2}} - \frac{x}{2} - 1) + 4e^{\frac{3x}{4}}\alpha(\frac{x}{4}) \\
\ldots \\
&= \sum_{j=0}^{\infty} 2^j exp((1 - \frac{1}{2^j})x)(exp(\frac{x}{2^j} - \frac{x}{2^j} - 1)
\end{aligned}$$

When expanding the exponentials, we have

$$\alpha_n = \sum_{j=0}^{\infty} 2^j (1 - (1 - \frac{1}{2^j})^n - \frac{n}{2^j}(1 - \frac{1}{2^j})^{n-1}) \quad (8)$$

As, we have

$$(1 - \epsilon)^n = e^{-n\epsilon + O(n\epsilon^2)} \quad (9)$$

splitting the right hand side of (8), one can legitimate the use of (9) and we have

$$\alpha_n \approx \frac{n}{\ln 2} \quad (10)$$

Note that *Mellin transforms* methods, see for instance [Knuth, 1973] and [Flajolet and Sedgewick, 1996] for more details, are well suited for asymptotic estimates of coefficients of linear functionnal equations like (6). Here, it gives an additional fluctuating and periodic term $P(\frac{\ln n}{\ln 2})$ where P is an absolute convergent Fourier series of variations $\leq 10^{-5}$. In [Knuth, 1973], Knuth derived explicit expressions of the fluctuating term. Note also that, $1/\ln 2 \approx 1.44$. So there are 44% "waste of broadcast rounds". □

4. Experimental results

This section presents graphics on the evolution in time of the variation of the number of stations which request the CS. The irregularities of the slopes of these curves are due to the random choice of it 1 or 0.

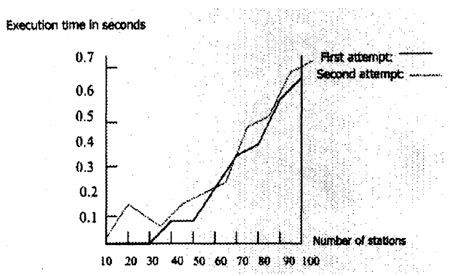

Figure 3. Mutual exclusion for 100 stations : two attempts

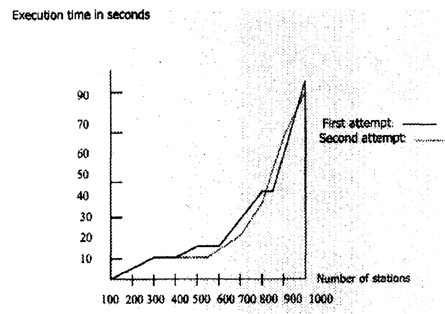

Figure 4. Mutual exclusion for 1000 stations : two attempts

5. Concluding remarks

In this paper we considered a MANET (N, k) from which we derived a distributed mutual exclusion protocol from a Markov process. Our approach does not use a token circulation technique. However it guarantees that each candidate for a CS will be satisfied. The performance of our protocol in terms of broadcast round is evaluated. More precisely, it requires $n/\ln 2$ broadcast rounds in average. Where ln is the logarithmic function in basis e and n is the number of stations wishing to enter the same CS. An interesting challenge is to derive a k-mutual exclusion protocol in a MANET from our approach. In our approach, we assume that confidential data items are encoded. The reader may argue that the use of token as in [19] guarantees more confidentiality of data items to be broadcasted. It is not always true, because it is well known that hijacking data items broadcasted in an air channel is easy to realize. Therefore even with the use of token the confidential data items must be encoded. Finally, a station can execute its CS as many times as its needs : after it has left its CS, it waits until it hears two consecutive NULL status of the channel. Then it runs the *Single-Hop-Ex-Mut*.

References

[Bertsekas and Gafni, 1981] Bertsekas, D. and Gafni, E. (1981). Distributed algorithms for generating loop-free routes in networks with frequently changing topology. In *IEEE Transactions on Communications, C-29(1) : 11-18.*

[Bose et al., 1999] Bose, P., Morin, P., Stojmenovic, I., and Urrutia, J. (1999). Routing with guaranteed delivery in ad hoc wireless networks. In *Proc. of 3rd ACM Int. Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications DIAL M99, Seattle, August 20, 1999, 48-55.*

[Chang et al., 1990] Chang, Y., Liu, M., and Singhal, M. (1990). A fault tolerant algorithm for distributed mutual exclusion. In *Proc. of 9th IEEE Symp. On Reliable Distr. Systems, pp. 146-154.*

[Chlebus et al., 2002] Chlebus, B.S., Gasienec, L., Gibbons, A., Pelc, A., and Rytter, W. (2002). Deterministic broadcasting in ad hoc networks. *Distributed computing*, 15:27–38.

[Comtet, 1974] Comtet, L. (1974). *Advanced combinatorics*. Reidel, Dodrecht.

[Dhamdhere and Kulkarni, 1994] Dhamdhere, D.M. and Kulkarni, S.S. (1994). A token based k-resilient mutual exclusion algorithm for distributed systems. *Inform. Process. Letters*, (50):151–157.

[Feller, 1957] Feller, W. (1957). *An introduction to probability theory and its applications*. John Wiley, New-York.

[Flajolet and Sedgewick, 1996] Flajolet, P. and Sedgewick, R. (1996). *Introduction to the analysis of algorithms*. Addison-Wesley.

[Garg and Wilkes, 1996] Garg, V.K. and Wilkes, J.E. (1996). *Wireless and personnal communication systems*. Prentice-Hall PTR, Upper Saddle River, NJ.

[Hayashi et al., 2000] Hayashi, T., Nakano, K., and Olariu, S. (2000). Randomized initialization protocols for ad hoc networks. *IEEE Trans. Parallel Distr. Syst.*, 11:749–759.

[Knuth, 1973] Knuth, D.E. (1973). *The art of computer programming, vol 3 : Sorting and Searching*. Addison-Wesley.

[Lakshdisi et al., 2001] Lakshdisi, M., Lin, X., and Stojmenovic, I. (2001). Location based localized alternate, disjoint, multi-path and component routing algorithms for wireless networks. In *ACM Symposium on Mobile Ad Hoc Networking And Computing MobiHoc, Long Beach, California, USA, October 4-5, 2001, 287-290.*

[Lin and Stojmenovic, 2001] Lin, X. and Stojmenovic, I. (2001). Loop-free hybrid single-path/flooding routing algorithms with guaranteed delivery for wireless networks. *IEEE Transactions on Parallel and Distributed Systems*, 12(10):1023–1032.

[Malpani et al., 2000] Malpani, N., Vaidya, N., and Welch, J.L. (2000). Leader election algorithms for mobile ad hoc networks. In *DIALM for Mobility 2000, pp. 96-103*.

[Malpani et al., 2001] Malpani, N., Vaidya, N., and Welch, J.L. (2001). Distributed token circulation in mobile ad hoc networks. In *Proceedings of the 9th International Conference on Network Protocols (ICNP)*.

[Myoupo, 2003] Myoupo, J.F. (2003). Dynamic initialization protocols for mobile ad hoc networks. In *11th IEEE Intern. Conf. on Networks (ICON2003), Sydney, Australia, pp. 149-154*.

[Myoupo et al., 2003] Myoupo, J.F., Ravelomanana, V., and Thimonier, L. (2003). Average case analysis based-protocols to initialize packet radio networks. *Wireless Communication and Mobile Computing*, 3:539–548.

[Raymond, 1989] Raymond, K. (1989). A tree-based algorithm for distributed mutual exclusion. *ACM Trans. Comput. Systems*, 7:61–77.

[Vaidya et al., 2001] Vaidya, N.H., Walter, J.E., and Welch, J.L. (2001). A mutual exclusion algorithm for ad hoc mobile networks. *Wireless Networks*, 7(6):585–600.