# AN ANALYTICAL STUDY OF MAC PROTOCOLS FOR UNDERWATER ACOUSTIC SENSOR NETWORKS


Laureanda: *Beatrice Tomasi*

Relatore: Ch.mo Prof. Michele Zorzi

Correlatore: Dott. Ing. Paolo Casari



**Laurea in Ingegneria delle Telecomunicazioni**

Anno Accademico 2007/2008

7 Ottobre 2008

# Table of contents

**Sommario**

Questo lavoro si focalizza su un campo di ricerca innovativo ed insolito: le reti di sensori sottomarine. L'ambito di studio risulta particolarmente affascinante per la presenza di una varietà di problemi aperti. L'utilità di una rete di sensori sottomarini risiede nella capacità di misurare alcune grandezze fisiche dell'acqua e dei fondali. Queste misurazioni permettono di supportare un'ampia varietà di applicazioni che spazia dal rilevamento di agenti inquinanti al monitoraggio delle correnti sottomarine per la predizione di eventi atmosferici, dalla sorveglianza del territorio allo studio dei comportamenti della fauna marina. Uno spettro di applicazioni così ampio ha fatto crescere l'interesse della comunità scientifica verso questa tecnologia.

In questa tesi viene studiato ed analizzato il problema del controllo di accesso al mezzo in reti acustiche sottomarine. Le peculiari condizioni di propagazione acustica limitano l'efficacia di alcuni paradigmi già proposti ed applicati in reti di sensori terrestri.

Lo scopo di questo lavoro è contribuire al processo di individuazione delle più efficienti strategie di accesso al mezzo. Ciò viene effettuato dando una rappresentazione stocastica al comportamento di un nodo e confrontando le prestazioni di rete, che vengono dedotte applicando le metodologie acquisite nel percorso di studi.

I protocolli analizzati in questa tesi sono Slotted FAMA e Tone-Lohi. Anche se entrambi prevedono una tecnica di accesso a contesa, evidenziano sostanziali differenze sia nel comportamento sia nelle prestazioni. Tali aspetti verranno descritti e analizzati in dettaglio per via analitica in questa tesi.

**Abstract**

This work aims at providing some insight into an innovative research field: underwater sensor networks. This field is particularly challenging due to the presence of a number of open problems. The main purpose of an underwater sensor network is to monitor some physical parameters of water and seabeds. These measurements allow to support many applications, ranging from detection of environmental pollution to underwater flows monitoring and from surveillance to the study of marine animals. These are some of the reasons why this line of research is growing momentum.

In this thesis the problem of designing an efficient medium access control (MAC) protocol is studied and analyzed. The specific channel characteristics limit the the effectiveness of radio access paradigms suitable for terrestrial radio networks.

The main aim of this study is to contribute to devising a good MAC strategy for underwater sensor networks. This is done by analyzing two MAC protocols by means of semi-Markov models and by comparing the corresponding network performance.

The analyzed protocols are Slotted Floor Acquisition Multiple Access and Tone-Lohi. Even if both are random access protocols, they differ a lot in strategy and performance; all these aspects are analyzed and described in the following.

# Chapter 1

# Introduction

Underwater sensor networks (USN) represent a new challenge for Information Technology researchers. More and more applications have been thought for USN (for instance: seabed monitoring, oilfields searching, security, tsunami preventing ), but the employment of such networks faces many practical problems such as high costs, high energy consumption, long propagation delays, frequency-dependent noise and small bandwidth. All these specific aspects require a novel communication paradigm, since terrestrial sensor networks protocols are usually unable to perform as well as in radio networks when straightforwardly applied to underwater networks. For instance, medium access protocol for USN should focus on avoiding collisions, because the transmission power is typically four orders of magnitude greater than the power used by terrestrial sensors in radio transmissions. Therefore, packet collisions would cause a great amount of energy to be wasted, thus seriously affecting the lifetime of the system.

The main aim of this work is to develop a stochastic analysis of MAC protocols for USN using semi-Markov models, in order to infer some important network metrics such as throughput, energy consumption and packet delay without need to simulate.

The first chapter presents an overview of the mathematical framework employed in the analysis: stochastic Markov processes are introduced and the theorem that establishes the equivalence between stationary and limiting distribution is recalled. The final part covers renewal theory, the most important result being the fundamental renewal theorem, by means of which the asymptotic performance of the system can be predicted.

The second chapter is devoted to the state of the art in designing MAC protocols for USN. Although many protocols have been proposed during the last ten years, ranging from Code Division Multiple Access, Carrier Sense Multiple Access/Collision Avoidance, Aloha and others, only random MAC protocols are considered in this chapter.

The third chapter covers the channel model and the acoustic propagation and summarizes the specific aspects of the underwater environment analyzed so far.

The fourth and the fifth chapters presents two protocols, Slotted Floor Acquisition Multiple Access (S-FAMA) and Tone-Lohi (T-Lohi): a model for each protocol is given thereafter.

The last chapter concludes with the main results of the research.

# Chapter 2

# Stochastic Modeling

## 2.1 Markov Process

A brief summary of stochastic modeling theory is given in the following, starting from the definition of Markov process, in order to provide the necessary material to understand the analytical tecniques applied in the following chapters.

**Definition 2.1.1** (Markov Property)**.** *A Markov process $X(t)$ is a stochastic process which satisfies the Markov property which establishes that the value assumed by the process in the future, given its present value, is not influenced by the past behavior.*

*If $k > s$ and $u < s$, the property can be expressed as:*

$$P[X(k) = j|X(s) = m, X(u) = p] = P[X(k) = j|X(s) = m]$$

If the Markov process assumes values from a finite or countable set of states, whose time index set is $(0, 1, 2, \dots)$, it is called a discrete-time Markov chain, and the Markov property becomes:

$$P[X_{n+1} = b|X_0 = a_0, \dots, X_n = a] = P[X_{n+1} = b|X_n = a].$$

The probability that $X_{n+1} = b$, given that $X_n = a$ is called *one-step transition probability* and it is denoted by $P_{ab}^{n,n+1}$. This probability in general depends on the initial and final states, and on the transition time. In the case the one-step transition probability is independent of the transition time, it is said that the Markov chain has *stationary transition probabilities* and the apex $n, n+1$ can be dropped. From now on, transition probabilities are assumed to be stationary.

The quantities $P_{ab}$ are organized in a square matrix called the *transition probabilities matrix* of the process. In general the matrix has a infinite range, but supposing the states set is finite, with $N+1$ states, the transition probabilities matrix is expressed as:

$$\mathbf{P} = \begin{pmatrix} P_{00} & P_{01} & \cdots & P_{0N} \\ P_{10} & P_{11} & \cdots & P_{1N} \\ \vdots & \vdots & \vdots & \vdots \\ P_{N0} & P_{N1} & \cdots & P_{NN} \end{pmatrix}$$

The transition probabilities, in each row, satisfy the conditions:

$$P_{ab} \geq 0 \tag{2.1}$$

$$\sum_{b=0}^{N} P_{ab} = 1 \quad \text{for } a = 0, 1, \ldots, N \tag{2.2}$$

$$\tag{2.3}$$

**Proposition 2.1.1.** *A Markov process is completely defined once its transition probability matrix and the probability distribution of the initial state are specified.*

**Theorem 2.1.1.** *The n-step transition probabilities of a Markov chain satisfy*

$$P_{ab}^{(n)} = \sum_{k=0}^{\infty} P_{ak} P_{kb}^{(n-1)} \tag{2.4}$$

*where* $P_{ab}^{(0)}$ *is defined as*

$$P_{ab}^{(0)} = \begin{cases} 1 & \textit{if } a = b \\ 0 & \textit{if } a \neq b \end{cases}$$

The proofs of both the theorem and the proposition are provided in Appendix A.

## 2.2 First Step Analysis

To explain how the first step analysis works, a simple example is provided; in the following, by means of this method it is shown how it is possible to derive, for instance, the mean latency in delivering a data packet.

Let's consider the transition probability matrix:

$$\mathbf{P} = \begin{pmatrix} 1 & 0 & 0 \\ \alpha & \beta & \gamma \\ 0 & 0 & 1 \end{pmatrix}$$

where the state set is $\{0, 1, 2\}$ and where $\alpha$, $\beta$, $\gamma$ are strictly positive quantities and $\alpha+\beta+\gamma = 1$. If the initial state is 0, then the process remains in this state, similarly it happens for state 2; while supposing the process is in state 1, then it can either move to states 0 or 2 and being absorbed there, or return to state 1, and the same procedure can be repeated. So state 0 and 2 are called absorbing states.

Let us consider the following problems. Assuming that 1 is the initial state, how long, on average, does it takes to reach state 0 or 2? What is the probability that the process will absorbed by 0 or 2? In order to find the answers to these questions, it is necessary to formulate them in formal terms. Defining $T$ as the minimum number of steps required to reach one of the two absorbing states:

$$T = \min\{n \geq 0 : X_n = 0 or X_n = 2\}$$

hence the two questions above can be written as:

$$u = P[X_T = 0|X_0 = 1]$$
$$v = E[T|X_0 = 1]$$

where $E[x]$ is the expectation of $x$.

Considering separately the cases that $X_1 = 0$, $X_1 = 1$ and $X_1 = 2$, the first step probabilities are respectively respectively $\alpha$, $\beta$ and $\gamma$. If $X_1 = 0$ then $T = 1$ and $X_T = 0$. If $X_1 = 1$ then $X_T$ can be either 0 or 1 or 2, and the probability that the process will be trapped in state 0 is again $u$. Then using the law of total probability and the Markov property, we get:

$$\begin{aligned} u &= P[X_T = 0|X_0 = 1] \\ &= \sum_{k=0}^{2} P[X_T = 0|X_0 = 1, X_1 = k]P[X_1 = k|X_0 = 1] \\ &= \sum_{k=0}^{2} P[X_T = 0|X_1 = k]P[X_1 = k|X_0 = 1] \\ &= 1 \cdot \alpha + u \cdot \beta + 0 \cdot \gamma \end{aligned}$$

Solving the equation yields:

$$u = \frac{\alpha}{1-\beta} \tag{2.5}$$

The average time for absorption in state 0 or in state 2 is at least 1, indeed if $X_1 = 0$ or $X_1 = 2$ then $T = 1$, otherwise ($X_1 = 1$) the process restarts and the average absorption time is again $v$:

$$\begin{aligned} v &= E[T|X_0 = 1] \\ &= 1 + 0 \cdot \alpha + v \cdot \beta + 0 \cdot \gamma \end{aligned}$$

$$v = \frac{1}{1-\beta}$$

We note that first step analysis is a simple procedure: absorption probabilities and mean time to absorption can be estimated straightforwardly. Moreover this method can be applied to calculate the average return time. Calling $\theta_{ij}$ the number of steps from $i$ to $j$, $\theta_{ij} = 1$ with probability $P_{ij}$, otherwise $\theta_{ij} = \theta_{kj}$ steps with probability $P_{ik}$. In formal terms:

$$\theta_{ij} = \begin{cases} 1 & \text{with probability } P_{ij} \\ \\ 1 + \theta_{kj} & \text{with probability } P_{ik} \quad k \neq j \end{cases}$$

Therefore the mean is given by:

$$E[\theta_{ij}] = P_{ij} + \sum_{k \neq j} P_{ik}(1 + E[\theta_{kj}]) \tag{2.6}$$

$$= 1 + \sum_{k \neq j} P_{ik} E[\theta_{kj}] \quad \forall i, j \tag{2.7}$$

Hence, in a Markov chain with $N$ states, a linear system of $N$ equations and $N$ unknowns can be written for all $i$ and $j$. To estimate the mean number of steps to return in state $i$ ($E[\theta_{ii}]$), it is sufficient to consider $j = i$.

**Example:** Consider the following 2D-transition probabilities matrix, where the set of states is $0, 1$

$$\mathbf{P} = \begin{pmatrix} 1-a & a \\ b & 1-b \end{pmatrix}$$

We want to estimate the mean return time to state 0 by using the equation (2.6).

$$E[\theta_{00}] = 1 + \sum_{k \neq 0} P_{0k} E[\theta_{k0}]$$

$$= 1 + P_{01} E[\theta_{10}]$$

We need to know the value of $E[\theta_{10}]$, the mean number of steps from state 1 to 0, using the first step analysis:

$$E[\theta_{10}] = 1 + P_{11} E[\theta_{10}]$$

$$E[\theta_{10}] = \frac{1}{b}$$

That leads to

$$E[\theta_{00}] = \frac{a+b}{b}$$

In order to estimate the return time ($\tau_{ij}$) from $i$ to $j$, given the average duration of a sojourn in state $i$ ($T_i$), one can use a similar tecniques by posing:

$$t_{ij} = \begin{cases} T_i & \text{with probability } P_{ij} \\ \\ T_i + t_{kj} & \text{with probability } P_{ik} \quad k \neq j \end{cases}$$

Therefore (2.6) can be rewritten as:

$$t_{ij} = T_i + \sum_{k \neq j} P_{ik} t_{kj} \quad \forall j \tag{2.8}$$

Equation (2.8) can be written in matrix form: call $\mathbf{t}_j$ . the vector of average times from state $k$ to $j$, for all $k$, $\mathbf{T}_0$ the vector of durations of state, and $\mathbf{P}_j$ the transition matrix with zeros in its $j$-th column. Therefore:

$$\mathbf{t}_j = \mathbf{T}_0 + \mathbf{P}_j \ \mathbf{t}_j.$$

The solution of the equation is hence

$$\mathbf{t}_j = \mathbf{T}_0 (\mathbf{I} - \mathbf{P}_j)^{-1} \tag{2.9}$$

## 2.3 The Asymptotic Behavior of a Regular Markov Chain

In this section it is justified why in a regular Markov chain the asymptotic behavior coincides with the stationary behavior, and how much independent of the initial state it is.
The definition of *regular* Markov chain is provided and the theorem that establishes the equivalence between asymptotic and stationary behavior is enunciated and proved.

**Definition 2.3.1.** *A Markov chain with a finite number of states is regular if the transition probability matrix* $\boldsymbol{P}$*, raised to some power* $k$*, has only strictly positive* *elements.*

The consequence is that a *limiting probability distribution*, defined in equation[1] (2.10)exists and this will be proved in the Appendix A.

$$\Pi_j = \lim_{n\to\infty} P_{ij}^{(n)} \tag{2.10}$$

The long-run probability distribution $\Pi = \{\pi_0, \pi_1, \ldots, \pi_N\}$ is strictly positive for each state and it is independent of the initial state. A regular markov chain can also be defined as a irreducible, aperiodic and recurrent[2].

**Theorem 2.3.1.** *Let* $\boldsymbol{P}$ *be a* $(N+1)\times(N+1)$ *regular transition probability matrix. Then the limiting probability distribution* $\Pi = (\pi_0, \cdots, \pi_N)$ *is the unique nonnegative solution of*

$$\Pi = \Pi \cdot \boldsymbol{P} \tag{2.11}$$

$$\Pi \cdot \boldsymbol{1} = 1 \tag{2.12}$$

*where* $\boldsymbol{1}$ *is a row vector whose elements are all ones.*

*Proof.* The result in Theorem 2.1.1 leads to:

$$P_{ij}^{(n)} = \sum_{k=0}^{N} P_{ik}^{(n-1)} P_{kj} \quad \forall i,j \tag{2.13}$$

Applying the limit in both sides:

$$\lim_{n\to\infty} P_{ij}^{(n)} = \lim_{n\to\infty} \sum_{k=0}^{N} P_{ik}^{(n-1)} P_{kj} \quad \forall i,j \tag{2.14}$$

Since the sum is finite, it is possible to exchange the order of the sum and limit to yield:

$$\lim_{n\to\infty} P_{ij}^{(n)} = \sum_{k=0}^{N} \lim_{n\to\infty} P_{ik}^{(n-1)} P_{kj} \quad \forall i,j \tag{2.15}$$

Substituting (2.10) in (2.15), we have:

$$\Pi_j = \sum_{k=0}^{N} \Pi_k P_{kj} \quad \forall 0 \le j \le N \tag{2.16}$$

[1] $P_{ij}^{(n)}$ is the $n$-steps transition probability

[2] see Appendix A

Hence the distribution $\Pi$ is a solution of equation (2.11). Given that the $i$-th row of a n-step transition probability matrix is the probability distribution of the event $\{X_n = j | X_0 = i\}$, then this distribution sums to 1:

$$\sum_{k=0}^{N} P_{ik}^{(n)} = 1 \tag{2.17}$$

Therefore, in the long-run,

$$\lim_{n\to\infty} \sum_{k=0}^{N} P_{ik}^{(n)} = \lim_{n\to\infty} 1 \tag{2.18}$$

which leads to:

$$\sum_{k=0}^{N} \lim_{n\to\infty} P_{ik}^{(n)} = 1 \tag{2.19}$$

$$\sum_{k=0}^{N} \Pi_k = 1 \tag{2.20}$$

We conclude that $\Pi$ is a solution of the equation (2.12). We have to prove that $\Pi$ is the unique solution as well. Let suppose that another solution to both equations (2.11) and (2.12) exists and we call it $X = \{X_0, X_1, \ldots, X_N\}$:

$$X_j = \sum_{k=0}^{N} X_k P_{kj} \quad \forall 0 \le j \le N \tag{2.21}$$

By multiplying both sides by $P_{jl}$ and summing over $j$

$$\sum_{j=0}^{N} X_j P_{jl} = \sum_{j=0}^{N} \sum_{k=0}^{N} X_k P_{kj} P_{jl} \tag{2.22}$$

Swapping the sums in the right-hand side:

$$\sum_{k=0}^{N} \sum_{j=0}^{N} X_k P_{kj} P_{jl} = \sum_{k=0}^{N} X_k P_{kl}^{(2)} \tag{2.23}$$

Noting that the left-hand side in equation (2.22) is $X_l$ we have:

$$X_l = \sum_{k=0}^{N} X_k P_{kl}^{(2)} \tag{2.24}$$

Repeating the argument we get,

$$X_l = \sum_{k=0}^{N} X_k P_{kl}^{(n)} \quad \forall n \tag{2.25}$$

By applying the limit for $n \to \infty$ in both sides:

$$X_l = \lim_{n\to\infty} \sum_{k=0}^{N} X_k P_{kl}^{(n)}. \tag{2.26}$$

This leads to:

$$X_l = \sum_{k=0}^{N} X_k \Pi_l \tag{2.27}$$

Since $X_l$ is a solution of the equation (2.12) we have proven that $\Pi$ is also the unique solution of the system.

$$X_l = \Pi_l \quad 0 \leq \forall l \leq N \tag{2.28}$$

□

The solution to equations (2.11) and (2.12) is the stationary distribution, hence we have proved that the limiting probability distribution is equal to the stationary distribution.

## 2.4 An Overview of semi-Markov processes

### 2.4.1 Renewal Processes

**Definition 2.4.1.** *A renewal process consists of:*

- *a counting process called* $\{N(t), t \geq 0\}$, *it is a nonnegative stochastic process which assumes integer values and counts how many times an event occurs in the interval time* $(0, t]$

- *a sequence of identically distributed random variables* $(x_i)$, *which represents time interval between consecutive events* $(e_{i-1}, e_i)$ *which are positive, independent.*

*Hence the probability distribution can written as:*

$$F(\tau) = P[x_i \leq \tau] \quad i = 1, 2, \ldots \tag{2.29}$$

We call $w_n$ the time elapsed from the first until the $n$-th event's occurrence. Formally:

$$w_n = x_1 + x_2 + \cdots + x_n \tag{2.30}$$

**Example:** A Poisson process with parameter $\lambda$ can be seen as a renewal process since:

- $N(t)$, the counting process which registers the number of arrivals in the interval $(0, t]$ is nonnegative and integer-valued

- the inter-arrival time are independent, identically distributed according to the exponential distribution $F(x) = 1 - e^{-\lambda x}, \quad x \geq 0$

We call $M(t)$ the expectation of the counting process:

$$M(t) = E[N(t)] \tag{2.31}$$

We will prove some asymptotic conditions concerning the counting process $N(t)$ and the inter-arrival times $x_k$.

**Proposition 2.4.1.**

$$P[\lim_{t\to\infty} N(t) = \infty] = 1 \tag{2.32}$$

*Proof.*

$$\begin{aligned} 0 \leq P[\lim_{t\to\infty} N(t) < \infty] &= P[\cup_{n=1}^{\infty}\{X_n = \infty\}] \\ &\leq \sum_{n=1}^{\infty} P[\{X_n = \infty\}] \\ &= 0 \end{aligned}$$

Hence $P[N(\infty) = \infty] = 1$. □

**Proposition 2.4.2.** *Let $S_n$ be the sum of the first $n$ inter-arrival times. Thanks to the law of large numbers, we have:*

$$\frac{S_n}{n} \to E[x] \tag{2.33}$$

*Hence if $E[x] > 0$, then $\lim_{n\to\infty} S_n = \infty$ else if $E[x] = 0$, then $\lim_{n\to\infty} S_n < \infty$*

**Theorem 2.4.1.**

$$\lim_{t\to\infty} \frac{N(t)}{t} = \frac{1}{E[x]} \tag{2.34}$$

*Proof.*

$$S_{N(t)} \leq t \leq S_{N(t)+1} \tag{2.35}$$

Dividing for $N(t)$ all the elements of the inequations we get:

$$\frac{S_{N(t)}}{N(t)} \leq \frac{t}{N(t)} \leq \frac{S_{N(t)+1}}{N(t)} \tag{2.36}$$

Applying the limit $\lim_{t\to\infty}$, from the proposition 2.4.1 $\lim_{t\to\infty} N(t) = \infty$ with probability one, this leads to:

$$\lim_{t\to\infty} \frac{S_{N(t)}}{N(t)} \leq \lim_{t\to\infty} \frac{t}{N(t)} \leq \lim_{t\to\infty} \frac{S_{N(t)+1}}{N(t)} \tag{2.37}$$

Using a well known theorem we have that:

$$\lim_{t\to\infty} \frac{t}{N(t)} = E[x] \tag{2.38}$$

therefore:

$$P[\lim_{t\to\infty} \frac{N(t)}{t} = \frac{1}{E[x]}] = 1 \tag{2.39}$$

□

**Definition 2.4.2.** *Let $x_1, x_2, x_3, \ldots$ be a sequence of independent and identically distributed stochastic variables, an unknown $N$ is called a* stopping time *for the sequence $x_1, x_2, x_3, \ldots$ if $\{N = n\}$ is independent of $X_m$, with $m > n$.*

**Example:** Throwing a coin the contingencies are head or tail, and at the n-th flip the probabilities are:

$$P[X_n = T] = \frac{1}{2}$$
$$P[X_n = H] = \frac{1}{2}$$

So defining $N$ as:

$$N = \min\{n : X_1 + X_2 + \cdots + X_n = 10\} \tag{2.40}$$

so $N$ represents the instant in which the experiment is ended and this instant is independent of future process realizations, but it is only influenced by the present and past states of the process. Hence the process

$$N_1 = \min\{n : X_1 + X_2 + \cdots + X_n > 10\} \tag{2.41}$$

represents a stopping time for the sequence $X_i$, while the variable defined as:

$$N_2 = \min\{n : X_1 + X_2 + \cdots + X_n < 10\} \tag{2.42}$$

An important result is presented here in order to justify some following theorems.

**Theorem 2.4.2** (Wald's equation)**.** *Let $X_1, X_2, \ldots$ be a sequence of independent and identically distributed variables and let $E[X] < \infty$ the mean, if $N$ is a* stopping time *such that $E[N] < \infty$ then:*

$$E[\sum_{i=1}^{N} X_i] = [E[N]E[X]] \tag{2.43}$$

We notice that $N$ and $X$ are not independent variables.

*Proof.* Considering a renewal process such that $X_1, X_2, \ldots$ are independent identically distributed and their mean is finite, if occurs that $N(t) = n$ if and only if $X_1 + X_2 + \cdots + X_n \leq t$ and $X_1 + X_2 + \cdots + X_n + X_{n+1} > t$ then the process $N(t) + 1$ is a stopping time for the process of $X_1, X_2, \ldots$. Thanks to Wald's equation we have that:

$$E[S_{N(t)+1}] = E[\sum_{i=1}^{N(t)+1} X_i] = [E[N(t) + 1]E[X]] \tag{2.44}$$

To prove that the condition

$$E[S_{N(t)}] = E[\sum_{i=1}^{N(t)} X_i] = [E[N(t) + 1]E[X]] \tag{2.45}$$

□

**Theorem 2.4.3.** *Let consider a renewal process such that the sequence of interarrival times has finite mean, then:*

$$\lim_{t\to\infty} \frac{M(t)}{t} = \frac{1}{E[X]}$$

*Proof.* Let consider $t$ such that

$$t < S_{N(t)+1} \tag{2.46}$$

this yields to:

$$t < E[S_{N(t)+1}] = E[N(t)+1]E[X] = (M(t)+1)E[X] \tag{2.47}$$

this leads to:

$$\frac{M(t)}{t} > \frac{1}{E[X]} - \frac{1}{t} \tag{2.48}$$

This term can be seen as the $n$-th term of a non decreasing succession such that:

$$\lim_{t\to\infty} \inf\{\frac{M(t)}{t}\} \geq \frac{1}{E[X]} \tag{2.49}$$

We define the following stochastic variable:

$$X_i^c = \begin{cases} X_i & \text{if } X_i \leq c \\ \\ c & \text{if } X_i > c \end{cases}$$

The sequence $X_i^c$ is still a sequence of i.i.d. variables so the process is a renewal process. The new renewal process has shorter inter-event intervals than the renewal process with inter-event time $X_i$. We have:

$$t + c \geq S_{N^c(t)+1} \tag{2.50}$$

which leads to:

$$t + c \geq E[{X_i}^c][1 + M^c(t)] \tag{2.51}$$

where

$$E[{X_i}^c] = \int_0^c [1 - F(x)]\mathrm{dx}$$

Since the inter-arrival times are shorter, there are more events than in the original renewal process so:

$$N(t) \leq N^c(t)$$

$$M(t) \leq M^c(t)$$

This yields to, recall 2.51

$$t + c \geq E[X_i{}^c][1 + M^c(t)] \geq E[X^c][1 + M(t)] \tag{2.52}$$

Dividing by $t$ we have:

$$\frac{M(t)}{t} \leq \frac{1}{E[X^c]} + \frac{1}{t}[\frac{c}{E[X^c]} - 1] \forall c > 0 \tag{2.53}$$

For fixed $c$, we have that:

$$\lim_{t\to\infty} \inf\{\frac{M(t)}{t}\} \geq \frac{1}{E[X^c]} \tag{2.54}$$

and for $c \to \infty$ we have that $E[X^c] \to E[X]$, which leads to the equation (2.49). Similarly we have

$$\lim_{t\to\infty} \sup\{\frac{M(t)}{t}\} \leq \frac{1}{E[X^c]} \tag{2.55}$$

and that:

$$\lim_{c\to\infty} \lim_{t\to\infty} \inf\{\frac{M(t)}{t}\} \geq \lim_{c\to\infty} \frac{1}{E[X^c]} = \frac{1}{E[X]} \tag{2.56}$$

Since the two limits converge to the same value the statement is provided. □

Let $R_i$ be a sequence of stochastic variable i.i.d., each of them corresponds to a gain associated to the inter-event interval between the $i$-th and $(i+1)$-th event. We call $R(t)$ as:

$$R(t) = \sum_{n=1}^{N(t)+1} R_n \tag{2.57}$$

**Theorem 2.4.4.** *If $E[R]$ and $E[X]$ are finite values then*

$$\lim_{t\to\infty} \frac{R(t)}{t} = \frac{E[R]}{E[X]} \tag{2.58}$$

*with probability one.*

*Proof.* Starting from the equation (2.57), and dividing by $N(t)$ we get:

$$\frac{R(t)}{t} = \frac{\sum_{n=1}^{N(t)+1} R_n}{N(t)} \frac{N(t)}{t} \tag{2.59}$$

Applying the precedent results, passing to the limit for $t \to \infty$ we have:

$$\lim_{t\to\infty} \frac{R(t)}{t} = \lim_{t\to\infty} E[R] \frac{1}{E[X]} \tag{2.60}$$

□

### 2.4.2 The Semi-Markov Chain

In this section we define semi-Markov process and give some examples thereof . For a semi-Markov process, the following feature hold:

- the transition probability from state $i$ toward state $j$ is $P_{ij}$

- the duration of this transition has probability distribution $F_{ij}$

We define $Z(t)$ as the state of the semi-Markov process at time $t$, $H_i$ as the distribution of time in state $i$ before a transition ($H_i(t) = \sum_j F_{ij}P_{ij}$), and let $\mu_i$ be the mean time in $i$ before a transition:

$$\mu_i = E[H_i] = \int_0^\infty x \mathrm{d}H_i(x). \tag{2.61}$$

Finally $T_{ii}$ is the time between two consecutive transition in state $i$, and $\mu_{ii} = E[T_{ii}]$ is the mean time (these instants are renewal instants). A semi-Markov process is characterized by the embedded Markov chain obtained sampling the process at certain instants. Some properties referred to a semi-Markov process, such as the irreducibility or regularity, are hold only if the embedded Markov chain has these properties.

**Proposition 2.4.3.** *If the semi-Markov process is irreducible and $\mu_{ii}$ is finite, then the limiting distribution*

$$P_i = \lim_{t\to\infty} P[Z(t) = i | Z(0) = j]$$

*exists and it is independent of the starting state $j$ and:*

$$P_i = \frac{\mu_i}{\mu_i i} \tag{2.62}$$

*Proof.* Consider the interval between two consecutive visit to state $i$. Divide this interval into two parts. In the first part the process is still in $i$, in the second part the process is not in state $i$. We can see the process of being in state $i$ as a renewal process so the preceding results can be applied. In particular we want to determine the probability that at a certain instant $t$ the process is in state $i$. By theorem 2.4.4, we can write:

$$\lim_{t\to\infty} \frac{\text{time during which the process is in state } i}{t} = \frac{\mu_i}{\mu_{ii}} = \frac{E[Y]}{E[X]} \tag{2.63}$$

□

In the following we illustrate the method we used in the analytical study presented in Chapters 5 and 6. We define $r_{ij}$ as the cost associated to the transition from $i$ to $j$ and its average value as $E[r_{ij}] = R_{ij}$. Then the mean cost, which corresponds to the visit to state $i$ is:

$$R_i = \sum_k P_{ik} R_{ik} \tag{2.64}$$

as in first step analysis, we want to estimate the overall cost incurred from $i$ to $j$, when all possible transitions are considered. We call $\theta_{ij}$ the cost of the transition $i \to j$. Therefore:

$$\theta_{ij} = \begin{cases} r_{ij} & \text{with probability } P_{ij} \\ \\ r_{ik} + \theta_{kj} & \text{with probability } P_{ik} \quad k \neq j \end{cases}$$

By defining $\rho_{ij}$ as the average value of $\theta_{ij}$ and applying the linearity of expectation, we have for all $j$, :

$$\begin{aligned} \rho_{ij} &= E[\theta_{ij}] \\ &= P_{ij} E[r_{ij}] + \sum_{k\neq j} P_{ik} E[r_{ik} + \theta_{kj}] \\ &= P_{ij} R_{ij} + \sum_{k\neq j} P_{ik} R_{ik} + \rho_{kj} \\ &= \sum_k P_{ik} R_{ik} + \sum_{k\neq j} P_{ik} \rho_{kj} \\ &= R_i + \sum_{k\neq j} P_{ik} \rho_{kj} \end{aligned}$$

Multiplying both terms by the limiting distribution of the embedded Markov chain for state $i$, called $\Pi_i$, and summing over all states we get:

$$\sum_i \Pi_i \rho_{ij} = \sum_i \Pi_i R_i + \sum_i \Pi_i \sum_{k \neq j} P_{ik} \rho_{kj} \tag{2.65}$$

This leads to:

$$\Pi_j \rho_{jj} = \sum_i \Pi_i R_i \tag{2.66}$$

In case $r_{ij}$ represents the time spent in the transition from $i$ to $j$ and $\rho_{jj}$ is the mean time spent to return in state $j$ we obtain:

$$\Pi_j \mu_{jj} = \sum_i \Pi_i \mu_i \tag{2.67}$$

therefore:

$$\mu_{jj} = \frac{\sum_i \Pi_i \mu_i}{\Pi_j} \tag{2.68}$$

and recalling the (2.63):

$$P_j = \frac{\mu_j}{\mu_{jj}} = \frac{\mu_j \Pi_j}{\sum_i \Pi_i \mu_i} \tag{2.69}$$

If $r_{ij}$ any reward function, we have that:

$$\lim_{t \to \infty} \frac{R(t)}{t} = \frac{\sum_i \Pi_i R_i / \Pi_j}{\sum_i \Pi_i \mu_i / \Pi_j} = \frac{\sum_i \Pi_i R_i}{\sum_i \Pi_i \mu_i} \tag{2.70}$$

Substituting the average cost associated to a state with the average cost of each transition:

$$R_i = \sum_k P_{ik} R_{ik}$$

we get:

$$\lim_{t \to \infty} \frac{R(t)}{t} = \frac{\sum_i \Pi_i \sum_k P_{ki} R_{ki}}{\sum_i \Pi_i \sum_k P_{ki} T_{ki}} \tag{2.71}$$

Thanks to these results we have obtained a procedure to analyze the evolution of a system.

```
1) Find the semi-Markov representation of the system
2) Calculate the transition probability of the embedded Markov chain
3) Associate to each transition or state its mean cost
4) Calculate the limiting distribution of the Markov chain
5) Infer the mean metrics from equation (2.71)
```

To give an idea of the applications derived from this analysis, we present as an example, the *Go-Back-N* automatic retransmission request technique used to make systems reliable. Go-Back-N retransmission algorithm consists in sending an acknowledgment about the last correctly received packet.

**Example:** We assume that the communication channel introduces errors according to a Markov process, whose transition probability matrix is:

$$\mathbf{C} = \begin{pmatrix} P_{00} & P_{01} \\ P_{10} & P_{11} \end{pmatrix}$$

where state 0 and 1 represent respectively Good and Bad channel states. The feedback channel is error-free, but the acknowledgment arrives after three round trip times (RTT). We want to determine throughput. So we can represent this system through a semi-Markov model, such that:

- if the state of the channel is Good, the next can be Good with probability $P_{00}$

- if channel's state is Good, the next can be Bad with probability $P_{01}$

- if channel's state is Bad, after three RTT the state can be Good with probability $P_{10}^{(3)}$

- if channel's state is Bad, after three RTT the state can be Bad with probability $P_{11}^{(3)}$

Therefore the transition probabilities matrix of the embedded Markov chain is:

$$\mathbf{P} = \begin{pmatrix} P_{00} & P_{01} \\ {P^{(3)}}_{10} & {P^{(3)}}_{11} \end{pmatrix}$$

Then we associate a gain to the state which corresponds to successful packet delivery, and the number of RTTs spent in each state.

$$\text{number of successes} \begin{cases} 1 & \text{when in Good} \\ 0 & \text{when in Bad} \end{cases}$$

$$\text{number of RTT} \begin{cases} 1 & \text{when in Good} \\ m & \text{when in Bad} \end{cases}$$

Hence we calculate the limiting probability distribution $\Pi$, and we obtain throughput $(S)$:

$$\text{S} = \frac{\Pi_0 \cdot 1 + \Pi_1 \cdot 0}{\Pi_0 \cdot 1 + \Pi_1 \cdot m} = \frac{\Pi_0}{\Pi_0 \cdot 1 + \Pi_1 \cdot m} = \frac{P^{(3)}{}_{10}}{P^{(3)}{}_{10} + mP_{01}} \tag{2.72}$$

# Chapter 3

# The underwater channel

## 3.1 Path Loss in Underwater Channel

Path loss $A(d,f)$ is defined as the attenuation on a single path from the transmitter to the receiver. As part of the transmitted power $P_T$ is lost, mainly due to spreading loss and absorption loss, the receiving power is:

$$P_R = \frac{P_T}{A(d,f)} \tag{3.1}$$

Spreading loss is due to the distribution of the fixed amount of transmitted power over an increasingly larger surface area when the signal propagates away from the source. This kind of loss is expressed as a polynomial function of the distance between the source and the sink; its degree ($k$), called the spreading factor[1], depends on the geometry of the propagation. In shallow water waves spread with a cylindrical shape ($k = 1$) while in deep water waves experience spherical propagation ($k = 2$). In the study presented, the spreading factor is assumed to be $k = 1.5$, which corresponds to the so-called "practical propagation".

The absorption loss is due to the conversion of acoustic pressure into heat. The absorption loss is computed through the absorption coefficient $a(f)$ which is a function of signal frequency. This function has been proposed by Thorp [6],

[1] $k$ is measured in $[km^{-1}]$

who found a fitting function from empirical measurements obtained with low-frequency signals.

$$a(f) = 0.11\frac{f^2}{1+f^2} + 44\frac{f^2}{4100+f^2} + 2.75 \cdot 10^{-4} f^2 + 0.003. \tag{3.2}$$

The absorption coefficient for frequency above a few hundred Hz is presented in the equation (3.2), it is measured in [dB/km] while frequency is expressed in kHz.

Path loss is described by the equation (3.3) expressed in linear scale.

$$A(d, f) = d^k a(f)^d \tag{3.3}$$

where $d^k$ is the counterpart of radio attenuation in the air and $[a(f)]^d$ is call the absorption loss.

## 3.2 Noise in Underwater Channel

The main causes of noise in the underwater environment are turbulence, shipping, wind, and thermal noise. These major sources of noise are dominant in different bands of the signal spectrum. In Figure 3.1 the power spectral density (p.s.d.) of noise is depicted with different colors in order to stress this fact.

Noise p.s.d. is impacted by turbulence in the very low frequency region ($f \leq 10$ Hz), while it is influenced by shipping in the band from ten Hz to a hundred Hz. In the region from a hundred Hz to a hundred kHz, which is the operating region of the majority of acoustic system, the most important noise source is wind. Finally thermal noise, which is the noise due to electrical components in the receiver circuits, is dominant in the region beyond a hundred kHz. Each of these contributions are expressed in terms of power spectral density measured in

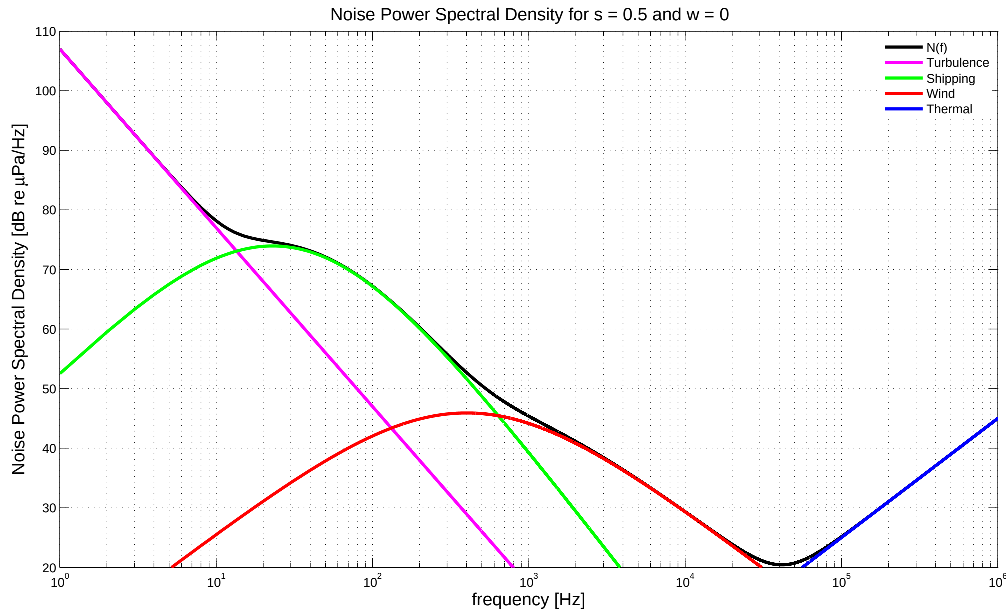


Figure 3.1: Power spectral density of noise in the underwater channel.

dB re $\mu$Pa/Hz.

$$N_t(f) = 17 - 30\log(f) \tag{3.4}$$

$$N_s(f) = 40 + 20(s - 0.5) + 26\log(f) - 60\log(f + 0.03) \tag{3.5}$$

$$N_w(f) = 50 + 7.5w^{1/2} + 20\log(f) - 40\log(f + 0.4) \tag{3.6}$$

$$N_{th}(f) = -15 + 20\log(f) \tag{3.7}$$

$$\tag{3.8}$$

## 3.3 *SNR* at the receiver and the packet probability

The signal to noise ratio at the receiver is obtained dividing the received power usually expressed in $\mu$Pa by the noise power in the signal band, as shown in equation (3.10).

$$SNR = \frac{P_R}{\int_{B(d)} N(f)df} \tag{3.9}$$

Assuming that the signal is transmitted in a sufficiently narrow band, the noise p.s.d. can be approximated with a constant value. Hence the $SNR$ becomes:

$$SNR = \frac{P_R}{B(d)N(f)} \tag{3.10}$$

Combining equation (3.1) with equation (3.10) we get:

$$SNR = \frac{P_T}{A(d,f)N(f)B(d)} \tag{3.11}$$

Drawing the quantity $\frac{1}{A(d,f)N(f)}$ it can be noticed that an optimal transmission frequency exists that maximizes $SNR$. This is depicted in Figure 3.2. In this study a piece-wise log-linear approximation of the optimal frequency and bandwidth derived in [21] is employed. Figure 3.3 shows the dependence on distance of both the optimal communication frequency $f_0(d)$ and the bandwidth $B(d)$, defined as $B(d) = \{f : SNR(d,f) \geq SNR(d,f_0)/2\}$. Since the actual frequency and bandwidth values are obtained by lengthy numerical integrations, a piece-wise log-linear approximation for both $f_0(d)$ and $B(d)$ is employed. This approximation is obtained by calculating the coefficient $\phi_0, \phi_1$, of the tangent lines in the actual trend of $f_0(d)$ and $B(d)$ in logarithmic scale, hence frequency (in logarithmic scale) is given by:

$$f = \phi_0 + \phi_1 d \tag{3.12}$$

where $d$ is in logarithmic scale too. The accuracy of these approximations is shown in Figure 3.3.

From the signal-to-noise ratio we can obtain the bit error probability $P_b$, assuming a particular modulation technique. In our study we assume a simple modulation such as Binary Phase Shift Keying (BPSK) which associate a phase to each bit. Given that $P_b$ is the bit error probability we get the packet error probability as:

$$P_e = (1 - P_b)^L \tag{3.13}$$

where $L$ is the packet length.

The bit error probability is finally inferred from $SNR$, and considering a BPSK modulation:

$$P_b = 2\mathrm{erfc}(\sqrt{2SNR(l,f)})$$

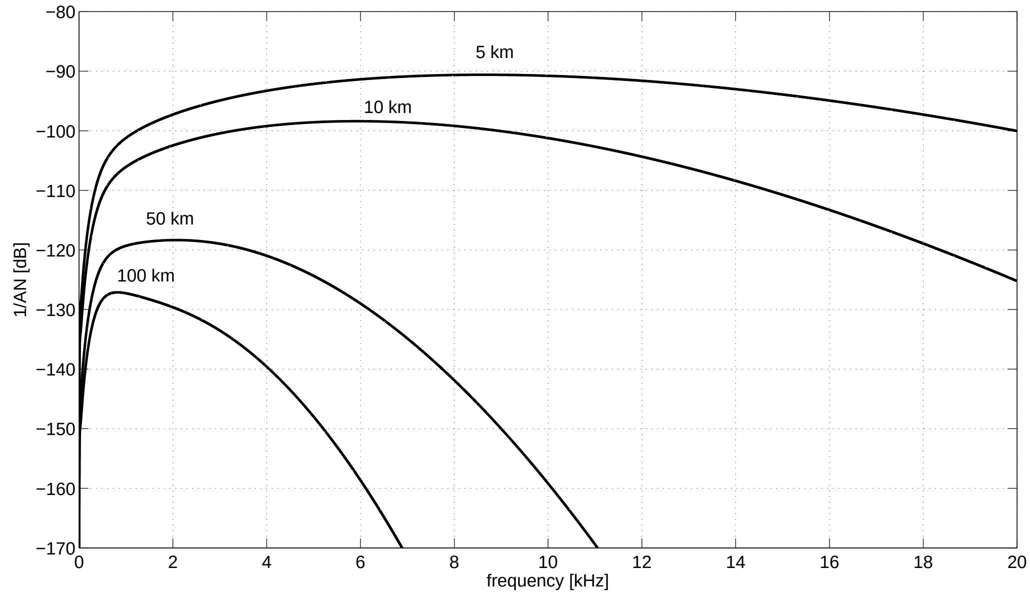


Figure 3.2: The quantity $(A(d,f)N(f))^{-1}$ drawn for some distances.

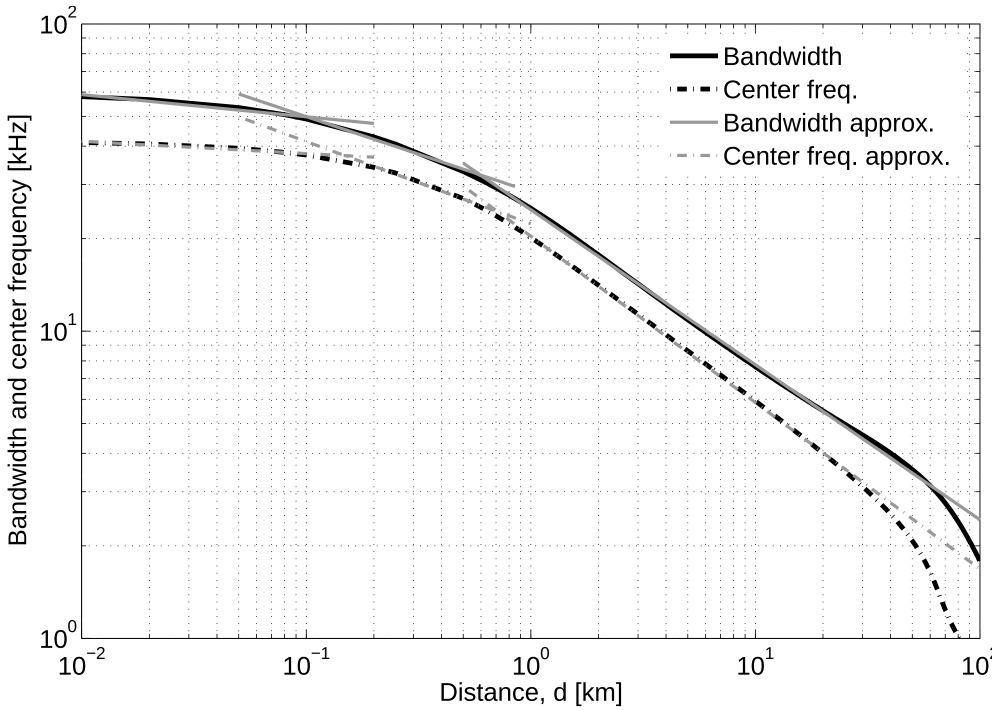


Figure 3.3: Piece-wise log-linear approximation of $B(d)$ and $f_0(d)$ as a function of distance.

## 3.4 The underwater acoustic propagation

Another problem underwater communications schemes have to face is the propagation speed of the acoustic waves: it is typically $1,5$ km/s, five orders of magnitude smaller than the propagation speed of electromagnetic waves in the air ($300,000$ km/s), actually the speed of sound is a function of salinity, depth and temperature. The low speed causes long propagation delays and therefore uneven access performance form distant nodes, which are reached by the acoustic wave at very different times. This phenomenon has to be considered when designing network protocols: for example employing channel sensing seems not to be a good strategy, since a packet could not be heard even if it has already been sent, due to this long propagation delays.

# Chapter 4

# State of the Art

The task of designing an efficient MAC protocol for underwater networks faces several issues mainly due to the specific characteristics of the channel, such as distance-dependent bandwidth, long propagation delay, and all propagation phenomena which have not been deeply understood yet. The consequence is that there is not a know-how, so far. The ideal protocol would control the access to the medium and be efficient in terms of bandwidth utilization, packet delay and energy consumption: actually designing such a protocol is not trivial, and requires to investigate a number of solutions via analysis and simulations. In order to give an idea of the ongoing research efforts on MAC protocol design for USNs, we present hereafter a survey of the protocols proposed so far.

## 4.1 Deterministic MAC Protocols

Deterministic MAC protocols reserve resources for each user: on one hand the advantage is that collisions are completely avoided, on the other hand the resources are wasted whenever the users do not use them.

Time Division Multiple Access (TDMA) is a channel access technique which consists in dividing time into frames; each frame is then partitioned into slots, each of them assigned to a user. Given the slot duration, the more users are present, the longer the time frame is. With $m$ nodes, an arriving packet has to wait for an average of $m/2$ slots for its turn to transmit. This means that packet delays are quite long, since in underwater networks the propagation delays are five order of magnitude longer than radio propagation delays, therefore time slot needs to be much longer in underwater than radio communication. On the one hand, TDMA ensures the absence of collisions, which means that no transmission power is wasted, on the other hand nodes have to be slot-synchronous, which requires distributed clock reference.

Frequency Division Multiple Access (FDMA) is a channel access technique which reserves a fraction of the available bandwidth for each user. In the underwater channel the available bandwidth depends on the distance between the source and the receiver sink, therefore partitioning the bandwidth among nodes may be a very difficult process. Moreover, due to the small amount of this resource, this technique does not seem a good solution.

Code Division Multiple Access (CDMA), consists in assigning a spreading different code to each user, such that simultaneous transmissions does not interfere with each other. It is realized by using modulation techniques such as Direct Sequence Spread Spectrum, and Frequency Hopping. However to implement these techniques in underwater scenarios, one has to account for specific channel features such as distance-dependent bandwidth available and the strong multipath phenomena.

An analytical study of these techniques as applied to underwater networks is currently missing, hence a contribution to this subject could provide an in-depth understanding of the trade-offs of these arising techniques when they are applied to this new environment.

In order to give some insight into the proposed solutions which use these techniques, consider the example in [19]. In this work, an application of deterministic access techniques is analyzed via simulation. The proposed scenario is a clustered network, which consists of some groups of sensors. Sensors belonging to a cluster are allowed to communicate with a cluster-head by means of Aloha or TDMA. In turn these cluster-heads relay the packets to an off-shore sink by means of access techniques such as Aloha, FDMA and CDMA, the latter with greater spreading factors in order to protect these more critical links from interference.

## 4.2 Random MAC Protocols

Random MAC protocols are medium access techniques that allow nodes to reserve the channel only when they have packets to send. This way the system's resources are not wasted, but collisions may occur.

In [18] some MAC protocols are analyzed and compared via simulations. This section provides a description of the proposed protocols: Aloha, slotted Aloha, Distance Aware Collision Avoidance Protocol (DACAP), Adaptive Propagation-delay-tolerant Collision Avoidance protocol (APCAP) and Propagation-Delay-Aware Protocol (PDAP).

The Aloha protocol allows nodes to transmit whenever they have packets to send. If no acknowledgment message is received, a collision is assumed to have taken place. Hence, after a random back-off time, the transmitter attempts to access the channel again. In a modified version of Aloha, namely slotted Aloha, the access to the channel is allowed only in particular instants and a global synchronization is still required.

The DACAP protocol uses Request-to-Send (RTS) and Clear-to-Send (CTS) to reserve the channel. Moreover it assumes that all nodes share the same global time reference. Collisions are avoided by means of some *warning* packets sent by the receiver to the transmitter. If a node overhears a control packet not meant for itself, after it has received an RTS, it notifies its transmitter, which delays the data delivery. This further time requested to avoid the collision depends on the distance between the source and the sink, which is measured by means of the Round Trip Time of control packets: for this reason this protocol is called "Distance Aware". The transmission is also delayed if the transmitter hears other control packets after sending the RTS.

APCAP applies the RTS/CTS scheme too, and assumes that all nodes are synchronized. The procedure used to avoid collisions is based on the deferral of both CTS and data packet transmissions. The transmitter establishes a window during which it can receive the CTS. The CTS packet brings the information about the instant at which the transmitter is allowed to deliver data.

PDAP is based on the same RTS/CTS mechanism and all nodes are synchronized. When a node has data to send, it schedules the time required for the whole communication (exchange of control packets and the packet transmission): this interval is known because nodes are propagation-delay-aware. Then it defers the communication's beginning of a random amount of time in order to avoid simultaneous transmissions by other nodes. Both the RTS and CTS contain an estimate of the distance between transmitter and receiver and this quantity is updated at each communication between nodes. When the transmitter sends the RTS, the neighboring nodes set their Network Allocation Vector (NAV) such that their scheduled communications will not collide with the occurring one. The receiver's neighbors do the same when they receive the CTS. This way, a great number of parallel communications are allowed without using an aggressive strategy, unlike in APCAP.

In the following, we show the access channel algorithm for a node that operates according to one of these protocols. In particular, for each protocol, we provide both reliable and unreliable version.

| DACAP Transmitter Algorithm |
|---|
| `1)Listen to the channel` |
| If a data packet arrives from the upper layer do: |
| `2)send an RTS and listen for the CTS` |
| `3)after receiving the CTS, wait for a predetermined time` $T_w$ |
| `4)if during this waiting time other nodes' control packets` |
| `or a` *warning* `from the receiver are heard do 5) else do 6)` |
| `5)abort the packet transmission and return to 1)` |
| `6)send data packet` |
| If the protocol ensures reliability then: |
| `7)wait for the ack` |
| `8)if no ack is received return to 6) else return to 1)` |

| DACAP Receiver Algorithm |
|---|
| `1)Listen to the channel` |
| If an RTS is received do: |
| `2)Send immediately a CTS` |
| `3)Wait for data packet` |
| `4)If in the meanwhile other nodes' control packets are heard do 5)` |
| `else do 6)` |
| `5)transmit a` *warning* `and return to 1)` |
| `6)wait for the packet` |
| `7)if packet arrives receive it` |
| `else return to 1)` |
| If the protocol ensures reliability then: |
| `8)if packet has been correctly received send an ack` |
| `else return to 6)` |

All neighboring nodes set their NAV according to the traditional mechanism described for example in IEEE 802.11 standard.

| APCAP Transmitter Algorithm |
|---|
| `1)Listen to the channel` |
| If a data packet generated do: |
| `2)schedule the RTS transmission and set the` $CTS_{win}$ `and` $DATA_{win}$ |
| `3)send RTS containing the information about` $CTS_{win}$ `and` $DATA_{win}$ |
| `4)if CTS is received during` $CTS_{win}$`do 5)` |
| `else return to 2)` |
| `5)transmit data at the negotiated time carried by the CTS` |
| If the protocol assures reliability then: |
| `6)wait for the ack` |
| `7)if no ack is received return to 2) else return to 1)` |

| APCAP Receiver Algorithm |
|---|
| `1)Listen to the channel` |
| If an RTS is received do: |
| `2)if you are busy during the` $CTS_{win}$ `return to 1)` |
| `else:` |
| `3)reply with a CTS containing a good time to receive data` |
| `4)wait for the packet` |
| `7)receive packet if it arrives during` $DATA_{win}$ |
| `else return to 1)` |
| If the protocol assures reliability then: |
| `8)if packet has been correctly received send an ack` |
| `9)return to 1)` |

The transmitter's neighboring nodes set their NAV, such that they do not transmit packets during the $CTS_{win}$, while the receiver's neighboring nodes set their NAV so that they do not send signals during the $DATA_{win}$.

| PDAP Transmitter Algorithm |
|---|
| `1)Listen to the channel` |
| If a data packet arrives from the upper layer do: |
| `2)schedule the whole communication exchange` |
| `3)wait for a random time` |
| `4)send RTS containing the information` |
| `about the whole communication exchange` |
| `5)if CTS is received during the expected time do 6)` |
| `else return to 2)` |
| `6)transmit data at the negotiated time carried by the CTS` |
| If the protocol assures reliability then: |
| `7)wait for the ack` |
| `8)if no ack is received return to 2) else return to 1)` |

| PDAP Receiver Algorithm |
|---|
| `1)Listen to the channel` |
| If an RTS is received do: |
| `2)if you are busy during time indicated by the tx do 1)` |
| `else:` |
| `3)reply with a CTS` |
| `4)receive the packet` |
| If the protocol assures reliability then: |
| `8)if packet has been correctly received send an ack` |
| `9)return to 1)` |

As done in APCAP, neighboring nodes' NAV are set, so that they do not transmit during both CTS and data packet receiving.

## 4.3 Other MAC Protocols

The authors in [12] propose an energy efficient MAC protocol. The main aim is to minimize the energy consumption in order to make the life of sensors longer. A communication between nodes is established by means of a SYNC signal transmitted at the beginning of a cycle. Then the transmitter turns off its transceiver in order to save energy. This SYNC signal contains the value corresponding to the duration of the cycle period, in this way all listening nodes know when they have to wake up to listen to the transmission. Moreover it has been proved that, by setting the cycle period to be much longer than data transmission, the collision probability is small. Anyway, this kind of strategy requires that all nodes share a common clock reference, in [13] this requirement has been analyzed via simulation, and the authors proved that the re-synchronization must be periodically carried out. The synchronization of nodes consumes further energy and resources that are not used to deliver data. For these reasons the protocol is not very efficient.

Another approach has been studied in [11]. Here the authors propose an hybrid mechanism. They subdivide time in frames, each of them consists of $N = N_s + N_u$ slots: $N_s$ are slots dedicated to the scheduled access, $N_u$ to the unscheduled access. Through this strategy the protocol can adapt varying traffic conditions. However it is not clear from the paper how this adaptation can be actually implemented, therefore this research direction is still very open.

In the following, we provide a description and analysis of other two MAC protocols, this approach can be applied to some protocols here described in order to compare with each other and find out a good strategy to access the underwater channel.

# Chapter 5

# An analytical study of S-FAMA

This chapter and the following one are devoted to illustrate the main contribution of this thesis.
First the Slotted Floor Acquisition Multiple Access protocol is described and analyzed, second its performance is derived and eventually some relevant network results are given and explained. In the analysis carried on we choose to represent the behavior of a single node in order to infer the network performance. A different procedure would be to consider the whole system as a queue, with Poisson arrivals, service times dependent on channel access time and only one server[1] However, we note that this model allows t find throughput, but, not e.g., energy consumption. For this reason, we choose to approach the problem through semi-Markov models.

## 5.1 Some hypothesis on the chosen scenario

We have considered a distributed sensor network, where nodes are both transmitter and receiver and they operate in half-duplex[2] mode, moreover they are placed

[1] Poisson arrivals represent the generated traffic, deterministic service times are the duration of a packet transmission, including the propagation delay and the server represents the channel. It is the typical case of M|G|1 queues.

[2] Half-duplex means that they cannot receive and transmit at the same time

on the seabed according to a two-dimensional Poisson process with rate $\lambda_2$. We have fixed the number of nodes within a node's coverage area.

The traffic generated by a node is modeled as a Poisson process with rate $\lambda$, therefore the traffic generation is memory-less, it would be interesting consider other traffic models, such as an event driven traffic. This latter is used when sensors are applied to warn about the occurrence of an event[3].

Moreover we assume that a node can generate packets only during the idle mode, and that even if more than one packet arrive, only one is transmitted, whereas the others are discarded.

Another assumption is that backoffs times are sufficiently long to accommodate all the pendent transmission in the coverage area so that the network is never backlogged. Other networks parameters, such as the length of data and control packets, are defined after the description of the protocol.

The channel attenuation is inferred as described in 3.1, and the SNR is calculated as said in 3.3.

## 5.2 Protocol description

Floor Acquisition Multiple Access is a reservation-based MAC protocol which prescribes the exchange of RTS/CTS messages. In its original version [20], FAMA makes nodes communicate transmission requests and grants through RTSs and CTSs, transmit data and wait for a confirmation of correct reception (ACK). In case no CTS is received in response to an RTS, the transmitter backs off and reschedules a later attempt. The protocol also includes error control over the data packet by means of stop-and-wait ARQ with infinite retransmissions. FAMA also assumes that, in order to save energy, nodes are deaf during backoff intervals and that nodes transmit RTSs without listening to the channel. Let $\tau_{max}$ be the propagation delay required to reach the maximum coverage radius of a node, which is taken to be equal to $R_{max}$. Two necessary conditions for collision

[3] It is out of the scope of this work

avoidance are defined: *i*) the duration of the RTS packet must be longer than $\tau_{max}$, and *ii*) the duration of the CTS packet must be longer than the duration of RTS plus $2\tau_{max}$.

For underwater communications these two conditions impose strong limitations as they would require the transmission of very long control packets, with a dramatic loss of efficiency and a useless increase in power consumption. Moreover, recall that transmit power is significantly higher than receive or idle power in typical underwater modems hardware: this fact discourages long transmission times, severely limiting the use of the original version of FAMA in underwater networks.

A solution to this problem is proposed in [6], where the authors suggest to make nodes share a global time synchronization and divide time into slots, forcing any transmission to be initiated only at the beginning of a slot. In order to avoid collisions between this transmission and any neighboring network activity, the minimum duration of a slot, $T_s$ must include a guard interval, whose duration is at least the maximum propagation time within a given coverage range. Therefore, $T_s = T_{sig} + \tau_{max}$, where $T_{sig}$ is the duration of a control packet.

With this "slotted" version of FAMA (S-FAMA hereafter), when a node has a data packet to send, it transmits an RTS at the beginning of the next slot. During idle periods, the node is constantly listening to the channel. Depending on neighboring communications, any of the following messages may be received:

- an RTS addressed to itself

- an RTS addressed to another node

- a CTS addressed to another node

In the first case the node starts reception procedures according to the above handshake description; otherwise, in the second and third cases it refrains from transmission in order to allow ongoing communications to be completed correctly. In the following section, the stochastic model for S-FAMA will be described, starting from this idle mode.

### 5.2.1 The Stochastic Model for S-FAMA and transition probabilities

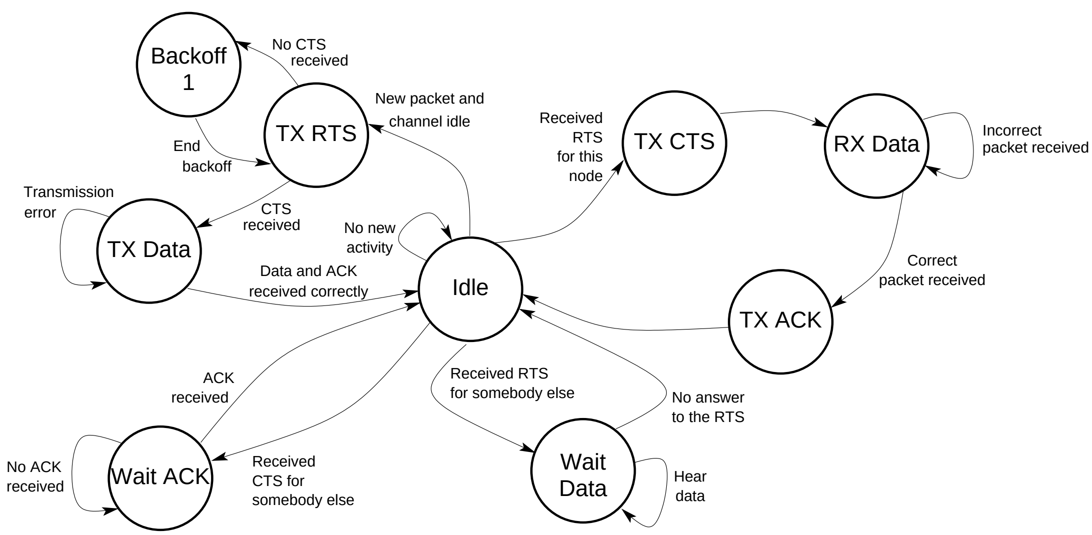


Figure 5.1: Embedded Markov chain that models the behavior of a node under the Slotted FAMA protocol.

In this section we track the evolution of a node by using a semi-Markov chain model and we calculate the transition probabilities. The current state of the node is sampled at the beginning of S-FAMA's time slots, so that the sequence of states is a Markov process. Figure 5.1 shows the embedded Markov chain.

While in Idle state, if the node has a data packet to send, it waits for the beginning of the next slot and transmits an RTS packet, which is received by all neighbors. In case the intended destination is not busy in other communications, it replies with a CTS packet at the beginning of the following slot. After a further time slot, if the source has received the CTS correctly, it starts to transmit the data packet, and backs off otherwise. The duration of the backoff is chosen to be a random number of slots uniformly chosen in the interval $[1, S_{max}]$, where $S_{max}$ is the number of slots that are required on average to transmit the packet correctly. Once the data packet has been sent, the source waits for the corresponding acknowledgement (ACK) to arrive in the following slot and applies stop-and-wait ARQ for error control as said before. If no ACK is received, the node re-transmits the whole data packet. Let us calculate the probabilities of these transitions:

$$P[\mathsf{Idle} \rightarrow \mathsf{TX\ RTS}] = P[\text{almost a pck is generated}|\text{idle}]$$
$$= 1 - e^{-\lambda T_s}$$

$$P[\mathsf{TX\ RTS} \rightarrow \mathsf{TX\ Data}] = P[\text{the CTS arrives at the node}|\text{it has sent the RTS}]$$
$$= 1 - P[\mathsf{TX\ RTS} \rightarrow \mathsf{Backoff\ 1}]$$

To derive the probability $P[\mathsf{TX\ RTS} \rightarrow \mathsf{Backoff\ 1}]$, each event which triggers a collision or a no-reply at the destination is listed below. When the test node transmits an RTS:

- the destination is receiving an RTS sent by one of its neighboring nodes

- the destination is receiving a CTS sent by a node inside the test node's transmission range

- the destination node is transmitting an RTS or it is replying with a CTS to a node outside the test node's transmission range

- the destination node is in backoff and cannot hear the RTS

All these events occur within the time slot during which the RTS is transmitted. Let us infer the probabilities of these contingencies. The first case pointed above occurs with rate $N\ \lambda$, which corresponds to the traffic generation rate of $N$ nodes.

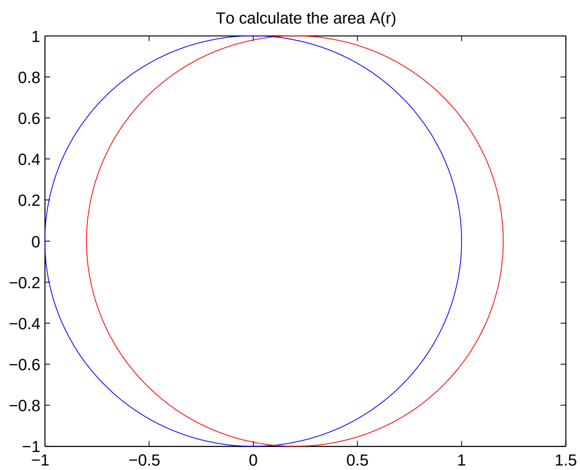


Figure 5.2: To calculate the area A(r).

In steady-state conditions, if a node becomes an RTS-transmitter with rate $\lambda$. Using a reciprocity argument, we can state that it is also a CTS-transmitter with rate $\lambda$. Nonetheless, only traffic originated outside the target node's transmission range must be considered, otherwise the same traffic would be counted twice.

Let $A(r)$ be the area of the union of two circles with the same radius R and distance between the centers r. With reference to Figure 5.2, this area can be expressed as:

$$A(r) = \pi R^2 + 2R^2 \arcsin\left(\frac{r}{2R}\right) + r\sqrt{R^2 - \left(\frac{r}{2}\right)^2}$$

Therefore the area outside the test node's transmission range but inside the destination's is

$$A(r) - \pi R^2, \tag{5.1}$$

hence the total rate at which the destination is transmitting an RTS or CTS can be found as:

$$\lambda_1\left(1 + \frac{A(r) - \pi R^2}{\pi R^2}\right) = \lambda_1 \frac{A(r)}{\pi R^2}.$$

Since both the node placement and the generated traffic are Poisson processes, hence the number of events which cause a collision or a no-reply at the destination is a Poisson random variable with average:

$$\begin{aligned}\Lambda &= \int_C \frac{1}{N}\frac{N}{\pi R^2}\lambda\Big(\frac{A(r)}{\pi R^2}+N\Big)T_s dA \\ &= \int_0^R \lambda\Big(\frac{A(r)}{\pi R^2}+N\Big)T_s\frac{2r}{R^2}dr\end{aligned}$$

Hence we get:

$$P[\mathsf{TX\ RTS} \rightarrow \mathsf{Backoff\ 1}] = 1 - e^{-\Lambda} \tag{5.2}$$

The transition probabilities from TX Data to Idle and from Backoff 1 to TX RTS are 1, since they are the only allowed transitions.

If the node, while in Idle, receives a RTS meant for itself, it transmits the CTS at the beginning of the next time slot (TX CTS state) and goes to RX Data, where it stays until a correct data packet is received. At this point, the node transmits an ACK (TX ACK state) and then goes back to the Idle state. If the node hears an RTS directed to another node, it waits for the packet to be sent (state Wait Data). If no packet transmission is heard, it means that the handshake initiated by the neighbor has not been successfully completed, so that the node can return to the Idle state. Otherwise, the node stays in Wait Data until the data transmission and all required retransmissions have been carried out. These transitions are calculated as:

$$\begin{aligned}P[\mathsf{Idle} \rightarrow \mathsf{TX\ CTS}] &= P[\text{one of the neighbouring nodes generates almost} \\ &\quad \text{one packet addressed to the target node} \mid \text{idle}] \\ &= \binom{N}{1}\frac{1}{N}(1-e^{-\lambda T_s})e^{-\lambda T_s(N-1)} \\ &= (1-e^{-\lambda T_s})e^{-\lambda T_s(N-1)}\end{aligned}$$

$$P[\mathsf{TX\ CTS} \rightarrow \mathsf{RX\ Data}] = 1$$
$$P[\mathsf{RX\ Data} \rightarrow \mathsf{RX\ Data}] = P_{ep}$$
$$P[\mathsf{RX\ Data} \rightarrow \mathsf{TX\ ACK}] = 1 - P_{ep}$$
$$P[\mathsf{TX\ ACK} \rightarrow \mathsf{Idle}] = 1$$

where $P_{ep}$ is the packet error probability. If the node, while in state Idle, hears a CTS addressed to another node, it goes to Wait ACK and waits for the ACK to be sent. If no ACK is received, it means that a packet error has occurred: the node hence stays in this state until the data transmission has been correctly completed. As soon as the node detects the ACK, it goes back to Idle.

$$\begin{aligned} P[\mathsf{Idle} \rightarrow \mathsf{Wait\ ACK}] &= P[\text{one node in the destination's coverage range} \\ &\quad \text{but not in the test node's, generates a packet}|\text{idle}] \\ &= \frac{N_2}{N}(1 - e^{-\lambda T_s})e^{-\lambda T_s (N-1)}, \end{aligned}$$

where $N_2$ is the amount of nodes outside the test node's transmission range but inside the destination node's transmission range:

$$N_2 = \lambda_2(A(r) - \pi R^2).$$

$$P[\mathsf{Wait\ ACK} \rightarrow \mathsf{Wait\ ACK}] = P_{ep}$$
$$P[\mathsf{Wait\ ACK} \rightarrow \mathsf{Idle}] = 1 - P_{ep}$$

It should be noted that the latter event occurs only if the node stays in the destination's transmission range and not in the source's otherwise the source would have heard the RTS too.

If the node, while in state Idle, hears an RTS addressed to another node, it goes to Wait Data and waits for the data to be sent. If no data packet is received, it means that a collision has occurred: the node hence goes back to Idle, otherwise it stays in this state until the data transmission has been correctly completed. To conclude we obtain these transition probabilities too.

$$
\begin{aligned}
P[\text{Idle} \rightarrow \text{Wait Data}] &= P[\text{one of the neighboring nodes generates one packet} \\
&\quad \text{not addressed to the test node}|\text{idle}] \\
&= (N-1)(1-e^{-\lambda T_s})e^{-\lambda T_s (N-1)} \\
P[\text{Wait Data} \rightarrow \text{Wait Data}] &= 1 - P_{coll} \\
P[\text{Wait Data} \rightarrow \text{Idle}] &= P_{coll}
\end{aligned}
$$

where $P_{coll}$ is the probability collision as in (5.2).

## 5.3 Network performance with Slotted FAMA

In this section we present the network performance we obtain through the theory presented in Chapter 2. We provide results for different network parameters e.g. coverage areas, in order to investigate how network performance are affected by such choices.

In particular we calculate the steady-state probabilities for the embedded Markov chain using the stationary equations (2.11) and (2.12). Then, we associate an average duration to each state visit, in order to infer any metric of interest. By assuming that the behavior of nodes is stationary, we can calculate the local network throughput as the node's throughput multiplied by the number of nodes $(N+1)$. Therefore network throughput $(S)$, which is expressed in packets/minute, is obtained by applying the equation (2.70), where in this case $\mu_i$ is the average time spent in each state and $R_i$ is derived by associating one packet to the transition TX Data.In Table 5.1 the average time, which is spent during each state's visit, is shown.

$$S = \frac{\Pi_{\text{TX Data}} P[\text{TX Data} \rightarrow \text{Idle}](\text{N}+1)}{\sum_j \Pi_j T_j}$$

| state | average sojourn |
|---|---|
| Idle | $T_s$ |
| TX RTS | $T_s$ |
| TX Data | $T_s S_{tx}$ |
| Backoff1 | $T_s \frac{(S_{max}+1)}{2}$ |
| TX CTS | $T_s$ |
| RX Data | $T_s S_{tx}$ |
| TX ACK | $T_s$ |
| Wait Data | $T_s$ |
| TX ACK | $T_s$ |
| Wait Data | $T_s + T_s S_{tx}$ |
| WAIT ACK | $T_s S_{tx}$ |

Table 5.1: Average sojourn durations for S-FAMA's states.

The network throughput is represented in Figure 5.3 in logarithmic scale, as a function of the generated traffic at each node. We can notice that throughput increases with the offered traffic until the high packet generation rate causes collisions, so that nodes must occupy the channel for a longer time, since a collision triggers retransmissions. This is why throughput plummets down correspondingly to high traffic load. Moreover we can observe that throughput achieved with smaller coverage areas is greater, due to the shorter propagation delays. In fact, let us consider the same time interval: the amount of packets delivered with shorter propagation delays is greater than the one with longer propagation delays.

Let us consider another important metric: average energy consumption. The mean energy consumption is derived by considering the power consumption asso-

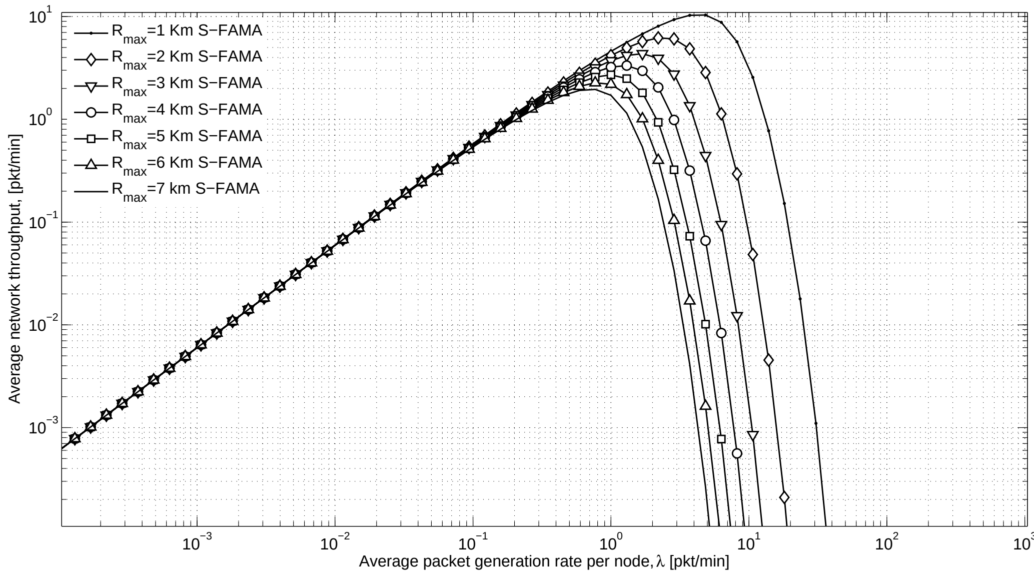


Figure 5.3: Average network throughput for a varying transmission range from 1 to 7 km.

ciated to each sensor[4] operating mode and the corresponding power requirements. Therefore, we did not consider the actual consumed power transmission that we use in the SNR calculus. The reason is that the main aim here was not to derive the actual energy consumption but to infer its trend for a sensor with S-FAMA MAC protocol at a varying generation traffic rate and node's coverage areas.

| operating mode | idle | receiver | transmitter |
|---|---|---|---|
| power [W] | $80 * 10^{-3}$ | 3 | 50 |

Table 5.2: Power levels for the WHOI micro-modem.

Since the node's transceiver is never turned off, the average cost associated to each node's state is normalized respect to the energy consumption during state Idle. The cost of each state is shown and summarized in Table 5.3. The metric we infer is the energy consumption for each correctly delivered packet. Hence it is shown the trend of the amount of energy is necessary to correctly deliver a packet. We chose this metric because if we really want to improve the MAC protocol, we

[4] E.g. see Table 5.2

have to minimize this quantity. We calculate the mean energy consumption by applying the results in 2 and we derive the following equation:

$$E = (N+1)\frac{\sum_j \Pi_j E_j}{\sum_j \Pi_j T_j} / \frac{\Pi_{\text{TX Data}} P[\text{TX Data} \rightarrow \text{Idle}](\text{N}+1)}{\sum_j \Pi_j T_j}. \tag{5.3}$$

| states | normalized energy |
|---|---|
| Idle | $1$ |
| TX RTS | $\frac{50(T_s-\tau_{max})}{80*10^{-3}T_s}$ |
| TX Data | $\S_{tx}\frac{50(S_{tx}T_s-\tau_{max})}{80*10^{-3}T_s}$ |
| Backoff1 | $\frac{(S_{max}+1)}{2}$ |
| TX CTS | $\frac{50(T_s-\tau_{max})}{80*10^{-3}T_s}$ |
| RX Data | $\frac{3(S_{tx}T_s-\tau_m ax)}{80*10^{-3}T_s}$ |
| TX ACK | $\frac{50(T_s-\tau_{max})}{80*10^{-3}T_s}$ |
| Wait Data | $1+\frac{3(S_{tx}T_s-\tau_m ax)}{80*10^{-3}T_s}$ |
| Wait ACK | $1$ |

Table 5.3: Energy consumption associated to each state normalized to the Idle state's cost.

We notice that there is a set of generation traffic rates which correspond to near-maximum throughput and correspondingly near minimum energy consumption for correct packet delivery. These values seem to be the same for increasing coverage ranges, but this is due to the assumption that the packet is always allocated in the same number of time slots, even if the time slot lasts more.

Now we want to calculate the average time which elapses between the transmission and the correctly reception of a single packet. This time interval is called latency and is inferred by applying the first step analysis described in 2.2. In particular we consider the transitions of the embedded Markov chain related to

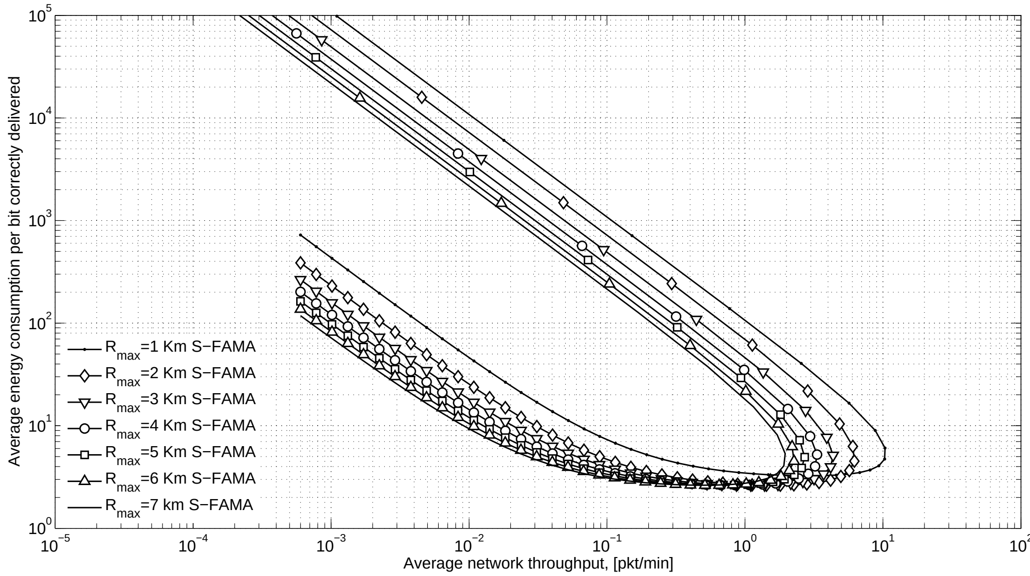


Figure 5.4: Network energy consumption for correctly delivered packet. Coverage radius in the interval $[1, 7]$ km.

transmission, and we derive the average passing time to go from TX RTS to Idle.

$$L = T_{res} + T_{\mathsf{TX\ RTS}\rightarrow\mathsf{Idle}} - (S_{tx}T_s + T_s) \tag{5.4}$$

Where $T_{res}$ is the average residual time of reception due to the mean of propagation delays. We depict the latency as a function of throughput.

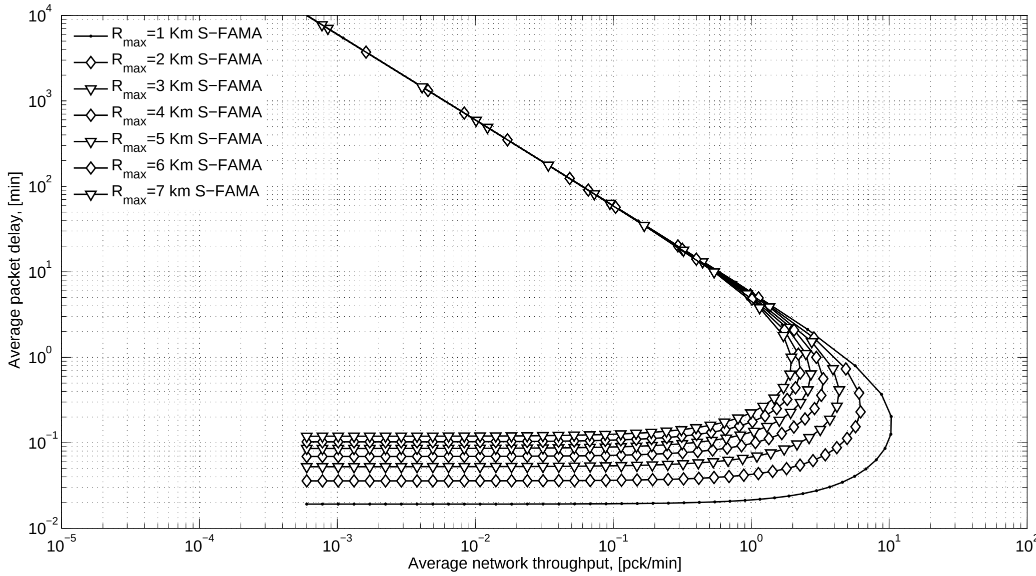


Figure 5.5: Packet delivery delay. Transmission radius in the interval [1, 7] km.

From Figure 5.5 we can infer that latency is longer for greater distance and that for increasing traffic generation rate latency tends to remain constant until the critical offered traffic rate. After this rate, latency increases substantially. The rapid rise of the correct delivery time is due mainly to retransmissions caused by collisions.

# Chapter 6

# An analytical study of T-Lohi

## 6.1 Protocol Description

Tone-Lohi (T-LOHI) [16] is a reservation-based protocol like S-FAMA, featuring a simpler handshake and contention resolution in case of concurrent channel access. With T-LOHI, time is divided in frames; each frame consists of two portions, namely the reservation period and the data period. The reservation period is further partitioned into contention rounds. In order to carry out a fair comparison with S-FAMA, a modified version of T-LOHI is considered here, where all nodes share a global time synchronization. In view of this assumption, each contention round lasts $T_{tone}+\tau_{max}$, where $T_{tone}$ is the duration of the tone transmission. As for S-FAMA's slots, T-LOHI's contention rounds are long enough to accommodate a maximum propagation delay, so that a tone transmitted within a round can potentially reach all nodes within $R_{max}$. Moreover, a tone transmission can take place only at the beginning of a round. Any node that wishes to send a data packet transmits a tone first. If no other tone is received from any neighbor, the node has been successful in reserving the channel, and can start transmitting the data packet. In order to pursue a fair comparison, we assume in the sequel that T-LOHI employs stop-and-wait ARQ with infinite retransmissions like S-FAMA.

In case the node hears other tones during the current contention round, a contention resolution procedure is started, whereby each contender backs off for

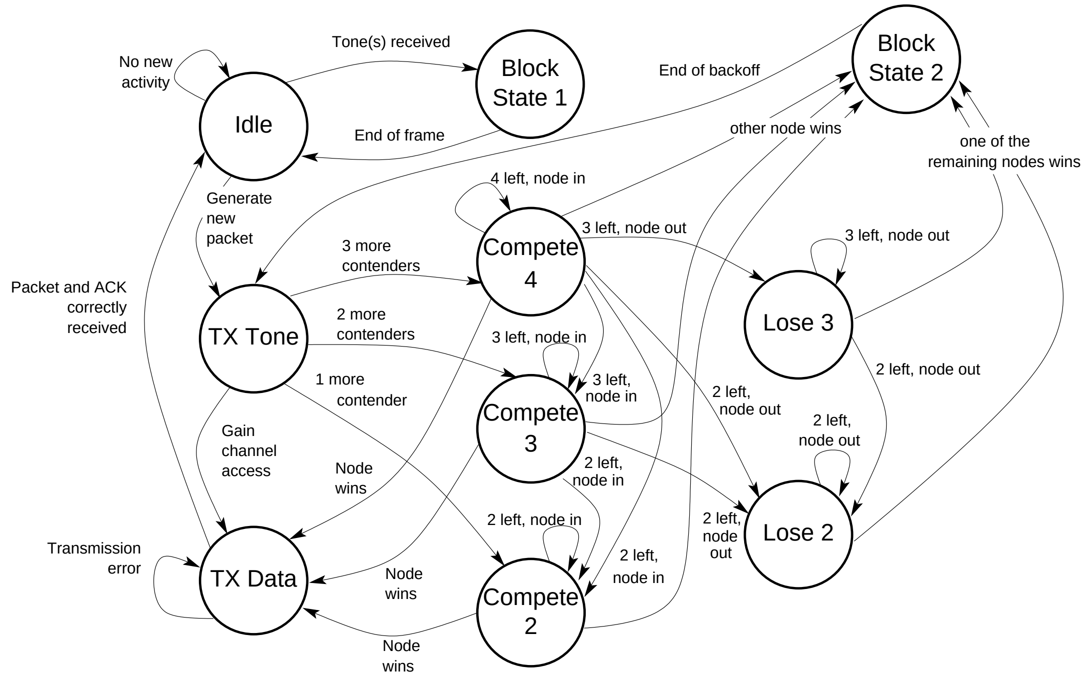


Figure 6.1: State-transition diagram for Tone-Lohi for $N = 3$ neighboring nodes.

a random number of rounds, uniformly chosen in the interval $[0, N]$. The node listens to the channel for the whole duration of this backoff. During this phase, if a node is the only one to choose the earliest round to transmit another tone among all contenders, it is called the *winner*, and is allowed to start data transmissions immediately. If more than one nodes choose the same (and the earliest) contention round simultaneously, they are called *competitors* and continue to contend for channel access, by repeating the above procedure. If a node hears one or more tones before attempting to access the channel again (i.e., one or more competitors chose a shorter back-off time), it is called a *loser* and exits the contention phase. All losers start listening to channel activity until both the contention and the data transmission phases end, after which they go back to idle mode.

### 6.1.1 Model for T-LOHI

The Markov chain that models T-LOHI is depicted in Figure 6.1. Starting from state Idle, if the node hears any tone, it goes to Block state 1, and keeps listening to the channel until the end of the current frame, after which it returns to Idle. Whenever the node generates one or more packets, it moves to state TX Tone and sends a tone. If it is the only one that tries to access the channel it goes to TX Data state, during which the node forwards the data packet and retransmits until it is correctly received. Otherwise it goes to one of the Compete k states, $2 \leq k \leq N$, where $N$ is the average number of neighbors in coverage. From any of the Compete k states, the node observes the following behavior:

- if the node chooses the earliest contention round along with other $n$ competitors, it moves to state Compete n, $2 \leq n \leq k$, and keeps contending after a back-off interval as discussed in section 6.1;

- if the node hears one or more tones, it moves to state Lose n, $2 \leq n \leq k-1$, where $n$ is the number of other nodes that are currently competing;

- if the node chooses the earliest contention round, and is the first and only to transmit a further tone, it gains channel access, and thus goes to state TX Data;

- if one of the other competitors gains channel access, the node goes to Block state 2.

From Lose n the node tracks the contention by recording the number of nodes that are still competing for the channel. Correspondingly,

- the node moves to state Lose p, $2 \leq p \leq n$, if $p$ competitors choose the earliest contention round altogether, and $n - p$ exit the contention, having chosen a later round (in other words, a longer back-off time);

- if one of the current competitors gains channel access, the node moves to Block state 2.

During Block state 2, the node receives the data packet being sent by the contention winner, in case it is its intended destination; otherwise, it discards the packet. After that, it goes back to state TX Tone to perform another transmission attempt (recall that the node entered Block state 2 after having lost the preceding contention).

## 6.2 The Transition Probabilities

In this section a deeper insight into the probabilities of the transition described in 6.1.1, is provided.

As seen, starting from the Idle state the target node can achieve Block State 1, TX Tone and Idle itself, so given that $p$ is the probability of generating packets according to Poisson Process, it results:

$$\begin{aligned}
P[\mathsf{Idle} \rightarrow \mathsf{Block\ State\ 1}] &= P[\text{at least one neighboring node transmits a tone}] \\
&= \sum_{k=1}^{N} \binom{N}{k} p^k (1-p)^{N-k} \\
P[\mathsf{Idle} \rightarrow \mathsf{TX\ Tone}] &= P[\text{the node transmits a tone}] \\
&= p \\
P[\mathsf{Idle} \rightarrow \mathsf{Idle}] &= P[\text{no tones are transmitted by either the node or its neighbors}] \\
&= 1 - \sum_{k=1}^{N} \binom{N}{k} p^k (1-p)^{N-k} - p
\end{aligned}$$

From the state Block State 1 the node goes to Idle after the frame has been completed, therefore:

$$P[\mathsf{Block\ State\ 1} \rightarrow \mathsf{Idle}] = 1.$$

When the target node transmits the tone, the probabilities of going to states TX Data, Compete n for $2 \leq n \leq N$ are in the following explained.

$$\begin{aligned} P[\mathsf{TX\ Tone} \rightarrow \mathsf{TX\ Data}] &= P[\text{the target node is the only node} \\ &\quad \text{that has transmitted the tone}] \\ &= (1-p)^N \\ P[\mathsf{TX\ Tone} \rightarrow \mathsf{Compete\ n}] &= P[\text{other n nodes has sent a tone}] \\ &= \binom{N}{n} p^n (1-p)^{N-n} \end{aligned}$$

As discussed in 6.1.1, the state Compete n represents the competition among $n$ nodes. Hence from this state, the node can move to any state Compete k and Lose k, for $2 \le k \le n-1$, Block State 1 and TX Data. In the following the process of deducing such probabilities will be provided, starting from $P[\mathsf{Compete\ n} \rightarrow \mathsf{Compete\ k}]$. Given that the node has uniformly chosen a certain contention round $m$[1] of N+1, we find the probability that more $k$, with $k < n$, competitors chose this contention round?

$$P[\mathsf{Compete\ n} \rightarrow \mathsf{Compete\ k}|m] = \begin{cases} \binom{j-1}{k-1}\left(\frac{1}{N+1}\right)^{k-1}\left(\frac{N+1-m}{N+1}\right)^{j-k} & \text{if } 1 \le m < N+1 \\ 0 & \text{if } m = N+1 \end{cases}$$

Applying the Total Probability Theorem it results:

$$P[\mathsf{Compete\ n} \rightarrow \mathsf{Compete\ k}] = \sum_{m=1}^{N} \binom{n-1}{k-1}\left(\frac{1}{N+1}\right)^{k-1}\left(\frac{N+1-m}{N+1}\right)^{n-k}\frac{1}{N+1}$$

The $P[\mathsf{Compete\ n} \rightarrow \mathsf{Lose\ k}]$ is deduced by conditioning to the contention round chosen by the remaining competitors and the contention round chosen by the target node, labeled ad $l$.

$$P[\mathsf{Compete\ n} \rightarrow \mathsf{Lose\ k}|m,l] = \begin{cases} \left(\frac{N+1-m}{N+1}\right)^{n-k-1} & \text{if } l > m \\ 0 & \text{if } l \le m \end{cases}$$

[1]this contention round is also the earliest chosen by competitors

By removing the conditioning on the contention round chosen by the target node, it results:

$$P[\mathsf{Compete\ n} \rightarrow \mathsf{Lose\ k}|m] = \sum_{l=m+1}^{N+1} \frac{1}{N+1}\Big(\frac{N+1-m}{N+1}\Big)^{n-k-1}$$

By removing the condition on contention round chosen by the remaining competitors we get:

$$P[\mathsf{Compete\ n} \rightarrow \mathsf{Lose\ k}] = \sum_{m=1}^{N} \binom{n-1}{k}\Big(\frac{1}{N+1}\Big)^{k}\Big(\frac{N+1-m}{N+1}\Big)^{n-k}$$

The probability of remaining in state Compete n, for $2 \leq n \leq N$ is:

$$\begin{aligned} P[\mathsf{Compete\ n} \rightarrow \mathsf{Compete\ n}] &= P[\text{all the competitors chose the same contention round}] \\ &= \Big(\frac{1}{N+1}\Big)^{n-1} \end{aligned}$$

As discussed, the node goes in Block State 2, if one of the $n-1$ other competitors picks the earliest contention round, so the probability of this event is:

$$P[\mathsf{Compete\ n} \rightarrow \mathsf{Block\ State\ 2}] = \sum_{m=1}^{N} \Big(\frac{N+1-m}{N+1}\Big)^{n-1} \frac{1}{N+1}\binom{n-1}{1}$$

At the end it is found out the probability of transmitting the data packet, therefore the probability that the target node is the only to choose the earliest contention round.

$$P[\mathsf{Compete\ n} \rightarrow \mathsf{TX\ Data}] = \sum_{m=1}^{N} \Big(\frac{N+1-m}{N+1}\Big)^{n-1} \frac{1}{N+1}$$

In Appendix, it is shown that all these probabilities sum up to one, as it is required by the Markov property. Moreover the others probabilities are inferred through similar reasoning. They are collected in the following for the sake of clarity.

$$P[\text{Lose n} \rightarrow \text{Lose n}] = \left(\frac{1}{N+1}\right)^n$$

$$P[\text{Lose n} \rightarrow \text{Lose j}] = \left(\frac{1}{N+1}\right)^j \binom{n}{j} \sum_{m=1}^{N} \left(\frac{N+1-m}{N+1}\right)^{n-j}$$

$$P[\text{Lose n} \rightarrow \text{BlockState 2}] = \left(\frac{1}{N+1}\right) n \sum_{m=1}^{N} \left(\frac{N+1-m}{N+1}\right)^{n-1}$$

From state Block State 2 the node can go only to state TX Tone, therefore this transition has probability one. After the transmission the node waits for the acknowledgement, so the transition is from TX Data to itself and the probability associated to this transition depends on the error probability due to noise in the channel, and on the residual collision probability.

## 6.3 Network performance for Tone Lohi

In this section we provide an analysis of the average network performance such as throughput, energy consumption and latency, already derived for S-FAMA MAC protocol:.

We estimate throughput by applying the results described in We infer the mean time which each state lasts in order to calculate the average network throughput by means of 2.70. The visit times are summarized in Table 6.1, where we indicate with $T_{bs1}$ the mean time elapses in Block State 1.

We estimate $T_{bs1}$ as the sum of the packet transmission time and the average duration of the contention. In order to calculate the latter we consider $N$ i.i.d stochastic variables $\{n_1, n_2, \ldots, n_N\}$, whose probability distribution is uniform one, and we compute the mean of the following stochastic variable:

$$v = \min\{n_1, n_2, \cdot, n_N\}$$

So we have to find $E[v]$, where $v$ is a positive stochastic variable. We know that:

$$E[x] = \sum_{t=0}^{\infty}(1 - F_x(t)),$$

and since we have for $0 \leq t \leq (N+1)T_s$

$$1 - F_v(t) = P[v > t] = P[n_1 > t, n_2 > t, \ldots, n_N > t],$$

but

$$P[n_1 > t, n_2 > t, \ldots, n_N > t] = (1 - \frac{1}{t})^N$$

we get:

$$E[v] = \sum_{t=0}^{(N+1)T_s} (1 - \frac{1}{t})^N.$$

This is the mean of $v$, given the number of nodes that generated a packet. We weigh all these events with the corresponding probability and finally we obtain $T_{bs}$.

We call $T_{bs2}$ the mean visit time in Block State 2; this time corresponds to the average time to correctly deliver a packet, here retransmissions are not caused by collisions but only by the channel attenuation and noise.

| state | visit time |
|---|---|
| Idle | $T_s$ |
| Block State 1 | $T_{bs1}$ |
| TX Tone | $T_s$ |
| Compete n | $T_s$ |
| Lose n | $T_s$ |
| TX Data | $T_sS_{tx}$ |
| Block State 2 | $T_{bs2}$ |

Table 6.1: This table shows the average visit time at each state.

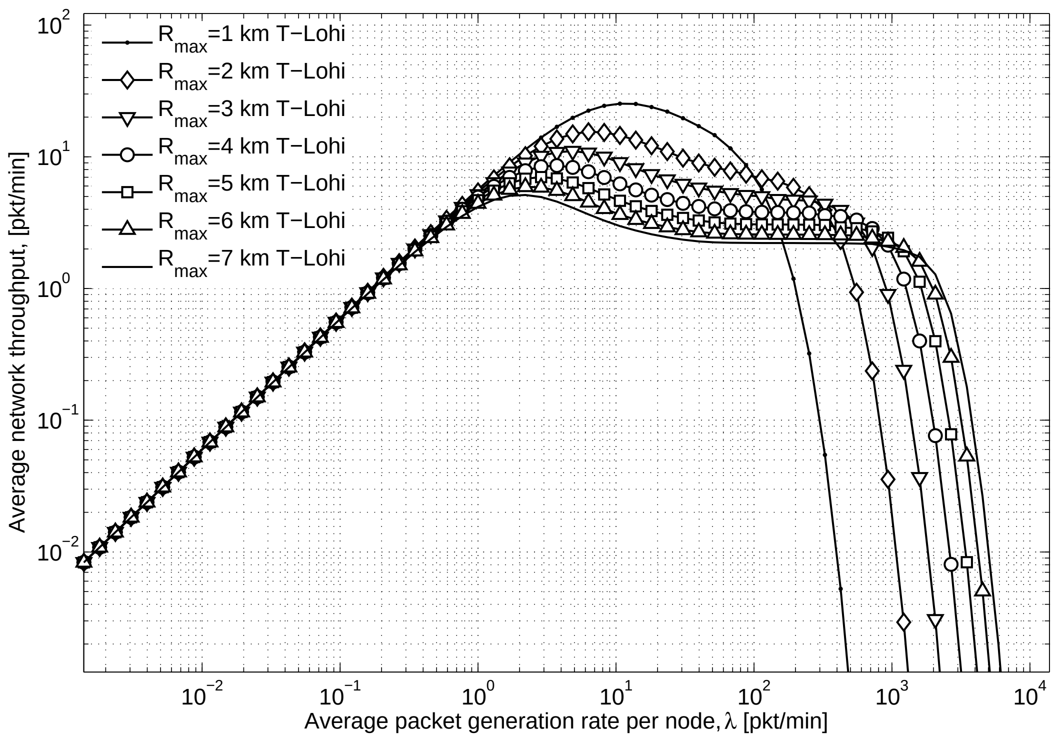


Figure 6.2: Average network throughput. Coverage radius in interval $[1, 7]$ km.

In Figure 6.2 we show the trend of the average network throughput as a function of the traffic generation rate, we observe that the amount of correctly arrived packets increases with the offered traffic until a critical value, after that the function assumes a roughly constant value and it plummets down at very high traffic load. The longer the maximum propagation delay is, the higher the supported traffic load.

As for S-FAMA, the average energy consumption is inferred by considering power used in some operating modes of a particular kind of micro-modem. In Table 6.2 are shown the energetic costs of each state, normalized to energy consumption in Idle, since the transceiver is never turned off.

Energy consumption necessary to correctly deliver a packet decreases at the increasing of throughput and traffic load, until a critical value, which represents the optimal traffic load for which the energy consumption has a minimum.

Eventually latency is derived by considering the mean time necessary to pass

| states | normalized energy |
|---|---|
| Idle | $1$ |
| Block State 1 | $\frac{3S_{tx}}{80*10^{-3}}$ |
| TX Tone | $\frac{50T_{tone}}{80*10^{-3}T_s}$ |
| TX Data | $S_{tx}\frac{50}{80*10^{-3}}$ |
| Compete n | $\frac{50T_{tone}}{80*10^{-3}T_s}+\frac{3S_{tx}}{80*10^{-3}}$ |
| Lose n | $1$ |
| Block State 2 | $1$ |

Table 6.2: Energy consumption associated to each state normalized to Idle's cost.

from TXTone to Idle.

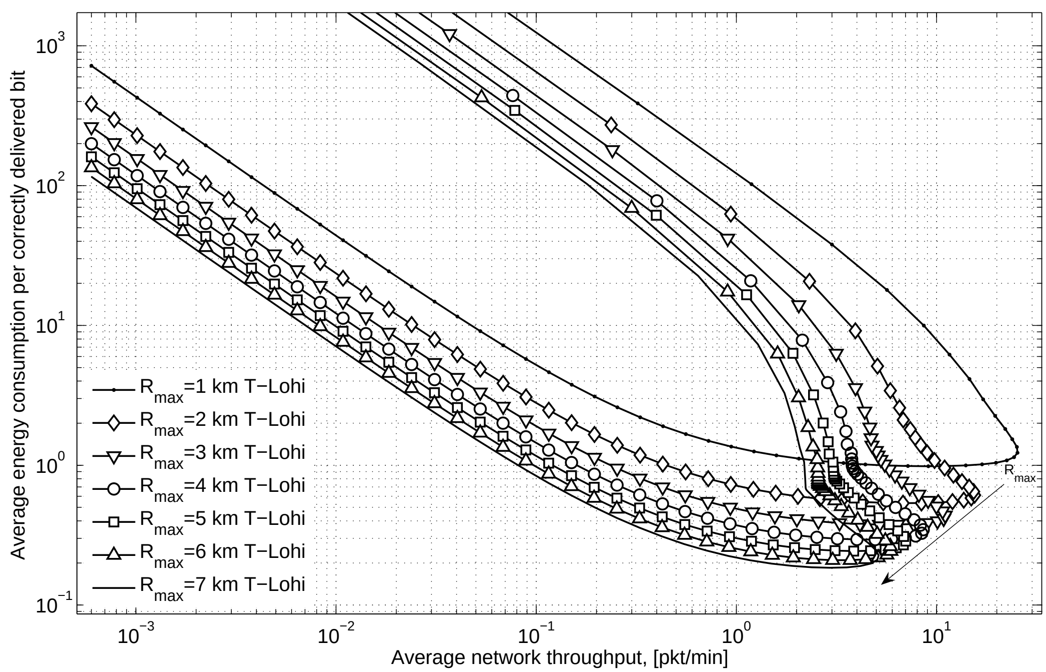


Figure 6.3: Network energy consumption for correctly delivered packet. Transmission radius $[1, 7]$ km.

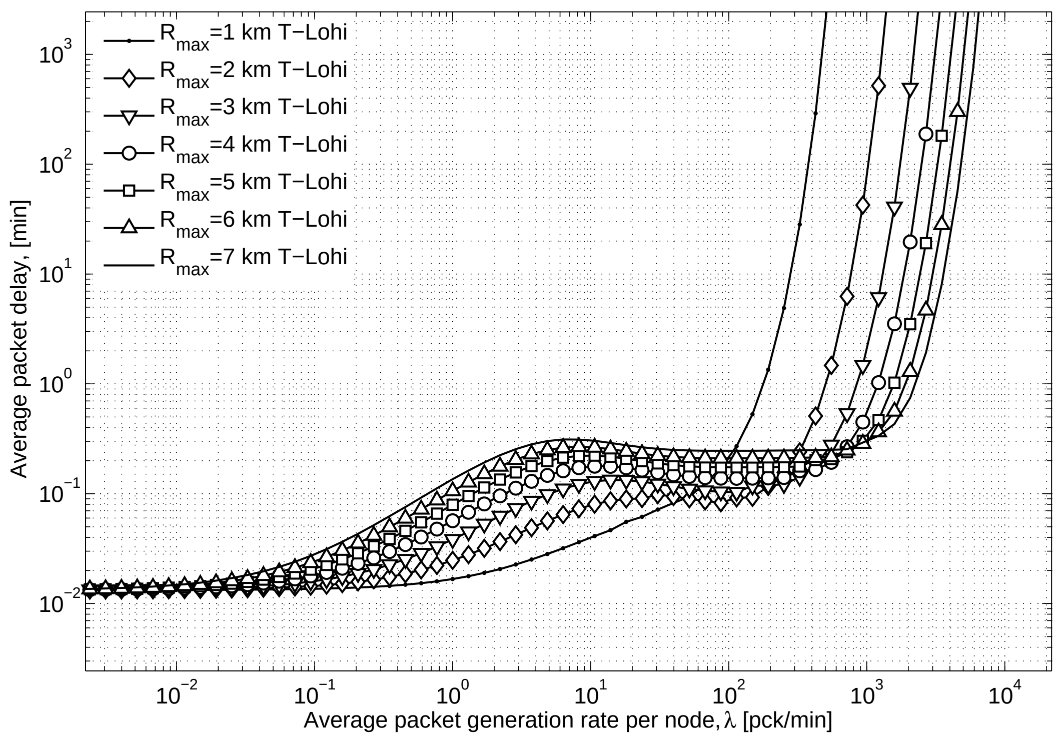


Figure 6.4: Delivery delay with coverage radius $[1, 7]$ .

# Chapter 7

# Conclusions

## 7.1 A qualitative comparison between S-FAMA e Tone-Lohi

To conclude we compare the two presented protocols for some representative coverage radius: we chose a coverage of 1 km, 3 km and 7 km. We represent the aggregate throughput achieved by S-FAMA and T-LOHI as a function of the traffic generation rate in Figure 7.2. Given that both protocols are slotted and that the duration of a slot is mainly determined by the propagation delay, both S-FAMA and T-LOHI offer the same throughput performance at low traffic. As traffic increases, however, the slope of the throughput curves progressively decreases until the a maximum throughput value is reached. After that, the curves tend to plummet down with increasing traffic due to network congestion. T-LOHI reaches a higher throughput, on average, for all considered values of $R_{max}$, ad its lighter handshake procedure allows throughput to remain more stable for higher offered traffic values, before congestion causes it to drop. However, the only drawback of the absence of an explicit CTS in T-LOHI is that no check is carried out to see if the intended receiver node is in fact available to receive the transmission (it may be involved in the reception of other packets), so that the whole contention phase for channel access could end up with a winner, but no node available to receive. Furthermore, the absence of a CTS leaves the receiver unprotected from hidden terminals: therefore, any tone sent in the proximity of a receiver could potentially disturb the reception.

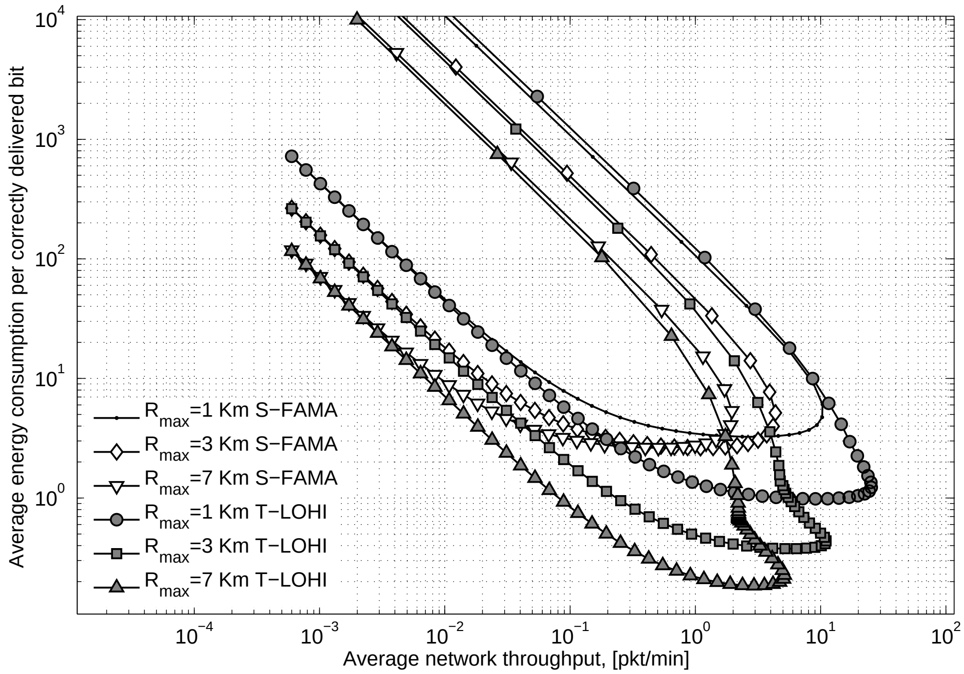


Figure 7.1: Network energy consumption for correctly delivered packet as a function of throughput.

Figure 7.3 shows a comparison of S-FAMA's and T-LOHI's transmission delay performance as a function of traffic, and it is in line with the above discussion. Interestingly, T-LOHI's lighter channel access technique allows for a shorter access delay, on average. From this figure, we also observe that T-LOHI's delay remains quite stable until collisions increase: at this point, the effect of repeated retransmissions on delay and the high probability that a node loses contentions and backs off tend to increase delay considerably. However, it should be noted that this happens at a higher traffic generation rate than for S-FAMA.

Some further information can be inferred from the results. Figure 7.2 suggests that congestion is very critical for S-FAMA, as the throughput tends to decrease very steeply at high traffic. This indicates that S-FAMA would be best used in conjunction with some congestion control mechanism, assuming that the packet generation rate can be controlled by the nodes. Conversely, T-LOHI exhibits a smoother trend around the maximum throughput value, hence a better resilience to temporary or local congestion situations.

The importance of traffic control is further confirmed by looking at Figure 7.1, which depicts the average energy consumption per correctly delivered bit against throughput. Note that, in this case, curves are spanned in a counter-clockwise direction by increasing traffic. Moreover, recall that as traffic increases, throughput first increases up to a maximum point, and then decreases, which explain why the curves fold back. Besides confirming that energy tends to increase very rapidly as the network becomes congested, Figure 7.1 shows that there exists an "optimum" value of the traffic generation rate for which the best energy and throughput performance is jointly achieved: more specifically, this traffic value corresponds to the point on the curves that is closer to the bottom-right corner of the graph. Such a point exists for both S-FAMA and T-LOHI curves, and depends on network parameters such as $R_{max}$ and the average number of neighbors $N$. Finally, Figure 7.1 suggests that T-LOHI's performance is more sensitive to the choice of a suboptimal traffic generation rate (the curves show a steeper change in the throughput-energy tradeoff for varying traffic.

## 7.2 Conclusions and future directions

We have developed two semi-Markov models that capture the evolution of a node's behavior according to either S-FAMA or T-LOHI, and have employed them to extract relevant network metrics such as throughput, energy consumption and packet delay. Our results suggest that while the four-way contention of Slotted FAMA yields a greater degree of protection against hidden terminal effects, it also considerably lowers throughput, especially in the presence of long coverage ranges, thus long maximum propagation delays.

We also observed that Tone-Lohi provides better resilience against congestion

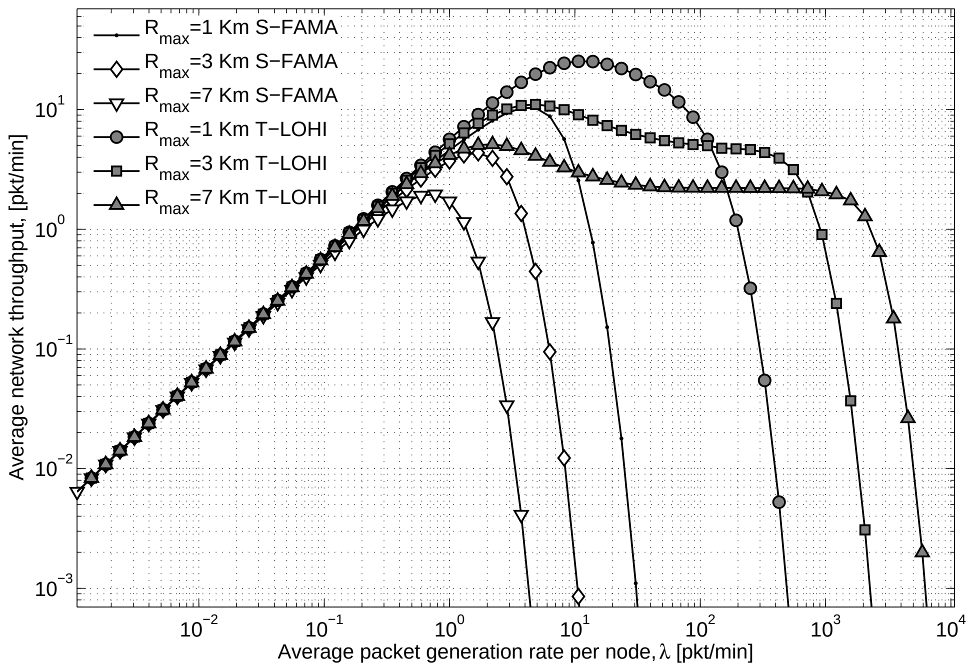


Figure 7.2: Average network throughput in S-FAMA and T-Lohi.

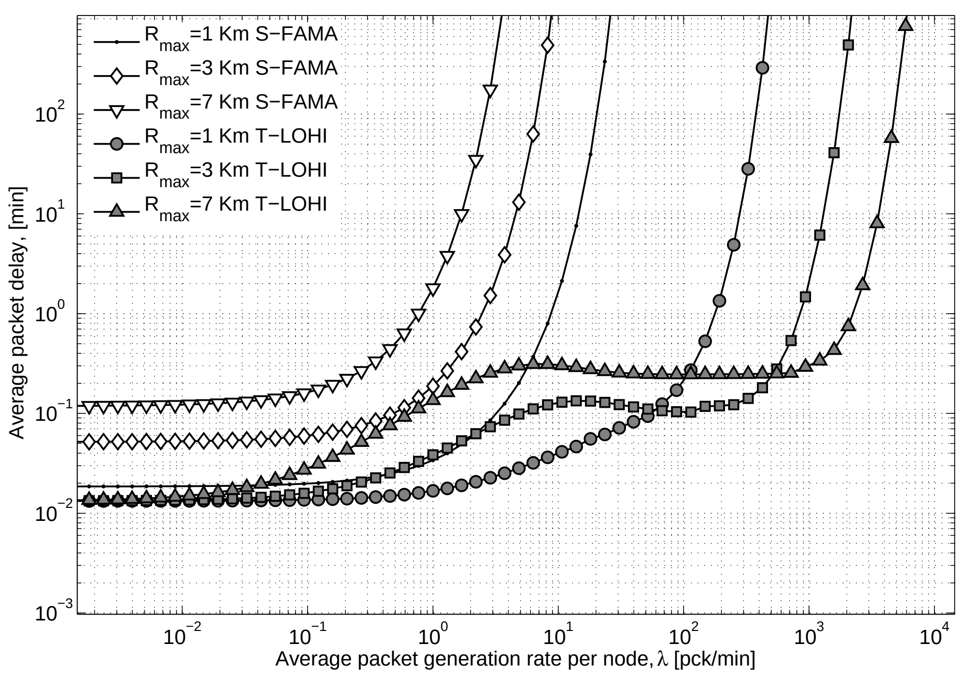


Figure 7.3: Average delivery delay in S-FAMA protocol compared with average delivery delay in Tone-Lohi.

and more convenient working points on energy-throughput tradeoff curves. These results allow to understand which protocol is more suited to a given network setting, and are expected to be of help in designing novel protocols that possibly outperform currently available solutions.

Future directions of this work include, in fact, the extension of this study through the application of the same analysis methodology to other MAC protocols, the comparison of analytical results with simulations, and the design of a novel protocol based on the insight provided by our analysis.

# Appendix A

In the following we prove that the probabilities computed in section 6 sum to 1.

$$
\begin{aligned}
\sum_{k=2}^{n-1}\sum_{m=1}^{N}\binom{n-1}{k-1}\Big(\frac{1}{N+1}\Big)^{n}(N+1-m)^{n-k}+ \\
+\sum_{k=2}^{n-1}\sum_{m=1}^{N}\binom{n-1}{k}\Big(\frac{1}{N+1}\Big)^{n}(N+1-m)^{n-k}+ \\
+\frac{N+1}{(N+1)^{n}}+ \\
+(n-1)\sum_{m=1}^{N}(N+1-m)^{n-1}\frac{1}{(N+1)^{n}} \\
+\sum_{m=1}^{N}(N+1-m)^{n-1}\frac{1}{(N+1)^{n}}
\end{aligned}
\quad (A.1)
$$

We define the following function for $N \geq 1$ and $n \geq 0$[1]:

$$S_n(N)=\sum_{C=1}^{N+1}C^{n} \quad \text{for } 0 \leq C \leq N, \quad (A.2)$$

and we can write $\sum_{m=1}^{N}(N+1-m)^{(n-k)}$ in terms of this function, hence:

$$\sum_{m=1}^{N}(N+1-m)^{(n-k)}=\sum_{C=1}^{N}C^{(n-k)}=S_{n-k}(N-1) \quad (A.3)$$

[1]Actually in the model we have $N \geq 1$ and $n \geq 2$

Starting from this function, and using the Newton's formula[2], we obtain the following equations for $N \geq 1$:

$$\begin{aligned}
S_n(N) &= \sum_{C=1}^{N+1} C^n \\
&= 1 + \sum_{C=2}^{N+1} (C-1+1)^n \\
&= 1 + \sum_{C=2}^{N+1} \sum_{l=0}^{n} \binom{n}{l} (C-1)^l \\
&= 1 + \sum_{l=0}^{n} \binom{n}{l} \sum_{C=2}^{N+1} (C-1)^l \\
&= 1 + \sum_{l=0}^{n} \binom{n}{l} S_l(N-1).
\end{aligned}$$

Moreover:

$$\begin{aligned}
S_n(N) &= \sum_{C=1}^{N+1} C^n \\
&= \sum_{C=1}^{N} C^n + (N+1)^n \\
&= S_n(N-1) + (N+1)^n.
\end{aligned}$$

By combining these two results we get:

$$1 + \sum_{l=0}^{n-1} \binom{n}{l} S_l(N-1) + S_n(N-1) = S_n(N-1) + (N+1)^n, \tag{A.4}$$

which yields:

$$\sum_{l=0}^{n-1} \binom{n}{l} S_l(N-1) = (N+1)^n - 1 \tag{A.5}$$

[2]Newton's formula states that $(a+b)^n = \sum_{j=0}^{n} \binom{n}{j} a^j b^{n-j}$

Now using the equation (A.3), we rewrite the initial sum as:

$$\Big(\frac{1}{N+1}\Big)^n\Big[\sum_{k=2}^{n-1}\binom{n-1}{k-1}S_{n-k}(N-1)+ \\ +\sum_{k=2}^{n-1}\binom{n-1}{k}S_{n-k}(N-1)+ \\ +N+1+ \\ +(n-1)S_{n-1}(N-1) \\ +S_{n-1}(N-1)\Big] \quad (A.6)$$

Hence we want to prove that the following expression is equal to one.

$$\frac{1}{(N+1)^n}\Big[\sum_{k=2}^{n-1}\binom{n-1}{k-1}S_{n-k}(N-1)+ \\ +\sum_{k=2}^{n-1}\binom{n-1}{k}S_{n-k}(N-1)+N+1+nS_{n-1}(N-1)\Big] \quad (A.7)$$

Given the well known identity

$$\binom{n-1}{k-1}=\binom{n}{k}-\binom{n-1}{k}, \quad (A.8)$$

we can write

$$\sum_{k=2}^{n-1}\binom{n-1}{k-1}S_{n-k}(N-1)=\sum_{k=2}^{n-1}\binom{n}{k}S_{n-k}(N-1)-\sum_{k=2}^{n-1}\binom{n-1}{k}S_{n-k}(N-1). \quad (A.9)$$

Substituting (A.9) in (A.6), we get the following expression:

$$\Big(\frac{1}{N+1}\Big)^n\Big[\sum_{k=2}^{n-1}\binom{n}{k}S_{n-k}(N-1)+N+1+nS_{n-1}(N-1)\Big]. \quad (A.10)$$

By posing $n-k=k'$, we get

$$\Big(\frac{1}{N+1}\Big)^n\Big[\sum_{k'=1}^{n-2}\binom{n}{k'}S_{k'}(N-1)+N+1+nS_{n-1}(N-1)\Big]. \quad (A.11)$$

Hence, the term $\sum_{k=1}^{n-2}\binom{n}{k'}S_{k'}(N-1)$ can be rewritten, thanks to the results inferred above:

$$\sum_{k=1}^{n-2}\binom{n}{k}S_k(N-1)=\sum_{k=0}^{n-2}\binom{n}{k}S_k(N-1)-S_0(N-1). \quad (A.12)$$

But since

$$\sum_{k=0}^{n-2} \binom{n}{k} S_k(N-1) = \sum_{k=0}^{n-1} \binom{n}{k} S_k(N-1) - nS_{n-1}(N-1) \tag{A.13}$$

we have

$$\sum_{k=1}^{n-2} \binom{n}{k} S_k(N-1) = \sum_{k=0}^{n-2} \binom{n}{k} S_k(N-1) - S_0(N-1) - nS_{n-1}(N-1) \tag{A.14}$$

Therefore we can finally write

$$\left(\frac{1}{N+1}\right)^n \Big[(N+1)^n - 1 - nS_n(N-1) - N + N + 1 + nS_{n-1}(N-1)\Big] = 1 \tag{A.15}$$

## A.1 Proof of Proposition 2.1.1

$$\begin{aligned} & P[X_0 = i_0, X_1 = i_1, \ldots, X_n = i_n] = \\ & = P[X_n = i_n | X_0 = i_0, X_1 = i_1, \ldots, X_{n-1} = i_{n-1}] \cdot \\ & \quad \cdot P[X_0 = i_0, X_1 = i_1, \ldots, X_{n-1} = i_{n-1}] \end{aligned} \tag{A.16}$$

Now, using the Markov property, the probability of being in state $i_n$ conditioned on the states at time instants $n-1, n-2, \ldots, 0$ is independent on time instants $n-2, \ldots, 0$, therefore one gets:

$$\begin{aligned} & P[X_0 = i_0, X_1 = i_1, \ldots, X_n = i_n] = \\ & = P[X_n = i_n | X_{n-1} = i_{n-1}] P[X_0 = i_0, X_1 = i_1, \ldots, X_{n-1} = i_{n-1}] \end{aligned}$$

Using the same procedure with $P[X_0 = i_0, X_1 = i_1, \ldots, X_{n-1} = i_{n-1}]$ we have,

$$\begin{aligned} & P[X_0 = i_0, X_1 = i_1, \ldots, X_n = i_n] = \\ & = P[X_n = i_n | X_{n-1} = i_{n-1}] P[X_{n-1} = i_{n-1} | X_{n-2} = i_{n-2}] \cdot \\ & \cdot P[X_0 = i_0, X_1 = i_1, \ldots, X_{n-2} = i_{n-2}] \end{aligned}$$

Then, upon repeating the procedure, (A.17) becomes:

$$P[X_0 = i_0, X_1 = i_1, \ldots, X_n = i_n] = \prod_{k=1}^{n} P[X_k = i_k | X_{k-1} = i_{k-1}] \cdot P[X_0 = i_0]$$

Finally, assuming that the Markov chain has stationary transition probabilities:

$$P[X_0 = i_0, X_1 = i_1, \ldots, X_n = i_n] = \prod_{k=1}^{n} P_{i_{k-1},i_k} \cdot P[X_0 = i_0]$$

## A.2 Proof of theorem 2.1.1

$$\begin{aligned} & P[X_0 = i_0, X_1 = i_1, \ldots, X_n = i_n] = \\ & = P[X_n = i_n | X_0 = i_0, X_1 = i_1, \ldots, X_{n-1} = i_{n-1}] \cdot \\ & \quad \cdot P[X_0 = i_0, X_1 = i_1, \ldots, X_{n-1} = i_{n-1}] \end{aligned} \tag{A.17}$$

Now, using the Markov property, the probability of being in state $i_n$ conditioned on the states at time instants $n-1, n-2, \ldots, 0$ is independent on time instants $n-2, \ldots, 0$, therefore one gets:

$$\begin{aligned} & P[X_0 = i_0, X_1 = i_1, \ldots, X_n = i_n] = \\ & = P[X_n = i_n | X_{n-1} = i_{n-1}] P[X_0 = i_0, X_1 = i_1, \ldots, X_{n-1} = i_{n-1}] \end{aligned}$$

Using the same procedure with $P[X_0 = i_0, X_1 = i_1, \ldots, X_{n-1} = i_{n-1}]$ we have,

$$\begin{aligned} & P[X_0 = i_0, X_1 = i_1, \ldots, X_n = i_n] = \\ & = P[X_n = i_n | X_{n-1} = i_{n-1}] P[X_{n-1} = i_{n-1} | X_{n-2} = i_{n-2}] \cdot \\ & \cdot P[X_0 = i_0, X_1 = i_1, \ldots, X_{n-2} = i_{n-2}] \end{aligned}$$

Then, upon repeating the procedure, (A.17) becomes:

$$P[X_0 = i_0, X_1 = i_1, \ldots, X_n = i_n] = \prod_{k=1}^{n} P[X_k = i_k | X_{k-1} = i_{k-1}] \cdot P[X_0 = i_0]$$

Finally, assuming that the Markov chain has stationary transition probabilities:

$$P[X_0 = i_0, X_1 = i_1, \ldots, X_n = i_n] = \prod_{k=1}^{n} P_{i_{k-1},i_k} \cdot P[X_0 = i_0]$$

$$P_{ij}^{(n)} = P[X_{m+n} = j | X_m = i] \tag{A.18}$$

For the low of total probability, it is possible to rewrite (A.18) as:

$$P_{ij}^{(n)} = \sum_{n=0}^{+\infty} P[X_{m+n} = j, X_{m+1} = n | X_m = i] =$$
$$= \sum_{k=0}^{+\infty} P[X_{m+n} = j | X_{m+1} = k, X_m = i] P[X_{m+1} = k, X_m = i] =$$
$$= \sum_{k=0}^{+\infty} P_{ik} P_{kj}^{(n-1)}$$

# Bibliography


[1] J.G. Proakis and E.M. Sozer, A. Rice, Milica Stojanovic , **Shallow Water Acoustic Networks** , *IEEE Communication Magazine*, November 2001.

[2] I.F. Alkyldiz, D. Pompili, T. Melodia, **State-of-the-Art in Protocol Research for Underwater Acoustic Sensor Networks** *WUWNet 2006.*

[3] Milica Stojanovic **On the Relationship Between Capacity and Distance in an Underwater Acoustic Communication Channel** *WUWNet 2006.*

[4] James Preisig **Acoustic Propagation Considerations for Underwater Acoustic Communication Network Development** *WUWNet 2006.*

[5] Albert F. Harris III, Michele Zorzi **Modeling the Underwater Acoustic Channel in ns2** *ACM International Conference Prceeding Series.* Vol.321

[6] M. Molins and M. Stojanovic **Slotted FAMA: a MAC Protocol for underwater acoustic networks** *in Proc. IEEE Oceans, Singapore, Sept. 2006*

[7] P. Casari, B. Tomasi, M. Zorzi **A Comparison Between the Tone-Lohi and Slotted FAMA MAC Protocols for Underwater Networks** *in Proc. IEEE/OES OCEANS, Quebec, Sept. 2008*

[8] H. Tan, W. Seah **Distributed CDMA-based MAC Protocol for Underwater Sensor Network** *IEEE Conference on Local Computer Networks*

[9] H.Doukkali, L. Nuaymi, S. Houcke **Distributed MAC Protocols for Underwater Acoustic Data Networks**

[10] P. Xie, J.H. Cui **Exploring Random Access and Handshaking Techniques in LArge-Scale Underwater Wireless Acoustic Sensor Networks**

[11] K.B. Kredo II, P. Mohapatra **A Hybrid Medium Access Control Protocol for Underwater Wireless Networks**

[12] M.K. Park, V. Rodoplu **UWAN-MAC: An Energy-Efficient MAC Protocol for Underwater Acoustic Wireless Sensor Networks** *IEEE Journal of Oeceanic Engineering* Vol. 32 NO.3 July 2007

[13] P.Casari, F.E. Lapiccirella, M. Zorzi **A Detailed Simulation Study of the UWAN-MAC Protocol for Underwater Acoustic Networks**

[14] X. Guo, M. Frater and M. Ryan **A Propagation-delay-tolerant Collision Avoidance Protocol for Underwater Acoustic Sensor Networs** *Proc. OCEANS'06 Asia Pacific IEEE - Singapore Conference/Exhibition,* 2006

[15] B.Peleato and M. Stojanovic. **A MAC Protocol for Ad Hoc Underwater Acoustic Sensor Networks** *Proc. 1st ACM International Workshop on UnderWater Networks,*pp. 113-115, 2006

[16] A. A. Syed, W. Ye and J. Heidemann. **Medium Access for Underwater Acoustic Sensor Networks** *USC/ISI Technical Report ISI-TR-624*, October 2006

[17] A. A. Syed, W. Ye and J. Heidemann. **Understanding Spatio-Temporal Uncertainty in Medium Access with ALOHA Protocols**

[18] C. Petrioli, R. Petroccia and Milica Stojanovic. **A Comparative Performance Evaluation of MAC Protocols for Underwater Sensor Networks**

[19] P. Casari, S. Marella and M. Zorzi. **A Comparison of Multiple Access Techniques in Clustered Underwater Acoustic Networks**

[20] 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

[21] P. Casari and M. Rossi and M. Zorzi. **Fountain codes and their application to broadcasting in underwater networks: performance modeling and relevant tradeoffs** *in Proc. of ACM WUWNet, San Francisco, CA, Sep. 2008.*