

# A Framework for Throughput and Energy Efficiency in Mobile ad Hoc Networks

Lucia Gallina, Sardaouna Hamadou, Andrea Marin, Sabina Rossi

► **To cite this version:**

Lucia Gallina, Sardaouna Hamadou, Andrea Marin, Sabina Rossi. A Framework for Throughput and Energy Efficiency in Mobile ad Hoc Networks. IFIP Wireless Days Conference, Oct 2011, Niagara Falls, ON, Canada. IEEE, pp.1-6, 2011, Wireless Days. <hal-00799168>

**HAL Id: hal-00799168**

**<https://hal.inria.fr/hal-00799168>**

Submitted on 11 Mar 2013

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# A Framework for Throughput and Energy Efficiency in Mobile ad Hoc Networks

Lucia Gallina and Sardaouna Hamadou and Andrea Marin and Sabina Rossi

Università Ca' Foscari Venezia, Italy

E-mail: {lgallina,sh,marin,rossi}@dais.unive.it

**Abstract**—In this paper, we propose a general framework to evaluate and compare both throughput and energy efficiency of mobile ad hoc networks. This framework is built upon a probabilistic, energy-aware, calculus for the analysis of broadcast, unicast and multicast wireless communications. The results of our analyses can be used to replace a network component with a more efficient one - with respect to different metrics like throughput and energy consumption - while maintaining connectivity.

As an application, we evaluate two well-known automatic repeat request (ARQ)-based error control protocols: stop-and-wait (SW) and go-back-N (GBN).

**Keywords**—*process algebra; manets; network throughput; energy consumption*

## I. INTRODUCTION

Mobile ad hoc networks (MANETs) are of paramount importance in today's communication networks. They rely on a collection of mobile devices communicating with each other through wireless links without a pre-established networking infrastructure. These devices usually share the following features: (1) they move independently from each other and will therefore change their communication links frequently; (2) they often rely on exhaustible power sources, such as batteries, and hence have strict requirements on the energy consumption.

Unlike wired networks for which the topology is fixed (except in cases of link failures), mobile ad hoc networks have dynamic topologies based on the location of the devices. The mechanism for computing the optimal transmission range for each node is called *topology control*, whose goal is to optimize performance metrics such as network lifetime and throughput while maintaining connectivity.

In this paper we propose a general framework to evaluate and compare both throughput and energy efficiency of mobile ad hoc networks. It is based on the probabilistic, energy-aware, calculus for mobile ad hoc networks introduced in [2]. This calculus allows us to model the ability of a node to broadcast a message to any other node within its physical transmission range, and to move in and out of the transmission range of other nodes in the network. Broadcast communications are limited to the transmission cell of the sender, while unicast and multicast communications are modelled by specifying, for each output action, the addresses of the intended recipients of the message. Moreover, the possibility for a node to control its transmission power is modeled by enabling nodes to modify the transmission radius of their communications. We define an energy-aware preorder over networks which allows us

to compare the average energy and time cost of different networks but exhibiting the same connectivity behaviour.

As an application, we evaluate and compare two well-known automatic repeat request (ARQ)-based error control protocols: stop-and-wait (SW) and go-back-N (GBN).

*Related works.* Probabilistic and stochastic models are nowadays widely used in the design and verification of complex systems. Song and Godsken [11] proposed a probabilistic broadcast calculus for mobile and wireless networks whose connections are unreliable. Palamidessi et al. [3] defined an extension of the applied pi-calculus with nondeterministic and probabilistic choice operators, while Priami [7] introduced a stochastic extension of the pi-calculus which allows one to describe dynamically reconfigurable or mobile networks. Jane Hillston introduced the stochastic process algebra PEPA [4], used for modelling systems composed of concurrently active components which co-operate and share work. Bernardo et al. introduced  $EMPA_{gr}$  [1], an extended Markovian process algebra including probabilities and priority.

As far as energy consumption is concerned, in [13] the authors define a Markov Reward process [8] modelling some protocols for pairwise node communications. In [10] the authors define a set of metrics on the energy consumption which are then estimated through simulation and show how some changes in the protocols can improve the efficiency.

With respect to the above mentioned works, the model proposed in this paper aims at providing a common framework for both qualitative and quantitative analysis.

*Plan of the paper.* Section II presents the probabilistic E-BUM calculus. Section III proposes a general framework to evaluate the cost of mobile wireless networks. In Section IV we analyse and compare the stop-and-wait (SW) and go-back-N (GBN) protocols. Finally, Section V concludes the paper.

## II. THE CALCULUS

We present the Probabilistic E-BUM calculus [2] which models mobile ad hoc networks as a collection of nodes, running in parallel, and using channels to broadcast messages. It supports multicast communications as well as power control.

*Syntax.* We use  $c$  for *channels*,  $n$  for *nodes*,  $l$  and  $k$  for *locations*,  $r$  for *transmission radii*;  $x$  for *variables*. *Closed values* contain nodes, locations, transmission radii and any basic value (booleans, integers, ...). *Values* include also variables. We use  $u$  and  $v$  for closed values and  $w$  for (open) values. We write  $\tilde{v}$ ,  $\tilde{w}$  for tuples of values. We write  $Loc$  for the set of all locations.

Networks are defined as the parallel composition of distinct nodes. We write  $n[P]_l$  for a node  $n$ , located at the location  $l$  and executing the process  $P$ . We denote by  $M_1|M_2$  the parallel composition of two networks and by  $\prod_{i \in I} M_i$  the parallel composition of the networks  $M_i$ , for  $i \in I$ . Nodes cannot be created or destroyed. Processes  $P$  have the form:

- $\mathbf{0}$  denotes the inactive process;
- $c(\tilde{x}).P$  can receive a tuple  $\tilde{w}$  of (closed) values via channel  $c$  and continue as  $P\{\tilde{w}/\tilde{x}\}$ , i.e., as  $P$  with  $\tilde{w}$  substituted for  $\tilde{x}$ ;
- $\bar{c}_{L,r}(\tilde{w}).P$  can send a tuple of (closed) values  $\tilde{w}$  via channel  $c$  and continue as  $P$ . The tag  $L$  is used to maintain the set of locations of the intended recipients:  $L = \infty$  represents a broadcast transmission, while a finite set of locations  $L$  denotes a multicast communication (unicast if  $L$  is a singleton). The tag  $r$  represents the power of the transmission. Syntactically, the tags  $L$  and  $r$  may be variables, but they must be instantiated when the output prefix is ready to fire.
- $[w_1 = w_2]P, Q$  behaves as  $P$  if  $w_1 = w_2$ , as  $Q$  otherwise.
- $A\langle\tilde{w}\rangle$  denotes a process defined via a (possibly recursive) definition  $A(\tilde{x}) \stackrel{\text{def}}{=} P$ , with  $|\tilde{x}| = |\tilde{w}|$  where  $\tilde{x}$  contains all channels and variables that appear free in  $P$ .

In the process  $c(\tilde{x}).P$ , the variables in  $\tilde{x}$  are bound in  $P$ . We identify processes up to  $\alpha$ -conversion and we assume that there are no free variables in a network. We write  $c_l$  for  $c_{\{l\}}$ .

Each node  $n$  is associated with a pair  $\langle r_n, \mathbf{J}^n \rangle$ , where  $r_n$  is a non negative real number denoting the maximum transmission radius that  $n$  can use to transmit, while  $\mathbf{J}^n$  is the transition matrix of a discrete time Markov chain, where  $\mathbf{J}_{lk}^n$  is the probability that the node  $n$  located at  $l$ , after executing a movement, will be located at  $k$ . Hence,  $\sum_{k \in Loc} \mathbf{J}_{lk}^n = 1$  for  $l \in Loc$ . Static nodes are associated with the identity Markov chain,  $\mathbf{J}_{ll}^n = 1$  for  $l \in Loc$  and  $\mathbf{J}_{lk}^n = 0$  for  $l \neq k$ .

*Probability distributions.* In our framework, the mobility of the nodes is the only source of probability. We associate probability distributions with located nodes and model the probabilistic evolution of the network according to these distributions. More formally, we denote by  $\mu_l^n$  the probability distribution associated with the node  $n$  located at  $l$ , that is a function over  $Loc$  such that  $\mu_l^n(k) = \mathbf{J}_{lk}^n$ , for  $k \in Loc$ .

Let  $M = \prod_{i \in I} n_i[P_i]_{l_i}$  be a network, then for all  $k$  in  $I$ ,  $\llbracket M \rrbracket_{\mu_{l_k}^{n_k}}$  denotes the probability distribution over the set of networks induced by  $\mu_{l_k}^{n_k}$ , i.e., for all network  $M'$ :

$$\llbracket M \rrbracket_{\mu_{l_k}^{n_k}}(M') = \begin{cases} \mu_{l_k}^{n_k}(l'_k) & \text{if } M' = \prod_{i \in I} n_i[P_i]_{l'_i} \\ & \text{for } l'_i = l_i \forall i \neq k \\ 0 & \text{otherwise} \end{cases}$$

Note that  $\llbracket M \rrbracket_{\mu_{l_k}^{n_k}}(M')$  is the probability that the network  $M$  evolves to  $M'$  due to the movement of the node  $n_k$  located at  $l_k$ . We say that  $M'$  is in the support of  $\llbracket M \rrbracket_{\mu_{l_k}^{n_k}}$  if  $\llbracket M \rrbracket_{\mu_{l_k}^{n_k}}(M') \neq 0$ . We write  $\llbracket M \rrbracket_{\Delta}$  for the Dirac distribution on the network  $M$ , namely the probability distribution defined as:  $\llbracket M \rrbracket_{\Delta}(M) = 1$  and  $\llbracket M \rrbracket_{\Delta}(M') = 0$  for all  $M' \neq M$ . We let  $\theta$  range over  $\{\mu_l^n \mid n \text{ is a node and } l \in Loc\} \cup \{\Delta\}$ .

*Reduction Semantics.* The dynamics of the calculus is specified

by the *probabilistic reduction relation* over networks ( $\rightarrow$ ), described in Table I. As usual in process calculi, it relies on an auxiliary relation, called structural congruence ( $\equiv$ ), such that for instance  $M|N \equiv N|M$ ,  $(M|N)|M' \equiv M|(N|M')$  and  $M|\mathbf{0} \equiv M$ . The probabilistic reduction relation takes the form  $M \rightarrow \llbracket M' \rrbracket_{\theta}$ , which describes a transition that leaves from network  $M$  and leads to a probability distribution  $\llbracket M' \rrbracket_{\theta}$ .

Rule (R-Bcast) models the transmission of a tuple of messages  $\tilde{v}$  to the set of intended recipients  $L$  using channel  $c$  and transmission radius  $r$ . Indeed, nodes communicate using radio frequencies that enable only message broadcasting. However, a node may decide to communicate with a specific set of nodes  $L$ : if  $L = \infty$  then the recipients set is the whole network and a broadcast transmission is performed, while if  $L$  is finite then a multicast (or a unicast) communication is realized. The tag  $r$  indicates the transmission radius required for that communication and may depend on the energy consumption strategy adopted by the surrounding protocol. Notice that a transmission is a *non-blocking action*: it proceeds even if there are no nodes listening for messages. The messages transmitted may be received only by those nodes which lie in the transmission area of the sender. Function  $d(\cdot, \cdot)$  takes two locations and returns the distance separating them.

Rule (R-Move) deals with node mobility. A node  $n$  located at  $l$  and executing a moving action will reach a location with a probability described by the distribution  $\mu_l^n$  that depends on the Markov chain  $\mathbf{J}^n$  associated with  $n$ . Movements are atomic actions: while moving, a node cannot do anything else.

Rules (R-Par) and (R-Struct) are standard.

Given a network  $M$ , we write  $M \rightarrow_{\theta} N$  if  $M \rightarrow \llbracket M' \rrbracket_{\theta}$  and  $N$  is in the support of  $\llbracket M' \rrbracket_{\theta}$ . An execution for  $M$  is a (possibly infinite) sequence of steps  $M \rightarrow_{\theta_1} M_1 \rightarrow_{\theta_2} M_2 \dots$ . In the rest of the paper, we write  $last(e)$  for the final state of a *finite* execution  $e$ ,  $e^j$  for the prefix execution  $M \rightarrow_{\theta_1} M_1 \dots \rightarrow_{\theta_j} M_j$  of length  $j$  of the execution  $e = M \rightarrow_{\theta_1} M_1 \dots \rightarrow_{\theta_j} M_j \rightarrow_{\theta_{j+1}} M_{j+1} \dots$ . We denote by  $behave(M) = \{\llbracket M' \rrbracket_{\theta} \mid M \rightarrow \llbracket M' \rrbracket_{\theta}\}$  the set of the possible behaviours of  $M$ . In order to solve the non-determinism in a network execution, we consider each possible probabilistic transition  $M \rightarrow \llbracket M' \rrbracket_{\theta}$  as arising from a *scheduler* (see [9]). A *scheduler* is a total function  $F$  assigning to a finite execution  $e$  a distribution  $\llbracket N \rrbracket_{\theta} \in behave(last(e))$ . Given a network  $M$  and a scheduler  $F$ , we denote by  $Exec_M^F$  the set of executions starting from  $M$  and driven by  $F$ , i.e.,

$$Exec_M^F = \{M \rightarrow_{\theta_1} M_1 \rightarrow_{\theta_2} M_2 \dots \mid \forall j, M_{j-1} \rightarrow \llbracket M'_j \rrbracket_{\theta_j}, \llbracket M'_j \rrbracket_{\theta_j} = F(e^{j-1}) \text{ and } \llbracket M'_j \rrbracket_{\theta_j}(M_j) > 0\}.$$

Moreover, for a finite execution  $e = M \rightarrow_{\theta_1} M_1 \dots \rightarrow_{\theta_k} M_k$  starting from  $M$  and driven by a scheduler  $F$  we define

$$P_M^F(e) = \llbracket M'_1 \rrbracket_{\theta_1}(M_1) \cdot \dots \cdot \llbracket M'_k \rrbracket_{\theta_k}(M_k)$$

where  $\forall j \leq k$ ,  $\llbracket M'_j \rrbracket_{\theta_j} = F(e^{j-1})$ . Given a measurable set of networks  $H$ , we denote by  $Exec_M^F(H)$  the set of executions starting from  $M$  and crossing a state in  $H$ , i.e.,  $Exec_M^F(H) = \{e \in Exec_M^F \mid last(e^j) \in H \text{ for some } j\}$ .

---

$\text{(R-Bcast)} \frac{n[\bar{c}_{L,r}\langle\tilde{v}\rangle.P]_l \mid \prod_{i \in I} n_i[c(\tilde{x}_i).P_i]_{l_i} \rightarrow \llbracket n[P]_l \mid \prod_{i \in I} n_i[P_i\{\tilde{v}_i/\tilde{x}_i\}]_{l_i} \rrbracket_{\Delta}}{\text{where } 0 < r \leq r_n, \forall i \in I. d(l, l_i) \leq r \text{ and }  \tilde{x}_i  =  \tilde{v}_i }$
$\text{(R-Move)} \frac{}{n[P]_l \rightarrow \llbracket n[P]_l \rrbracket_{\mu_r^p}} \quad \text{(R-Par)} \frac{M \rightarrow \llbracket M' \rrbracket_{\theta}}{M \mid N \rightarrow \llbracket M' \mid N \rrbracket_{\theta}} \quad \text{(R-Struct)} \frac{M \rightarrow \llbracket M' \rrbracket_{\theta} \quad M' \equiv N'}{M \rightarrow \llbracket N' \rrbracket_{\theta}}$

---

TABLE I: Reduction Semantics

In [2] we introduced a probabilistic observational congruence, denoted  $\cong_p^{\mathcal{F}}$ , in the style of [3], such that  $M \cong_p^{\mathcal{F}} N$  if  $M$  and  $N$  exhibit the same probabilistic behaviour, in terms of communications relative to the corresponding sets of intended recipients and the fixed set of schedulers  $\mathcal{F}$ . A bisimulation-based proof technique for  $\cong_p^{\mathcal{F}}$  is also defined.

### III. MEASURING EFFICIENCY

Based on the above semantics, we propose here a flexible and adaptable method to evaluate the cost of mobile wireless networks. We assume quasilinear cost functions of the form

$$\sum_k \rho_k \cdot \Phi_k(\_) \quad (1)$$

that is, a sum of arbitrary factor functions  $\Phi_k(\_)$  each of which maps one metric (e.g., energy cost, throughput, noninterference level) to some common measure (e.g., the  $[0, 1]$  interval). We will in general give a different weight  $\rho_k \geq 0$  to each factor  $\Phi_k(\_)$  to reflect its importance. We assume that  $\sum_k \rho_k = 1$ .

Here we consider *energy cost* and *time cost*: they often need to be traded off against each other, as a higher throughput level often requires more complex protocols which, as a side effect, reflect in higher energy consumption. The ability to give the factors different weights in the *same* cost function allows us to select the optimal strategy to finely trade them off.

*Energy Cost.* We associate an energy cost with the probabilistic network reductions as follows:

$$\mathbf{E}(M, N) = \begin{cases} r & \text{if } M \rightarrow_{\theta} N, M \equiv n[\bar{c}_{L,r}\langle\tilde{v}\rangle.P]_l \mid M', \\ & N \equiv n[P]_l \mid N' \text{ for some } c, L, \tilde{v}, l \\ 0 & \text{otherwise.} \end{cases}$$

In other words, the energy cost to reach  $N$  from  $M$  in one single reduction step is  $r$  if  $M$  can reach  $N$  after firing on a channel of radius<sup>1</sup>  $r$ .

*Time Cost.* We assume the following two auxiliary functions. The first, denoted  $\mathcal{T}_b$ , associates a duration  $\mathcal{T}_b(c, \tilde{v})$  to the broadcasting of  $\tilde{v}$  over  $c$ , whilst  $\mathcal{T}_m$  returns the duration  $\mathcal{T}_m(n, l, k)$  of the movement of node  $n$  from its current position  $l$  to the position<sup>2</sup>  $k$ . In order to compute the average

time a network takes to reach a given set of states, we associate a duration with the reductions as follows.  $\mathbf{T}(M, N) =$

$$\begin{cases} \mathcal{T}_b(c, \tilde{v}) & \text{if } M \rightarrow_{\Delta} N, M \equiv n[\bar{c}_{L,r}\langle\tilde{v}\rangle.P]_l \mid M' \\ & \text{and } N \equiv n[P]_l \mid N' \text{ for some } c, L, v, l \\ \mathcal{T}_m(n, l, k) & \text{if } M \rightarrow_{\mu_r^p} N, M \equiv n[P]_l \mid M' \\ & \text{and } N \equiv n[P]_k \mid M' \end{cases}$$

Now let  $\mathcal{X} \in \{\mathbf{E}, \mathbf{T}\}$  be one of the two metrics above, and  $e = M_0 \rightarrow_{\theta_1} M_1 \rightarrow_{\theta_2} M_2 \cdots \rightarrow_{\theta_k} M_k$  be an execution.

$$\mathcal{X}(e) = \sum_{i=1}^k \mathcal{X}(M_{i-1}, M_i).$$

Let  $H$  be a set of networks, we denote by  $Paths_M^F(H)$  the set of all executions from  $M$  ending in  $H$  and driven by  $F$  which are not prefix of any other execution ending in  $H$ . More formally,  $Paths_M^F(H) = \{e \in Exec_M^F(H) \mid last(e) \in H \text{ and } \forall e' \text{ such that } e <_{prefix} e', e' \notin Paths_M^F(H)\}$ .

*Definition 3.1:* Let  $H$  be a set of networks and  $\mathcal{X} \in \{\mathbf{E}, \mathbf{T}\}$ . The average energy and time cost of reaching  $H$  from  $M$  according to the scheduler  $F$  is

$$\mathcal{X}_M^F(H) = \frac{\sum_{e \in Paths_M^F(H)} \mathcal{X}(e) \times P_M^F(e)}{\sum_{e \in Paths_M^F(H)} P_M^F(e)}.$$

The average energy and time cost is computed by weighting the cost of each execution by its probability according to  $F$  and normalized by the overall probability of reaching  $H$ .

*Cost Function.* Now given  $\Phi_E$  and  $\Phi_T$  mapping energy and time costs to some common measure respectively, and  $\rho$  the weight of the energy factor, we define the cost of reaching a set  $H$  from the network  $M$  according to the scheduler  $F$  as:

$$\mathbf{Cost}_M^F(H) = \rho \Phi_C(\mathbf{E}_M^F(H)) + (1 - \rho) \Phi_T(\mathbf{T}_M^F(H)). \quad (2)$$

*Definition 3.2:* Let  $\mathcal{H}$  be a countable set of sets of networks and  $\mathcal{F}$  be a set of schedulers. We say that  $N$  is *more efficient than*  $M$  relative to  $\mathcal{F}$  and  $\mathcal{H}$ , denoted

$$N \sqsubseteq_{(\mathcal{F}, \mathcal{H})} M,$$

if  $N \cong_p^{\mathcal{F}} M$  and, for all schedulers  $F \in \mathcal{F}$  and for all  $H \in \mathcal{H}$ , there exists  $F' \in \mathcal{F}$  such that  $\mathbf{Cost}_{N'}^{F'}(H) \leq \mathbf{Cost}_M^F(H)$ .

### IV. ANALYSIS OF ARQ PROTOCOLS

In this section we show how the proposed calculus allows us to formally describe a network with two nodes whose relative

<sup>1</sup>Note that considering the radius of the communication channel as the energy cost of the transmitted data is standard.

<sup>2</sup>With the convention that  $\mathcal{T}_m(n, l, k) = +\infty$  iff  $\mu_r^n(k) = 0$ .

position changes over the time. They communicate according to a Positive Acknowledge Request (PAR or ARQ) protocol in which the sender's windows size is  $k \in \mathbb{N}$  while the receiver's one is 1. These assumptions specify a Go-Back-N (GBN) protocol if  $k > 1$  or a Stop & Wait (SW) if  $k = 1$ . Informally, when a packet is lost for some reasons, the sender must wait at least a whole Round Trip Time (RTT) in order to be aware of this fact. Then it resends all the packets starting from the lost one because the receiver has discarded all the following ones (even if correctly received) since their sequence numbers are incorrect and its window size is 1.

From the qualitative point of view the two protocols exhibit an equivalent behaviour, i.e., they ensure a reliable communication over an unreliable (wireless) link. However, some major quantitative differences may be conjectured according to the sender's window size. Indeed, when its size is close to 1 retransmissions of potentially correctly received packets are very unlikely and hence the protocol tends to reduce the energy consumption. On the other hand, small sender's window size do not allow the device to exploit the pipelining of packets. Indeed, assuming a bandwidth  $b$ , and a frame size  $f$ , the maximum throughput is obtained when the window size is  $RTT \cdot b/f$  (see e.g., [12]).

*Assumptions on the model.* We consider the case of a mobile station  $A$  which communicates with a static station  $B$ . The former has a limited transmission power, hence when it moves it may happen that station  $B$  is outside the transmission range of its antenna. Whereas, we assume that  $B$  is always able to send to  $A$  the acknowledge for the correctly received packets. Time is discrete and divided into slots. The size of a time slot is  $f/b$ , i.e., the time required by the sender to push a packet into the channel.  $A$  movements occur according to a discrete time Markov chain  $J^s$  with two states representing the two possible locations,  $l_c$  and  $l_e$ . The transition matrix is:

$$P = \begin{bmatrix} p_{cc} & p_{ce} \\ p_{ec} & p_{ee} \end{bmatrix},$$

where  $p_{cc} + p_{ce} = p_{ec} + p_{ee} = 1$ . Station  $A$  keeps sending one packet each time slot, and this is received if the sender location is  $l_c$ , while it is lost otherwise. We ignore other possible reasons of packet loss. We are interested only in the energy consumption of the mobile device, hence we assume that the energy cost for sending an acknowledge is negligible.

*Modelling the Protocols.* In our analysis, we assume that the energy consumption of the feedback messages is negligible. Therefore, they are sent over channels with zero radius. For this reason the static receiver  $rec$  is located at  $l_c$ , i.e., at the same location of the sender in its good state, so that the feedback will be received with no cost. Note that the sender still transmits over channels with radius  $r$  and thus it consumes an amount of energy equal to  $r$  for each fired packet.

The process executed by  $rec$ , the receiver node, is

$$REC\langle i \rangle = c^{(i)}(x).\bar{c}_{l_1,0}\langle ack(i) \rangle.REC\langle i+1 \rangle$$

which, upon receiving packet  $p_i$  over the channel  $c^{(i)}$ , sends

$ack(i)$  over  $c$ , then waits for the next packet on  $c^{(i+1)}$ .

For each channel  $c^{(i)}$ , we use a static auxiliary node  $b_i$  ( $(0, I)$ ) located at  $l_e$ , the bad state of the sender, capturing bad transmissions over  $c^{(i)}$ . It executes the following process which upon receiving packet  $p_i$  over the channel  $c^{(i)}$ , sends  $nack(i)$  over the channel  $c$ :

$$BAD\langle i \rangle = c^{(i)}(x).\bar{c}_{\emptyset,0}\langle nack(i) \rangle.BAD\langle i \rangle.$$

The sender's window size is  $k$ , whilst the full capacity of the communication channel is  $n$ . The sender node is:

$$GB\langle i, k \rangle = \bar{c}_{\emptyset,r}^{(i)}\langle p_i \rangle.c(x_1).\bar{c}_{\emptyset,r}^{(i+1)}\langle p_{i+1} \rangle.c(x_2) \cdots \bar{c}_{\emptyset,r}^{(i+k-1)}\langle p_{i+k-1} \rangle.c(x_k).WAIT\langle i, k, n-k, x_1, x_2, \dots, x_k \rangle$$

where  $WAIT$  is the process

$$WAIT\langle i, k, j, v_1, v_2, \dots, v_k \rangle = [j \neq 0]\bar{d}_{\emptyset,0}\langle \rangle. \\ WAIT\langle i, k, j-1, v_1, v_2, \dots, v_k \rangle, ([v_1 = nack(i)] \\ GB\langle i, k \rangle, SEND\langle i+k, k, v_2, \dots, v_k \rangle)$$

and the process  $SEND$  is defined as follows

$$SEND\langle i, k, v_1, v_2, \dots, v_{k-1} \rangle = \bar{c}_{\emptyset,r}^{(i)}\langle p_i \rangle.c(z). \\ [v_1 = nack(i-k)]GB\langle i-k, k \rangle, \\ SEND\langle i+1, k, v_2, \dots, v_{k-1}, z \rangle.$$

Though that the feedback of a packet is received after a whole Round Trip Time and the transmission of its  $(k-1)$  successors, for practical reason, we read a feedback of a packet right after sending it. Indeed, since we do not want feedback to be costly, both sender and receiver must be located at the same place when the feedback is sent. However, the sender node will verify it only after having sent the following  $(k-1)$  packets and having been waiting for the whole RTT. The latter is modelled by the process  $WAIT$ .

Recall that the receiver node in our modelling above, reads each packet  $p_i$  on its specific channel  $c^{(i)}$ . Thus, if the transmitter sends  $p_i$  while being in its good state, then moves to bad, sends  $p_{i+1}$  and finally moves back to the good state and sends  $p_{i+2}$ , hence the latter packet will not be read by the receiver as it is blocked on  $c^{(i+1)}$ . Then, the firing on  $c^{(i+2)}$  is lost and this models the fact that packets sent after a bad packet are just wasted. But since the sender process  $GB\langle i, k \rangle$  is blocked on the feedback channel  $c$ , we introduce a static auxiliary node  $loose$  located at  $l_c$  and executing the process:

$$WAST = \bar{c}_{\emptyset,0}\langle LOST \rangle.WAST$$

The full protocol, where  $k$  is the senders windows size, is:

$$GBN\langle k \rangle = send[GB\langle 1, k \rangle]_{l_c} | rec[REC\langle 1 \rangle]_{l_c} \\ | loose[WAST]_{l_c} | \prod_{i \geq 1} b_i[BAD\langle i \rangle]_{l_e}$$

*Behavioural equivalence of the protocols obtained with different values of  $k$ .* In order to compare the observational behaviours of the protocols, we assume that the communications over the feedback channel are observable for any observer node located at  $l_c$ . Thus  $GBN\langle k \rangle$  and  $GBN\langle k' \rangle$  are equivalent with respect to a set of schedulers  $\mathcal{F}$  if for all schedulers  $F$  in  $\mathcal{F}$  driving one of the protocols, there exists

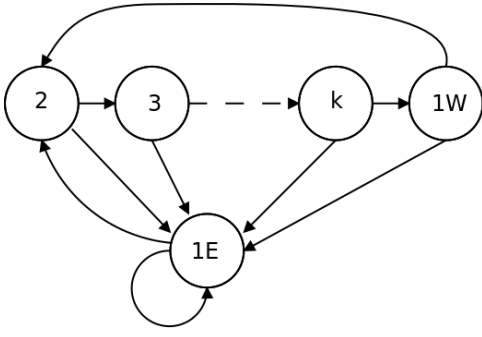


Fig. 1: DTMC after the lumping. Transition probabilities are:  $i \rightarrow (i+1) : p_{cc}, 1 < i < k; i \rightarrow 1E : p_{ce}, 1 < i \leq k; k \rightarrow 1W : p_{cc}^*(n-k+1); 1W \rightarrow 1E : p_{ce}^*(n-k+1); 1W \rightarrow 2 : p_{cc}^*(n-k+1); 1E \rightarrow 1E : p_{ec}^*(n); 1E \rightarrow 2 : p_{ec}(n).$

a scheduler  $F'$  in  $\mathcal{F}$  driving the other one such that both protocols correctly transmit the same packets with the same probabilities. modelling communication protocols. However, many schedulers could be in fact unrealistic. Consider for example schedulers giving priority to communication actions over movements of the nodes. Such schedulers cancel the two states nature of the communication channel since the sender remains in the same state until there is no longer available communication action. Thus, if the network started in its good state then all the messages will be transmitted correctly without enduring any lost. In contrast, if it started in its bad state, then it will be retransmitting indefinitely the first packet since the channel remains always bad. Though, under such schedulers, both protocols behave exactly the same way in terms of our observability, they represent however unrealistic implementation scenarios. Therefore, we consider the following set of schedulers denoted  $\mathcal{F}_{alt}$  which: 1) always alternate between sending packets and node's movement so that at each interaction of the transmitter with the receiver, the former can be either in a good or bad state; 2) give priority to acknowledgment actions (*ack* and *nack*) to model the standard assumption of an error-free feedback channel; 3) allow interaction with the outside environment only through its observable actions so that we capture exactly the observable behaviour of the protocol. Under these assumptions, we can prove that both protocols exhibit the same observable behaviour for  $k, k' \in \{1, \dots, n\}$ .

*Proposition 4.1:*  $GBN\langle k \rangle \approx_{\mathcal{F}_{alt}} GBN\langle k' \rangle$ .

*Quantitative evaluation of the protocols.* Now we compare the cost of our protocols in the context of the set  $\mathcal{H} = \{H_j | j \geq 1\}$  where  $H_j$  means that all the packets up to  $j-1$  have been correctly transmitted and is defined as

$$H_j = \{M | M \equiv rec[REC(j)]_{l_c} | N, \text{ for some } N\}.$$

In the following, we derive the expected energy and time costs for each correct packet transmission required by the considered protocols (each one obtained using a different window size  $k$ ). As, e.g., in [6], we focus on the stationary

analysis, which applies when the number of transitions that occur from the initial instant tends to infinity. The expected energy cost ( $\bar{E}$ ) should be interpreted as follows:

$$\bar{E} = \sum_{i=1}^k \left( \mathbf{E}_{M_{i,j}}^F(H_j) \right) \pi_i,$$

where  $M_{i,j}$  denotes the network in which the packet with sequence number  $j$  is going to be sent from window position  $i$ .  $\pi_i$  denotes the steady-state probability of observing the sender-side window of the protocol sending the packet from position  $i$ . In a completely analogous way we define  $\bar{T}$  as the expected time required for a successful transmission.

In order to keep the protocol encoding simple, each sequence number is used only once to identify one packet, and this leads to an underlying process with infinite state space. However, it is easy to observe that states may be opportunely clustered according to the window position in which one packet is sent. This turns to be an exact lumping (see e.g., [5]) of the infinite state space DTMC into the finite state space depicted in Figure 1. It is worth to point out that state  $i$  with  $2 \leq i \leq k$  of the DTMC do not denote the window size, but the position taken by the packet which is being sent. State  $1E$  represents the situation in which a transmission error occurred and the next packet will be sent from position 1 of the window, whereas state  $1W$  denotes the situation in which the next packet will be in position 1 of the window but the sender needs to wait for an acknowledge.

Observe that the ergodicity of the underlying Markov chain is ensured by the fact that it is irreducible, finite and trivially aperiodic. We recall that  $p_{l_1 l_2}$  with  $l_1, l_2 \in \{l_e, l_c\}$  denotes the one-step transition probability of the discrete time Markov Chain underlying the movement of the stochastic process, and  $p_{l_1 l_2}^*(i)$  with  $i \in \mathbb{N}$  denotes  $l_1, l_2$  elements of the  $i$ -step transition probability matrix  $P^i$ .

*Proposition 4.2:* [Expected energy cost in steady-state] The expected energy cost for  $m$  received packets, with sender's window size  $k$ , is given by the following expression:

$$\bar{E} = \left( 1 + \frac{p_{ce} (1 - p_{cc}^{k-1} p_{cc}^*(n-k+1)) k}{p_{ec}^*(n) (1 - p_{cc}^k)} \right) m \cdot r$$

In order to keep the time cost computing simple, we assume that the time it takes to send a packet or an acknowledgement is negligible. Thus  $\mathcal{T}_b(c, \bar{v}) = 0$  for each channel and each message. Therefore we associate one full time-slot to the movement of the sender node, i.e.,  $\mathcal{T}_m(send, l, l') = 1$  for  $l, l' \in \{l_c, l_e\}$ . We express the time in terms of the number of time-slots and summarize it in the following proposition.

*Proposition 4.3:* The expected number of time-slots taken from  $m$  received packets (time cost), with sender's window size  $k$ , is given by the following expression:

$$\bar{T} = \left( 1 + \frac{p_{cc}^{k-1} p_{ce} (n-k)}{(1 - p_{cc}^k)} + \frac{p_{ce} (1 - p_{cc}^{k-1} p_{cc}^*(n-k+1)) n}{p_{ec}^*(n) (1 - p_{cc}^k)} \right) m.$$

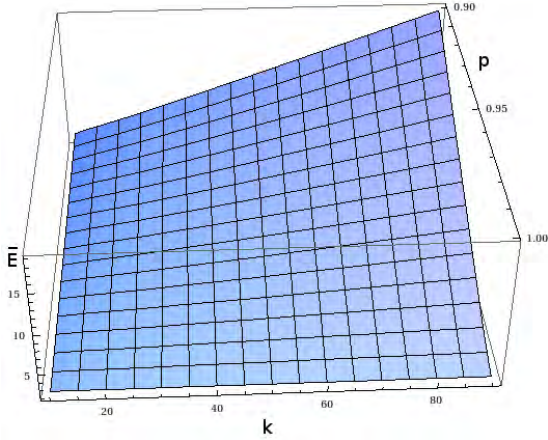


Fig. 2: Energy cost of ARQ protocols.

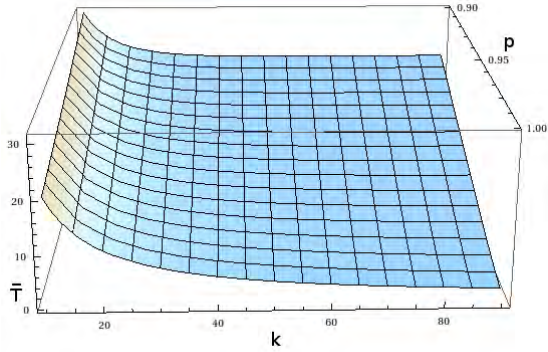


Fig. 3: Time cost of ARQ protocols.

We now conclude this section with an optimality analysis. This should be considered as an illustrative application of our calculus that aims to show how behavioural and standard quantitative results about a model can be integrated. Figure 2 and 3 show the plotting of  $\bar{E}$  and  $\bar{T}$  (where  $m = r = 1$ ) as functions of the probability of movement  $1 - p = p_{ce} = p_{ec}$  and the sender window size  $k$ . We assume a maximum window size of  $n = 100$ . Finally, we define function  $\overline{\text{Cost}}$  as:

$$\overline{\text{Cost}} = \sum_{i=1}^n \text{Cost}_{M_{i,j}}^F(H_j)\pi_i,$$

where  $M_{i,j}$ ,  $H_j$ ,  $\pi_i$  are defined as before, while  $\text{Cost}_{M_{i,j}}(H_j)$  is obtained according to Equation (2) with  $\Phi_-$  as the identity function and  $\rho = \frac{1}{2}$ . Function  $\overline{\text{Cost}}$  is depicted in Figure 4. For the parameters in Figure 4, a sender's window size of 30 gives the optimal cost. Indeed, for all  $k \in \{1, \dots, 100\}$  it holds:

*Proposition 4.4:*  $GBN\langle 30 \rangle \sqsubseteq_{\mathcal{F}_{alt}, \mathcal{H}} GBN\langle k \rangle$ .

## V. CONCLUSION

In this paper, we proposed a general framework to evaluate and compare the cost of such networks. We believe that our model is applicable and realistic because its cost functions:

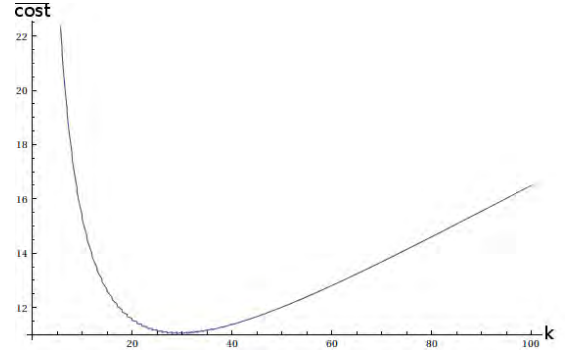


Fig. 4:  $\overline{\text{Cost}}$  function for ARQ protocols.

1) are flexible and adaptable to the specific applications; 2) are composed of different independently-configurable metrics, e.g., energy cost, delivery time, interference level, which can be extended or canceled or augmented; 3) have weights which allow us to model different importance level for each factor of interest.

## ACKNOWLEDGMENTS

Work partially supported by the MIUR Project IPODS “Interacting Processes in Open-ended Distributed Systems”.

## REFERENCES

- [1] M. Bernardo and M. Bravetti. Performance measure sensitive congruences for markovian process algebras. *Theoretical Computer Science*, 290(1):117 – 160, 2003.
- [2] L. Gallina, S. Hamadou, A. Marin, and S. Rossi. A probabilistic energy-aware model for mobile ad-hoc networks. In *Proc. of the 18th International Conference on Analytical and Stochastic Modelling Techniques and Applications (ASMTA'11)*, volume 6751 of *LNCS*, pages 316–330. Springer-Verlag, 2011.
- [3] J. Goubault-Larrecq, C. Palamidessi, and A. Troina. A probabilistic applied pi-calculus. In *Proc. of the 5th Asian Symposium on Programming Languages and Systems (APLAS '07)*, volume 4807/2009 of *LNCS*, pages 175–190. Springer-Verlag, 2007.
- [4] J. Hillston. *A Compositional Approach to Performance Modelling*. Cambridge University Press, 1996.
- [5] J. G. Kemeny and J. L. Snell. *Finite Markov chains*. D. Van Nostrand Company, inc., 1960.
- [6] L.B. Le, E. Hossain, and M. Zorzi. Queueing analysis for gbn and sr arq protocols under dynamic radio link adaptation with non-zero feedback delay. *IEEE Transactions on Wireless Communications*, 6(9):3418–3428, 2007.
- [7] C. Priami. Stochastic  $\pi$ -calculus. *The Computer Journal*, 38(7):578–589, 1995.
- [8] S. M. Ross. *Stochastic Processes*. John Wiley & Sons, 2nd edition, 1996.
- [9] R. Segala and N.A. Lynch. Probabilistic simulations for probabilistic processes. In *Proc. of the 5th International Conference on Concurrency Theory (CONCUR'94)*, volume 836 of *LNCS*, pages 481–496. Springer-Verlag, 1994.
- [10] S. Singh, M. Woo, and C.S. Raghavendra. Power-aware routing in mobile ad hoc networks. In *Proc. of the 4th annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom '98)*, pages 181–190. ACM, 1998.
- [11] L. Song and J. Godskesen. Probabilistic mobility models for mobile and wireless networks. In *Theoretical Computer Science*, volume 323 of *IFIP Advances in Information and Communication Technology*, pages 86–100. Springer Boston, 2010.
- [12] Andrew S. Tanenbaum. *Computer Networks*. Prentice Hall, 2003.
- [13] M. Zorzi and R. R. Rao. Error control and energy consumption in communications for nomadic computing. *IEEE Transactions on Computers*, 46(3):279 – 289, 1997.