PCTMC models of Wireless Sensor Network protocols

Marcel C. Guenther Jeremy T. Bradley

Imperial College London, 180 Queen's Gate,
London SW7 2AZ, United Kingdom,
Email: {mcg05,jb}@doc.ic.ac.uk

Abstract. Wireless Sensor Networks (WSNs) consist of a large number of spatially distributed embedded devices (nodes), which communicate with one another via radio. Over the last decade improvements in hardware and a steady decrease in cost have encouraged the application of WSNs in areas such as industrial control, security and environmental monitoring. However, despite increasing popularity, the design of end-to-end software for WSNs is still an expert task since the choice of middleware protocols heavily influences the performance of resource-constrained WSNs. As a consequence, WSN designers resort to discrete event simulation prior to deploying networks. While such simulations are reasonably accurate, they tend to be computationally expensive to run, especially for large networks. This particularly limits the number of distinct protocol configurations that engineers can test in advance of construction and hence their final setup may be suboptimal. To mitigate this effect we discuss how highly efficient mean-field techniques can be brought to bear on models of wireless sensor networks. In particular, we consider the practical modelling issues involved in constructing appropriately realistic Population CTMC (PCTMC) models of WSN protocols.

Keywords: Mean-field Analysis, PCTMC Modelling, WSN Modelling

1 Introduction

Recent hardware improvements and decreasing deployment costs have increased the popularity of Wireless Sensor Network (WSN) in various application areas. Examples include security and surveillance [1], forest fire detection [2], structural monitoring and controlling [3,4] as well as wildlife habitat monitoring [5] and healthcare [6] to name but a few. The emphasis of WSNs is to sample different kinds of environment data and forward the information to data sinks for further processing and analysis. While the general architecture of such networks is simple, the challenge lies in guaranteeing a number of Quality of Service (QoS) constraints for different application scenarios. Most commonly, sensor network applications require a specific balance between energy-efficiency, link reliability, security, bandwidth, and latency.

To ensure, prior to installing a WSN, that the software meets QoS demands, many WSN designers simulate their applications using discrete event simulation

(DES) frameworks such as Castalia [7], ns 2/3 [8] and TOSSIM [9]. These low-level network simulators have fairly sophisticated models for channel noise and interference and generally provide a realistic simulation environment for WSN applications [10]. However, discrete event simulation becomes computationally expensive as we increase the number of nodes in the network [11]. Therefore predicting the behaviour of a large network for a particular configuration cannot be done in real-time and optimising protocols by means of parameter sweeping can become computationally infeasible even if it is done offline. Mean-field analysis methods [12] for Population Continuous Time Markov Chains (PCTMCs) may help to overcome this problem. Originally, PCTMC models were used to approximate molecule levels in chemical reaction systems [13,14]. Recently, this paradigm has gained popularity in the performance analysis community as an efficient means to study large scale client-server systems [15,16]. The use of PCTMC models for WSNs has been rare in the literature, despite encouraging results presented in [17,18]. One of the main reasons for this is that PCTMCs only allow negatively exponentially distributed state sojourn times,[1] which may at first seem unsuitable for WSN modelling since these networks feature many deterministic, clock driven state changes. In this paper we will illustrate that this does not necessarily disqualify PCTMCs as a useful modelling paradigm for WSNs. In particular when analysed using the fast mean-field analysis method, PCTMC models can be seen a heuristic tool that enables a designer to discount certain configurations without the need for expensive simulations.

Our paper is organised as follows. In Section 2 we present an overview over WSN hardware, middleware and other protocol related issues. Moreover, we formally introduce PCTMC models and mean-field analysis. Subsequently Section 3 looks at how WSNs can be represented as PCTMCs and further points out open modelling challenges. Section 4 compares an example PCTMC model of the dataflow behaviour in a fail-safe WSN to the behaviour observed in an analogous low-level Castalia simulation of the same network. In Section 5 we present our conclusions and propose further research opportunities.

2 Background

The most compelling reason for studying fast performance analysis techniques is that they allow designers to conduct real-time behavioural prediction and efficient offline parameter sweeping for large networks. While protocol parameterisation has often been ignored in former studies [19], recent WSN protocol research highlights the performance benefit of optimising protocols for a given environment. In [20] the authors use a low-level network simulation to optimise the IEEE 802.15.4 MAC protocol, to show that it can deliver good performance when tuned correctly. Due to the simulation complexity, however, the authors only investigate a limited number of parameter setups. Clearly, a faster analysis method would help to reject inefficient protocol setups without the need for

[1] In practice any short-tailed distribution can be approximated via combinations of exponential distributions or phase-type distributions.

simulation. In [21] a promising centralised real-time protocol optimisation framework is presented, which uses deterministic formulas to infer the current network behaviour. Subsequently multi-objective programming is applied to find a better global parameter configuration for the network. Despite showing significant performance improvements in empirical tests, the framework currently cannot guarantee network improvements as protocol parameter changes alter the network state. Therefore the optimisation has to be run frequently to continually improve the network based on the latest performance measurements. Here, a fast prediction method could potentially reduce the optimisation frequency. In the following we briefly introduce the WSN hardware and software landscape. Subsequently, we formally present the PCTMC formalism and the mean-field method, which has the potential to provide a computationally efficient way of analysing large WSN models.

2.1 The WSN protocol stack

Nodes, also referred to as Motes, are small, embedded, battery powered radio devices with significant processing, bandwidth, radio and energy constraints [22]. The radio range heavily depends on the environment in which the network is deployed [23]. As for bandwidth, nodes such as the MicaZ can transmit up to 250 Kbps [22], although in many applications the actual throughput is much lower because of channel contention and other communication overheads. Similarly, as many types of nodes are battery powered, energy has to be used efficiently. In the literature the energy aspect has received the largest attention among all of these hardware related constraints. For deployments in which node batteries are hard to replace, application and middleware need to be tuned to increase network lifetime, i.e. the time until the WSN stops functioning due to energy depletion in one or more nodes. Since idle listening is the largest source of energy waste [24], the main method for reducing nodes' energy consumption is to introduce duty-cycling. When duty-cycling, nodes turn off their radio units whenever possible. If, over a time period T, a node has its radio turned on $x\%$ of the time, we say that the node has a duty-cycle of $x\%$. The lower x, the longer the network lifetime will be. Yet, while duty-cycling increases battery lifetime, it has a great impact on bandwidth, latency and reliability. To overcome the resulting QoS related challenges, a vast number of protocols have been suggested over the last decade [25], each of which aims to optimally balance different QoS aspects.

Figure 1 gives a high-level overview over the basic software architecture of wireless sensor applications. A more detailed representation can be found in [26]. The **Application layer** contains the logic required for data acquisition and processing. A simple application might measure quantities such as temperature, humidity or luminosity in regular intervals and forward the data to a sink node. Other applications might also process measured data, serve data requests or send messages in response to external events. Furthermore applications also need to decide which nodes to forward their data to. This can either be specific nodes or a high-level destinations such as data sinks. The **Network layer** [25] is responsible for ensuring that data from the application layer is routed towards

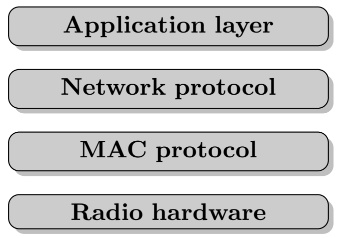

Fig. 1. A simple wireless sensor network protocol stack.

its destination. A common communication pattern is convergecast, where all nodes in the network sample information and forward the data to dedicated sink nodes via multi-hop routes. In multi-hop networks, routing protocols need to relay incoming packets from other nodes in addition to handling packets coming from their own application layer. Network protocols are either centralised or a decentralised. A centralised routing protocol elects one or several nodes which control the routing behaviour of the network, whereas decentralised protocols let nodes autonomously decide where to forward messages to. Protocols in the latter category are sometimes referred to as swarm intelligence or bio-inspired protocols [27]. **MAC layer** protocols on the other hand determine how neighbouring sensor nodes communicate with each other. There are three classes of MAC protocols, contention based protocols, schedule based protocols and hybrid approaches. In contention based protocols such as CSMA, nodes can send messages at any time provided the channel is clear, whereas in schedule based protocols like TDMA each node is allocated a time window during which it can transmit messages [28]. Additionally MAC protocols are in charge of managing the node's duty-cycle behaviour to ensure nodes are only awake when necessary. Finally the **Radio layer** controls nodes' radio hardware and can be used to configure signal modulation, frequency or transmission power.

Even though the vast protocol landscape provides solutions for nearly any kind of WSN application, building software for WSNs still requires experienced designers, who choose appropriate protocol setups that match QoS demands. To simplify the WSN application development process, researchers have come up with a variety of universal middleware frameworks [29,30,31,32] some of which are already capable of dynamically adapting their setup for performance gain [33]. Despite being suitable for particular application types, there is no guarantee that they will perform optimally in all scenarios. To balance the demand for application optimised WSN middleware and ease of application development, other researchers have proposed auto-generating bespoke middleware based on the application profile [34].

2.2 PCTMCs

Population models assume that a large number of identical individuals belonging to a particular population interact with individuals from other populations and thereby alter population levels. This abstraction from individuals to populations vastly reduces the complexity and the state-space of the underlying model. Common examples of population models are chemical reaction models [14,35] where populations represent molecule concentrations, ecology models [36] describing the behaviour of groups of animals or plants and software performance models [37,38] capturing the interactions between components in massively parallel systems.

Population continuous time Markov chains (PCTMCs) have a finite set of populations D, $n = |D|$ and a set E of transition classes. States are represented as an integer vector $\boldsymbol{p} = (p_1, \dots, p_n) \in \mathbb{Z}^n$, with the i^{th} component being the current population level of species $S_i \in D$. A transition class $(r_e, \boldsymbol{c}_e) \in E$ for an event e describes a transition with negatively exponentially distributed firing delay that occurs at rate $r_e : \mathbb{Z}^n \to \mathbb{R}$ and changes the population vector $\boldsymbol{p}$ into $\boldsymbol{p} + \boldsymbol{c}_e$. The analogue to PCTMCs in the systems biology literature are Chemical Reaction Systems, were $\boldsymbol{p}$ describes a molecule count vector and transition classes represent chemical reactions between the molecules with r_e being the reaction rate function and $\boldsymbol{c}_e$ the stoichiometric vector for a specific reaction. For notational convenience we write an event/reaction e as

$$\underbrace{S_* + \cdots + S_*}_{\text{in}} \to \underbrace{S_* + \cdots + S_*}_{\text{out}} \qquad\qquad \text{at } r_e(\boldsymbol{p}) \tag{1}$$

where $S_* \in D$ represent different species that are affected by the event. The corresponding change vector $\boldsymbol{c}_e = (s_1^{\text{out}} - s_1^{\text{in}}, \dots, s_n^{\text{out}} - s_n^{\text{in}}) \in \mathbb{Z}^n$ where s_i^{in} represents the number of occurrences of a species $S_i \in D$ on the left hand side of the event and s_i^{out} the number occurrences on the right hand side. The event rate is

$$\begin{cases} r_e(\boldsymbol{p}) & \text{if } p_i \geq s_i^{\text{in}} \text{ for all } i = 1, \dots, n \\ 0 & \text{otherwise} \end{cases}$$

An important aspect of PCTMC models is that approximations to the evolution of population moments of the underlying stochastic process can be represented by a system of ODEs [16]

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbb{E}[T(\boldsymbol{p}(t))] = \sum_{e \in E} \mathbb{E}[(T(\boldsymbol{p}(t) + \boldsymbol{c}_e) - T(\boldsymbol{p}(t)))\, r_e(\boldsymbol{p}(t))] \tag{2}$$

To obtain the ODE describing the evolution of the mean of a population p_i for instance, all we need to do is to substitute $T(\boldsymbol{P}) = P_i$ in the above equations, where P_i is the random variable representing the population count of species S_i. In the literature the resulting ODEs are often referred to as mean-field approximations [12,38]. Similarly ODEs for higher joint moments can be obtained by choosing adequate $T(\boldsymbol{P})$, e.g. $T(\boldsymbol{P}) = (P_i - \mu_i)^2$ for the variance of P_i. Alternatively stochastic simulation [35] can be used to evaluate PCTMCs. Like discrete event simulation for low-level protocol models, this latter simulation technique captures the stochastic behaviour of the PCTMC exactly, but it also does not scale for models with large populations.

When modelling spatially distributed networks such as WSNs, it is often easier to use a subclass of PCTMCs, so-called spatial PCTMCs (SPCTMCs). SPCTMCs have a discrete, finite number of locations each with a finite population of different agent states. By agent state population we mean the number of agents that are in a specific state of the underlying discrete state automata representing the agent, i.e. each agent description generates a number of species in the resulting PCTMC. When evaluating an SPCTMC we keep track of the evolutions of all agent state populations in all locations. The reason we distinguish between SPCTMCs and PCTMCs is that the population replication and the spatial notion of neighbourhoods can be exploited in order to simplify the higher-order moment ODE analysis [39]. A common way to design SPCTMC models is to use stochastic Petri nets or stochastic process algebras. The idea behind such high-level languages is to first describe local agent states, which can then be put together in a composite model. The composite model describes the topology, initial agent state populations in different locations and the interactions between neighbouring agent populations. For simplicity we refer to a species S at location l as $S@l$ [14,40]. Moreover, $S@l_*$ is used as a shorthand when defining events that occur in all locations in the same way.

3 PCTMC models of WSNs

When simulating WSN protocol stacks in network simulators like Castalia, TOSSIM or ns 2/3, each protocol is commonly represented as an individual module. While this works well in low-level modelling, we found that for PCTMC modelling it is easier to create models of cross-layer protocols which express the behaviour of the entire application. In the following we outline which features can be expressed in PCTMC models of WSNs and how this can be achieved.

3.1 WSN message exchange and buffers

In a PCTMC model of a WSN we assume that there are a discrete number of locations, each of which hosts one WSN node. Moreover, we assert that the radio range is fixed, so that every node has a set of neighbours that it can send messages to. Take for instance Figure 2, a simple topology with 15 nodes where each node has at most 4 neighbours. Even though this is an extremely regular topology, it is not hard to see that we can also express more sophisticated topologies with asymmetric links or varying neighbourhood densities in a similar way. Another important feature of a WSN node is its buffer. Generally a node's message buffer is small, allowing it to temporarily cache packets before forwarding them to other nodes in the network. Packet transmission is atomic in the sense that packets are either received entirely or not at all. Packet loss occurs due to channel interference, modulation errors and congestion control mechanisms. The biggest challenge in representing a buffer as a population of a PCTMC is to ensure that nodes send messages at a constant bandwidth until the buffer is empty. In the following we will explain how this can be done for synchronised unicast

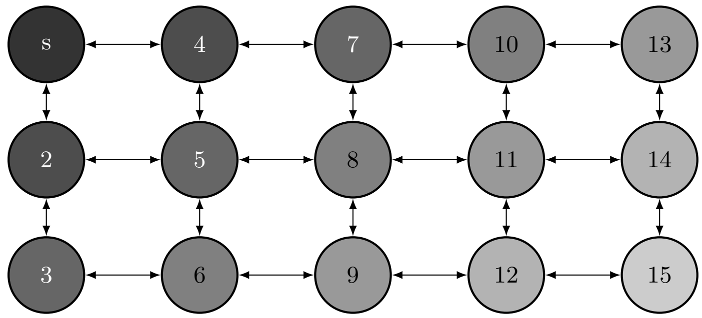

Fig. 2. Node 's' is the sink to which all other nodes route their messages. Any two nodes that are connected by arrows can communicate. In more realistic topologies some links may only be unidirectional since radio links can be asymmetric.

communication. Synchronised broadcast communication can be modelled similarly. Assuming interference free communication without packet loss, we obtain the following evolutions for a node at location l_1 that is sending unicast messages to nodes l_2 and l_3

$$\begin{aligned} \emptyset &\to \mathit{Buf}@l_1 && \text{at } \mathit{sampleRate} \\ \mathit{Buf}@l_1 &\to \mathit{Buf}@l_2 && \text{at } \mathit{bw}^* \mathit{TX}@l_1{}^* \mathit{RX}@l_2{}^* \mathbb{P}_{12} \\ \mathit{Buf}@l_1 &\to \mathit{Buf}@l_3 && \text{at } \mathit{bw}^* \mathit{TX}@l_1{}^* \mathit{RX}@l_3{}^* \mathbb{P}_{13} \end{aligned} \tag{3}$$

where $\mathit{Buf}@l_*$ is the buffer population at each location with $\mathit{Buf}@l_* = 0$ initially. The rate constants *sampleRate* and *bw* are the average number of sensor readings and the average number of packets that can be sent per time unit, respectively. Moreover, we assume that a single node is either in state RX or TX. Naturally communication can only happen if the sender is in TX and the receiver is in RX mode. The $\mathbb{P}_{ab}$ terms express the proportion of messages that node l_a sends to a node located at l_b such that $\sum_i \mathbb{P}_{ai} = 1$. To incorporate message loss we can add evolutions such as

$$\mathit{Buf}@l_1 \to \emptyset \qquad\qquad \text{at } \mathit{msgLossRate}$$

When analysing the evolutions shown in Eqn. (3) using mean-field techniques, the continuous buffer representation causes problems as it introduces indicator function terms to the ODEs. To overcome this problem we decided to use a discrete buffer representation instead. Assuming a single node has buffer states $\{\mathit{Buf}_0, \ldots, \mathit{Buf}_m\}$, where Buf_i represents the state in which the buffer contains i messages. The corresponding unicast communication reactions for lossless message transmission from l_1 to l_2 and l_3 are

$$\begin{array}{rl} Buf_i@l_1 \to Buf_{i+1}@l_1 & \text{at } \mathit{sampleRate} \\ Buf_j@l_1 + Buf_i@l_2 \to Buf_{j-1}@l_1 + Buf_{i+1}@l_2 & \text{at } bw^* \, TX@l_1{}^* \, RX@l_2{}^* \\ & \quad \mathbb{P}_{12}{}^* \, Buf_i@l_1{}^* \, Buf_j@l_2 \\ Buf_j@l_1 + Buf_i@l_3 \to Buf_{j-1}@l_1 + Buf_{i+1}@l_3 & \text{at } bw^* \, TX@l_1{}^* \, RX@l_3{}^* \\ & \quad \mathbb{P}_{13}{}^* \, Buf_i@l_1{}^* \, Buf_j@l_3 \end{array}$$

where initially $Buf_0@l_* = 1$, $0 \leq i < m$ and $0 < j \leq m$ and $(1 - Buf_0@l)$ is 1 whenever l has a non-empty buffer. In this case the transmission rate is always bw or 0 and the ODEs representing the evolution of the mean of populations $Buf_*@l_1$ and $Buf_*@l_2$ can be integrated everywhere. To simplify this model, in Section 4 we assert that nodes can always receive messages and attempt to send messages whenever their buffer is non-empty, i.e. we can ignore all $RX@l_*$ and $TX@l_*$ terms the above evolutions. In the mean-field ODEs the gradient for the expected buffer level $\mathbb{E}[Buf_i@l(t)]$ then becomes the sum of

$$\mathbb{E}[Buf_i@l(t)]^* \, bw^* \left(\sum_k \mathbb{P}_{kl}{}^* (1 - \mathbb{E}[Buf_0@k(t)]) \right) \tag{4}$$

and

$$-\mathbb{E}[Buf_i@l(t)]^* \, bw^* \left(\sum_k \mathbb{P}_{lk}{}^* (1 - \mathbb{E}[Buf_m@k(t)]) \right) \tag{5}$$

which represent the terms for the incoming and the outgoing messages respectively. To approximate the average buffer size for any location l at time t we then simply evaluate

$$\sum_{i=1}^{m} i^* \, \mathbb{E}[Buf_i@l(t)] \tag{6}$$

There are two drawbacks to the discrete buffer approach. Firstly, representing buffer levels as a discrete number of m populations creates m extra mean-field ODEs for every location. Secondly, when analysing the resulting mean-field ODEs, we have to bear in mind that the use of small populations ($\sum Buf_i@l = 1$) can lead to significant errors in the mean-field estimate of the real population means. Despite the latter shortcoming our example in Section 4 shows that this PCTMC buffer representation works qualitatively well, when comparing the mean-field solution to the results of a realistic low-level discrete event simulation of a WSN.

3.2 Network protocol

Having discussed how to represent basic WSN message exchange in PCTMC models, we now discuss how network protocols can be modelled. As mentioned in Section 2.1, there exist centralised and decentralised routing approaches. In our opinion the best way to create a PCTMC model of a centralised WSN routing protocol is to write an algorithm, which, given the network topology and the behaviour of the centralised routing protocol, generates the reactions outlined in Eqn. (3), with $\mathbb{P}_{ij}$ chosen to reflect the network topology. The network shown in Figure 3, for example, could have been generated according to a centralised routing algorithm executed at node 's'. Even though static routing models can be

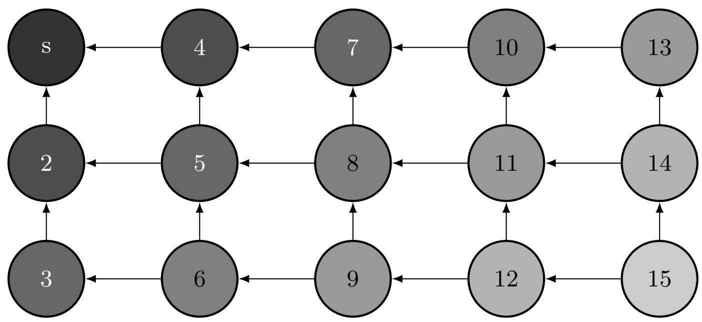

Fig. 3. Node 's' is the sink to which all other nodes route their messages.

analysed efficiently using mean-field methods, we are generally more interested in dynamic routing behaviour, for instance when studying fail-safe protocols [41]. We will now show how decentralised dynamic routing can be represented in a PCTMC model. Decentralised schemes require nodes to make decisions as to where they send messages to. To make informed decisions, nodes need to collect meta-information about their immediate neighbours, e.g. their buffer occupancy, link reliability, distance to the sink or battery status. This information can subsequently be used to compute $\mathbb{P}_{ij}$. In [18] the authors abstract such neighbourhood information as pheromone levels. From zoology, pheromone is a hormone used by foraging insects to mark routes between their nest and food sources. The higher the pheromone level along a certain path, the more insects will travel along that route. In models where peripheral nodes need to relay messages towards a sink node, e.g. Figure 3, we assume all nodes disseminate pheromone and that they infer routing decisions based on the resulting pheromone gradient. While Bruneo *et al.* [18] use discrete pheromone levels, represented in a manner similar to our buffer representation, we suggest a continuous pheromone level representation. A typical pheromone model for a convergecast network will assert that sink nodes are the pheromone sources, whereas all other nodes spread the pheromone emitted by the sinks. This way a pheromone gradient, represented by the shading in Figure 3, between sink and peripheral nodes emerges. As long as the pheromone level of nodes decreases with increasing hop distance from surrounding sink(s), we can easily use the resulting gradient to make local routing decisions that guarantee message delivery to the nearest sink(s). The reactions for the pheromone spread look as follows

$$\begin{aligned} \emptyset &\rightarrow Ph@l_* && \text{at } \mathit{pheroInc}@l_* \\ Ph@l_* &\rightarrow \emptyset && \text{at } \mathit{pheroDec}@l_* \end{aligned}$$

where $\mathit{pheroInc}@l_*$ is the sum of the difference between a node's pheromone level and that of its neighbours, e.g. at location 3

$$\mathit{pheroInc}@l_3 = \max(0, Ph@l_2 - Ph@l_3) + \max(0, Ph@l_6 - Ph@l_3) \tag{7}$$

and for sinks we assume that *pheroInc* is some constant. Moreover, let

$$\mathit{pheroDec}@l_* = \min(0.1, Ph@l_* - 2) \tag{8}$$

The min term ensures that the pheromone level will not fall below 0. Although the pheromone gradient presented here only encodes a node's distance from the sink, it is possible to incorporate other neighbourhood information such as buffer levels or battery status in the pheromone concentration in case further QoS constraints have to be met by the protocol. Having shown how to express continuous pheromone levels in a PCTMC model, we can further utilise these levels to make dynamic routing decisions. A straightforward way of doing this is illustrated by the following reactions for the node at location 3 in Figure 3.

$$\begin{aligned} Buf_i@l_1 &\rightarrow Buf_{i+1}@l_1 && \text{at } \mathit{sampleRate} \\ Buf_j@l_3 + Buf_i@l_2 &\rightarrow Buf_{j-1}@l_3 + Buf_{i+1}@l_2 && \text{at } \mathit{RouteUp}@l_3{}^* Buf_j@l_3{}^* \\ &&& \quad bw^* Buf_i@l_2 \\ Buf_j@l_3 + Buf_i@l_6 &\rightarrow Buf_{j-1}@l_3 + Buf_{i+1}@l_6 && \text{at } \mathit{RouteRight}@l_3{}^* Buf_j@l_3{}^* \\ &&& \quad bw^* Buf_i@l_6 \\ Buf_j@l_2 + Buf_i@l_3 &\rightarrow Buf_{j-1}@l_2 + Buf_{i+1}@l_3 && \text{at } \mathit{RouteDown}@l_2{}^* Buf_j@l_2{}^* \\ &&& \quad bw^* Buf_i@l_3 \\ Buf_j@l_6 + Buf_i@l_3 &\rightarrow Buf_{j-1}@l_6 + Buf_{i+1}@l_3 && \text{at } \mathit{RouteLeft}@l_6{}^* Buf_j@l_6{}^* \\ &&& \quad bw^* Buf_i@l_3 \end{aligned} \tag{9}$$

In contrast to the static centralised approached, where we assumed fixed routing probabilities for all neighbours, we now have $\mathit{RouteUp}@l_*$, $\mathit{RouteDown}@l_*$, $\mathit{RouteLeft}@l_*$ and $\mathit{RouteRight}@l_*$ instead of $\mathbb{P}_{ij}$. We can express $\mathit{RouteLeft}@l_6$ as

$$\mathit{RouteLeft}@l_6 = \frac{max(0, Ph@l_3 - Ph@l_6)}{\mathit{pheroInc}@l_6} \tag{10}$$

i.e. the pheromone excess of node 3 over 6 divided by the sum of the excesses of all neighbours of node 6. Clearly, if location 3 has a lower pheromone level than location 6, node 6 will not route messages via node 3. If node 3 has a higher pheromone level, a proportion of messages from node 6 is relayed to the sink via location 3. Sink nodes have to be handled separately as they would have 0 denominators. However, fractions of populations are undesirable in moment approximating ODEs as they cause significant loss of accuracy for small denominators, which can cause errors when approximating higher-order moments. A suitable alternative that works better for mean-field analysis can be obtained by treating the routing probabilities as populations

$$\begin{aligned}
\mathit{RouteUp}@l_6 &\to \mathit{RouteLeft}@l_6 && \text{at } \mathit{max}(0, \mathit{Ph}@l_3 - \mathit{Ph}@l_6)^* \mathit{RouteUp}@l_6 \\
\mathit{RouteDown}@l_6 &\to \mathit{RouteLeft}@l_6 && \text{at } \mathit{max}(0, \mathit{Ph}@l_3 - \mathit{Ph}@l_6)^* \mathit{RouteDown}@l_6 \\
\mathit{RouteRight}@l_6 &\to \mathit{RouteLeft}@l_6 && \text{at } \mathit{max}(0, \mathit{Ph}@l_3 - \mathit{Ph}@l_6)^* \mathit{RouteRight}@l_6
\end{aligned}$$

Reactions for $\mathit{RouteUp}@l_*$, $\mathit{RouteDown}@l_*$ and $\mathit{RouteRight}@l_*$ follow a similar pattern. If we ensure that $\mathit{RouteUp}@l_* + \mathit{RouteDown}@l_* + \mathit{RouteLeft}@l_* + \mathit{RouteRight}@l_* = 1$ then this will yield routing populations that have similar steady state behaviour as Eqn. (10).

3.3 MAC protocol

While we argue that it is possible to model and evaluate non-trivial routing protocols using PCTMCs and mean-field analysis, it is much harder to represent sophisticated MAC protocols using the PCTMC formalism. In [17] Gribaudo *et al.* show that PCTMCs can represent duty-cycled MAC protocols with sender initiated transfers, i.e. protocols where nodes that want to propagate a message stay awake until the receiving node wakes up. Protocols like S-MAC, however, which require nodes to wake up in regular, synchronised intervals are hard to represent using PCTMCs. This is because feasible phase-type approximations cannot accurately represent deterministic or even near deterministic delays. Whether a MAC protocol can be represented by a PCTMC thus depends on how deterministic the cycles are.

3.4 Physical layer

One of the most challenging aspects of WSN modelling is to capture the behaviour of the wireless medium [42]. The two most important factors are the natural variation in signal strength and packet collisions. Despite the use of log-normal shadowing for path loss and sophisticated collision models that simulate capture effects,[2] even simulators such as Castalia do not manage to replicate the exact behaviour of empirical networks [42]. In this light it is unrealistic to expect PCTMC models to capture the characteristics of the wireless medium with high quantitative accuracy. Nevertheless, it is worth aiming at obtaining qualitative agreement, especially when using PCTMC models for protocol optimisation. Thus far, however, attempts to recreate the effects of radio interference in our PCTMC models have only been moderately successful and are subject to further research.

[2] When considering capture effects, collisions only occur if interfering signals are sufficiently strong.

3.5 Other limitations and opportunities

Many publications on WSN protocols deal with the prediction and optimisation of energy consumption in WSNs. As we mentioned earlier, common energy saving features such as duty-cycling are generally hard to express in PCTMC models. Similarly, the evolution of battery levels is difficult to represent in models since batteries discharge in a highly non-linear fashion [43,44]. Regardless of these restrictions, PCTMC models can be used to analyse the dataflow of messages under static and dynamic routing conditions (cf. Section 4). Insights into the dataflow of a WSN application can be used to estimate the energy consumption. Generally, a more evenly distributed message load in the network will equate to better energy durability in individual nodes.

A final aspect of WSNs is node mobility. While it is straightforward to model node failure in particular locations, thus far we have not found a strategy for representing mobile nodes in PCTMC models. Nevertheless it might be possible to port some of the concepts developed for gossip models [45] and epidemics [46], to represent roving nodes.

4 Worked example

In Section 3 we described how a PCTMC can be used to model unicast communication in a WSN with decentralised dynamic routing. We now illustrate that the mean-field analysis for our PCTMC abstraction of such a WSN can indeed produce a good qualitative representation of the dataflow behaviour in a WSN, even in presence of light interference. The comparison shown in Table 1 was taken from [47]. It compares our mean-field results for the pheromone model discussed in Section 3 to the average obtained from 200 Castalia simulations of the same WSN. Note that Castalia is a low-level network simulator, which simulates the exchange of every message separately. Since Castalia simulations use a sophisticated collision model, we also added a simple collision model to our PCTMC model. The modified PCTMC model makes receivers discard messages when two or more neighbours send messages simultaneously. The heat map in Table 1 shows the normalised buffer sizes in a network with 100 nodes, where every node produces 1 message per second and can relay up to 20 messages per second. To normalise these mean buffer levels at steady state, we set the non-sink node with the highest buffer to 100%. The resulting spatial heat maps represent relative buffer levels. To create a strong contrast, all nodes with a relative buffer size between $x\%$ to $(x-5)\%$ are coloured black at $x\%$ opacity, and sink nodes are coloured at 100% opacity. Sinks are marked 's' and broken nodes are left white and are marked 'x'.

5 Conclusions and future work

We have illustrated how PCTMC models can be used to represent various aspects of WSN protocol stacks. Even though it is undoubtedly true that realistic

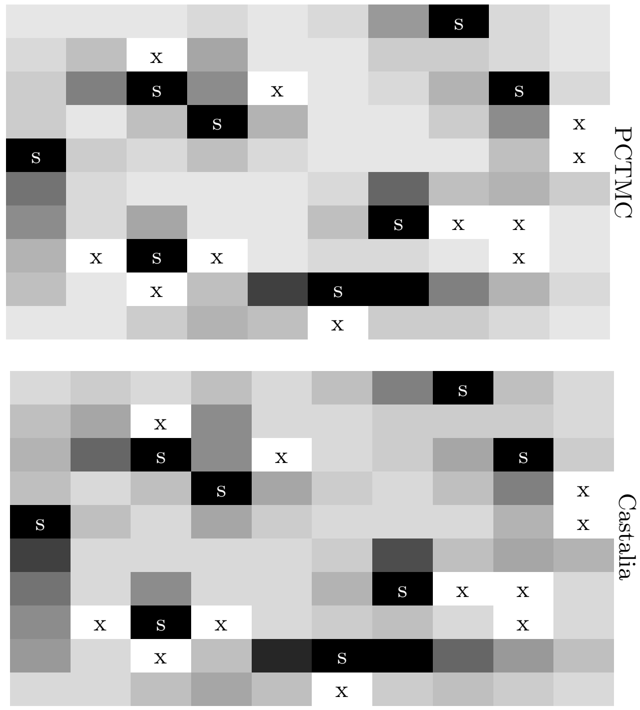

Table 1. Data flow in a network with 100 nodes several sinks and broken nodes in presence of interference [47]. Hotspot regions have darker shades, sink locations are marked ‘s’, broken nodes are marked ‘x’.

low-level simulations will remain predominant in the WSN community, our aim is to establish mean-field analysis techniques as a rapid heuristic that can be used to focus computationally expensive low-level simulations. Analysing the routing behaviour and the dataflow of nodes running decentralised routing protocols is one such example, but we aim to provide more case studies in the future. Current limitations for our PCTMC models are the lack of techniques to express interference, synchronous duty-cycle behaviour and mobility. Clearly these features deserve further attention, as they are key concepts whose implementation would make PCTMC modelling more appealing to the WSN community. In case PCTMC models are not capable of capturing all of these aspects, mean-field evaluation techniques for Generalised Semi-Markov Processes (GSMPs) [16] are worth investigating too. Aside from mean-field approaches we also intend to consider hybrid modelling paradigms such as HYPE [48]. Moreover, in the future we plan to perform formal benchmarks in order to compare the speed and accuracy of realistic low-level network simulations of WSN protocols with analysis techniques for abstract WSN performance models.

References

1. F. Viani, G. Oliveri, M. Donelli, L. Lizzi, P. Rocca, and A. Massa, “WSN-based Solutions for Security and Surveillance,” *Computer*, no. September, pp. 1762–1765, 2010.
2. M. S. Al-Fares and Z. Sun, “Self-Organizing Routing Protocol to achieve QoS in Wireless Sensor Network for Forest Fire Monitoring,” *Systems Research*, pp. 211–216, 2009.
3. N. Xu, S. Rangwala, K. K. Chintalapudi, D. Ganesan, A. Broad, R. Govindan, and D. Estrin, “A wireless sensor network For structural monitoring,” *Proceedings of the 2nd international conference on Embedded networked sensor systems SenSys 04*, vol. 20, no. 7, pp. 13–24, 2004.
4. M. R. Akhondi, A. Talevski, S. Carlsen, and S. Petersen, “Applications of Wireless Sensor Networks in the Oil, Gas and Resources Industries,” *2010 24th IEEE International Conference on Advanced Information Networking and Applications*, pp. 941–948, 2010.
5. R. Bagree, V. R. Jain, A. Kumar, and P. Ranjan, “TigerCENSE: wireless image sensor network to monitor tiger movement,” in *REALWSN10 Proceedings of the 4th international conference on Realworld wireless sensor networks*, REALWSN’10, pp. 13–24, Springer-Verlag, 2010.
6. J. Micallef, I. Grech, A. Brincat, V. Traver, and E. Monto, “Body area network for wireless patient monitoring,” *IET Communications*, vol. 2, no. 2, pp. 215–222, 2008.
7. A. Boulis, “Castalia: revealing pitfalls in designing distributed algorithms in WSN,” in *Proceedings of the 5th international conference on Embedded networked sensor systems* (S. Jha, ed.), pp. 407–408, ACM, 2007.
8. Ns, “The Network Simulator - ns-2,” 2002.
9. P. Levis and N. Lee, “TOSSIM : A Simulator for TinyOS Networks,” *UC Berkeley September*, pp. 1–17, 2003.
10. L. Bergamini, C. Crociani, A. Vitaletti, and M. Nati, “Validation of WSN simulators through a comparison with a real testbed,” in *Proceedings of the 7th ACM workshop on Performance evaluation of wireless ad hoc sensor and ubiquitous networks*, pp. 103–104, ACM, 2010.
11. E. Egea-Lopez, J. Vales-Alonso, A. Martinez-Sala, P. Pavon-Mario, and J. Garcia-Haro, “Simulation Scalability Issues in Wireless Sensor Networks,” *IEEE Communications Magazine*, vol. 44, no. 7, pp. 64–73, 2006.
12. M. Opper and D. Saad, *Advanced Mean Field Methods: Theory and Practice*. The MIT Press, 2001.
13. N. G. Van Kampen, *Stochastic Processes in Physics and Chemistry*, vol. 11 of *North-Holland personal library*. North-Holland, 1992.
14. F. Ciocchetta and J. Hillston, “Bio-PEPA: A framework for the modelling and analysis of biological systems,” *Theoretical Computer Science*, vol. 410, no. 33-34, pp. 3065–3084, 2009.
15. J. Hillston, “Fluid flow approximation of PEPA models,” *Second International Conference on the Quantitative Evaluation of Systems QEST05*, pp. 33–42, 2005.
16. R. A. Hayden, “Mean-field approximations for performance models with generally-timed transitions,” *Accepted for publication in ACM SIGMETRICS Performance Evaluation Review*, 2011.

17. M. Gribaudo, D. Cerotti, and A. Bobbio, “Analysis of On-off policies in Sensor Networks Using Interacting Markovian Agents,” *6th IEEE International Conference on Pervasive Computing and Communications PerCom (2008)*, pp. 300–305, 2008.
18. D. Bruneo, M. Scarpa, A. Bobbio, D. Cerotti, and M. Gribaudo, “Markovian agent modeling swarm intelligence algorithms in wireless sensor networks,” *Performance Evaluation*, vol. In Press,, 2011.
19. A. Förster and A. L. Murphy, “A Critical Survey and Guide to Evaluating WSN Routing Protocols,” in *The First International Workshop on Networks of Cooperating Objects (CONET)*, (Stockholm), 2010.
20. G. Anastasi, M. Conti, and M. D. Francesco, “A Comprehensive Analysis of the MAC Unreliability Problem in IEEE 802 . 15 . 4 Wireless Sensor Networks,” *Ieee Transactions On Industrial Informatics*, vol. 7, no. 1, pp. 52–65, 2011.
21. M. Zimmerling, F. Ferrari, L. Mottola, T. Voigt, and L. Thiele, “pTunes: Runtime Parameter Adaptation for Low-power MAC Protocols,” in *Proceedings of the 11th international conference on Information Processing in Sensor Networks - IPSN ’12*, (New York, New York, USA), p. 173, ACM Press, 2012.
22. “Crossbow: Crossbow datasheet on MicaZ,” 2006.
23. K. Sohrabi, B. Manriquez, and G. J. Pottie, “Near ground wideband channel measurement in 800-1000 MHz,” *1999 IEEE 49th Vehicular Technology Conference Cat No99CH36363*, vol. 1, no. 3, pp. 571–574, 1999.
24. T. Van Dam and K. Langendoen, “An adaptive energy-efficient MAC protocol for wireless sensor networks,” in *Proceedings of the first international conference on Embedded networked sensor systems SenSys 03* (I. F. Akyildiz, D. Estrin, D. E. Culler, and M. B. Srivastava, eds.), vol. 03pages of *SenSys ’03*, p. 171, ACM Press, 2003.
25. J. Yick, B. Mukherjee, and D. Ghosal, “Wireless sensor network survey,” *Computer Networks*, vol. 52, no. 12, pp. 2292–2330, 2008.
26. P. Yadav, *Cross-Layer Protocols to Support Periodic Data Collection and Event Driven Wireless Sensor Network Applications.* Phd thesis, Imperial College, 2011.
27. M. Paone, L. Paladina, D. Bruneo, and A. Puliafito, “A Swarm-based Routing Protocol for Wireless Sensor Networks,” *Sixth IEEE International Symposium on Network Computing and Applications NCA 2007*, vol. 31, no. Nca, pp. 265–268, 2007.
28. A. Bachir, M. Dohler, T. Watteyne, and K. K. Leung, “MAC Essentials for Wireless Sensor Networks,” *IEEE Communications Surveys Tutorials*, vol. 12, no. 2, pp. 222–248, 2010.
29. K. Römer, O. Kasten, and F. Mattern, “Middleware challenges for wireless sensor networks,” *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 6, no. 4, pp. 59–61, 2002.
30. I. Chatzigiannakis, G. Mylonas, and S. Nikoletseas, “50 ways to build your application: A survey of middleware and systems for Wireless Sensor Networks,” *2007 IEEE Conference on Emerging Technologies Factory Automation EFTA 2007*, pp. 466–473, 2007.
31. S. Tong, “An Evaluation Framework for middleware approaches on Wireless Sensor Networks,” tech. rep., Helsinki University of Technology, Helsinki, 2007.
32. M.-M. Wang, J.-N. Cao, J. Li, and S. K. Dasi, “Middleware for Wireless Sensor Networks: A Survey,” *Journal of Computer Science and Technology*, vol. 23, no. 3, pp. 305–326, 2008.

33. T. Liu and M. Martonosi, "Impala: a middleware system for managing autonomic, parallel sensor systems," in *System*, vol. 38 of *PPoPP '03*, pp. 107–118, ACM, 2003.
34. C. Buckl, S. Sommer, A. Scholz, A. Knoll, and A. Kemper, "Generating a Tailored Middleware for Wireless Sensor Network Applications," *Sensor Networks Ubiquitous and Trustworthy Computing 2008 SUTC 08 IEEE International Conference on*, pp. 162–169, 2008.
35. D. T. Gillespie, "Exact stochastic simulation of coupled chemical reactions," *Journal of Physical Chemistry*, vol. 81, no. 25, pp. 2340–2361, 1977.
36. P. J. Wangersky, "Lotka-Volterra population models," *Annual Review of Ecology and Systematics*, vol. 9, no. 1, pp. 189–218, 1978.
37. A. Stefanek, R. A. Hayden, and J. T. Bradley, "Fluid computation of the performance-energy trade-off in large scale Markov models," *Accepted for publication in ACM SIGMETRICS Performance Evaluation Review*, 2011.
38. M. Benaim and J. Le Boudec, "A class of mean field interaction models for computer and communication systems," *Performance Evaluation*, vol. 65, no. 11-12, pp. 823–838, 2008.
39. A. Stefanek, M. C. Guenther, and J. T. Bradley, "Normal and inhomogeneous moment closures for stochastic process algebras," in *10th Workshop on Process Algebra and Stochastically Timed Activities (PASTA'11)*, (Ragusa), 2011.
40. V. Galpin, "Towards a spatial stochastic process algebra," in *Proceedings of the 7th Workshop on Process Algebra and Stochastically Timed Activities (PASTA)*, (Edinburgh), 2008.
41. A.-F. Liu, M. Ma, Z.-G. Chen, and W.-H. Gui, "Energy-Hole Avoidance Routing Algorithm for WSN," in *Natural Computation 2008 ICNC 08 Fourth International Conference on*, vol. 1, pp. 76—-80, 2008.
42. G. P. Halkes and K. G. Langendoen, "Experimental Evaluation of Simulation Abstractions for Wireless Sensor Network MAC Protocols," *EURASIP Journal on Wireless Communications and Networking*, vol. 2010, pp. 1–10, 2010.
43. E. Perla, A. O. Catháin, R. S. Carbajo, M. Huggard, and C. Mc Goldrick, "PowerTOSSIM z : Realistic Energy Modelling for Wireless Sensor Network Environments," in *Proceedings of the 3nd ACM workshop on Performance monitoring and measurement of heterogeneous wireless and wired networks*, pp. 35–42, ACM, 2008.
44. M. R. Jongerden and B. R. Haverkort, "Which battery model to use?," *IET Software*, vol. 3, no. 6, p. 445, 2009.
45. A. Chaintreau, J. Y. Le Boudec, and N. Ristanovic, "The age of gossip: spatial mean field regime," *Evolution*, pp. 109–120, 2009.
46. G. Caravagna and J. Hillston, "Modeling biological systems with delays in Bio-PEPA," *Electronic Proceedings in Theoretical Computer Science*, vol. 40, no. MeCBIC, pp. 85–101, 2010.
47. M. C. Guenther and J. T. Bradley, "Mean-field analysis of data flows in Wireless Sensor Networks," in *Submitted to VALUETOOLS'12 (http://www.doc.ic.ac.uk/˜mcg05/wsnrouting)*, 2012.
48. V. Galpin, L. Bortolussi, and J. Hillston, "HYPE: a process algebra for compositional flows and emergent behaviour," in *CONCUR 2009Concurrency Theory*, no. September, pp. 305–320, Springer, 2009.