Mean-field Analysis of Data Flows in Wireless Sensor Networks

Marcel C. Guenther
Imperial College London
180 Queen's Gate
London SW7 2RH, United Kingdom
mcg05@doc.ic.ac.uk

Jeremy T. Bradley
Imperial College London
180 Queen's Gate
London SW7 2RH, United Kingdom
jb@doc.ic.ac.uk

ABSTRACT

Wireless Sensor Networks (WSNs) are often used for environment monitoring, an application which requires reliable routing of messages from source to sink nodes via multi-hop networks. Prior to installing such WSNs, engineers commonly analyse the network using discrete event simulation (DES). Whilst sophisticated simulators such as Castalia and TOSSIM take into account many low-level features of WSNs, their biggest drawback is the lack of scalability. This inhibits design-time system optimisation for large or complex networks. In this paper, we discuss how Population CTMC (PCTMC) models, used in conjunction with mean-field analysis, can be used to mitigate this problem. To illustrate the potential of PCTMC models in the WSN domain, we present a PCTMC model for a failsafe, dynamic routing protocol, which we implemented in Castalia. We show that the mean-field solution for the model yields good qualitative agreement with corresponding low-level simulations, but at a fraction of the computational cost. In particular we see good agreement for average metrics describing buffer occupancy and data flow behaviour. Moreover, our PCTMC model produces good results when packets are lost due to channel interference, an important consideration for WSNs.

Categories and Subject Descriptors

C.4 [**Performance of systems**]: Modeling techniques

General Terms

Application of modelling formalisms

Keywords

PCTMC, Mean-field analysis, Spatial modelling

1. INTRODUCTION

With the increasing availability of inexpensive wireless sensor hardware, the number of application areas for wireless sensor networks (WSNs) has continually grown over the last couple of years. One such application is environment monitoring, an application in which WSN nodes periodically capture local information such as temperature, humidity or luminosity and forward the information to a sink node via a multi-hop *ad hoc* network. In addition to periodical samples, some of these applications further allow event notification. This is beneficial, for instance, in the case of forest fire being detected by an environment monitoring WSN or where an intruder has entered an area surveilled by a security monitoring WSN. In the literature there are numerous examples of WSNs for different applications such as fire detection [3], landslide detection [25] and agricultural area monitoring [30] as well as structural monitoring and controlling [32, 2].

While all of the above examples use periodic and event based communication patterns, the requirements in terms of energy efficiency, reliability or throughput vary considerably between different deployments. An outdoor application for instance usually requires better energy efficiency as battery replacement is more expensive in locations that are difficult to access. An indoor intrusion detection system on the other hand may require low latency and high throughput to transmit pictures of an intruder. As a consequence, many MAC and network protocols have been developed for WSNs over the last decade in order to cater for a large variety of Quality of Service (QoS) demands [4]. However, it is difficult to decide which protocol to use for a given application, as the performance depends on the WSN topology and the configuration of the protocol stack. A recent study [10] came to the conclusion that protocol comparison literature generally provides insufficient information regarding experimental setups. Moreover, the study claims that too many protocol comparisons do not adequately tune protocols for their test environment. Being left with a range of candidate protocols for a specific application, engineers need to test a large number of different protocol stacks with different parameter configurations in order to optimise their WSN before deployment.

The most widely used low-level WSN simulators for this task are TOSSIM [21], Castalia [6] (an extension to the Omnet++ [20] simulation environment) and ns-2/3 [22]. These tools provide state-of-the-art models for simulating wireless networks, enabling engineers to test complete protocol stacks under fairly realistic conditions [5]. The major downside of these discrete event simulation (DES) frameworks, however, is the lack of scalability, so that the analysis of large wireless networks with hundreds of nodes becomes computationally expensive [8]. This makes it virtually impossible to test a large range of protocol stacks and con-

ICPE'13, March 21–24, 2013, Prague, Czech Republic.

figurations in an effort to optimise the software for an application. Population CTMC (PCTMC) analysis methods potentially provide an alternative to DES when investigating particular aspects of WSNs. PCTMCs are continuous time Markov chains that keep track of the evolution of populations over time. Their main advantage is that the population moments of the underlying stochastic process, such as mean and variance of the populations, can be approximated by ordinary differential equations (ODEs) [18, 19]. This efficient form of analysis, which is also referred to as mean-field analysis [23], has made PCTMCs a popular paradigm for modelling large client-server systems. Naturally one of the major constraints of PCTMCs is that the delay between any two events must be negatively exponentially distributed or alternatively belong to a distribution which can be approximated by a phase-type distribution. In the past this may have deterred some engineers from using PCTMC models to represent WSN protocols, which often exhibit deterministically timed behaviour. As a consequence PCTMC-style models for large WSNs are hardly covered in the literature[1]. Nevertheless, a promising example of a PCTMC-style model was given in [7], where the authors showed how mean-field analysis can be used to investigate the swarm intelligence based network reorganisation of a failsafe network originally described in [24]. Another example was the use of PCTMC-style models to investigate different sleep policies for WSN nodes [13]. In [16] we reviewed existing PCTMC WSN modelling techniques, presented the simple interference-free PCTMC WSN model that is reviewed in Section 3.2 and discussed open challenges. Moreover, in [15] we showed that for the resulting models, recently developed mean-field analysis tools are capable of efficiently approximating higher-order statistics for various WSN specific metrics. This is particularly useful when analysing transient behaviour of WSNs during network topology changes.

Though self-contained, this paper builds upon the work presented in [15, 16]. The contributions of this paper are as follows. In Section 3.3 this paper addresses the missing representation of channel interference in the simple PCTMC model described in [16] (cf. Section 3.2). This may appear to be a mere incremental improvement, however, it is generally a non-trivial problem to incorporate and analyse interference in mathematical models with complex topologies. Moreover, interference representation is an essential feature that is required if we aim to promote PCTMC modelling as a tool in the engineering community. In addition to this, we show that mean-field analysis is a tractable, fast solution technique for PCTMC WSN models even when simultaneously considering dynamic routing, message exchange and interference effects. Finally, in contrast to [15, 16] this paper compares the steady-state mean-field results of our PCTMC models with the simulation results obtained from a low-level Castalia protocol stack implementation and shows that the PCTMC model can be seen as a limiting case when message sampling and forwarding rates in the network become large.

Our paper is organised as follows; in Section 2 we provide background on PCTMC and spatial PCTMC modelling and evaluation techniques. In particular we introduce the so-called mean-field ODE analysis technique [23]. Section 3 subsequently describes a Castalia and a PCTMC model implementation of a data gathering WSN with dynamic routing. In Section 4 we then compare the data flow results from the Castalia simulation and the PCTMC ODE analysis for different topologies and communication constraints. In particular we compare our model to the Castalia simulation under both ideal and more realistic wireless channels. Finally, Section 5 concludes our work and suggests further research opportunities.

[1]By PCTMC-style we mean models that have been expressed in any formalism for which a lossless mapping to PCTMCs exists.

2. BACKGROUND

In this section we first introduce the notion of population modelling, which has inspired the development of population CTMCs (PCTMCs). Subsequently, we formally review PCTMCs and the mean-field moment approximation method for PCTMCs. We then introduce some extra notation for PCTMCs to simplify the representation of spatially distributed populations, which will later be used for describing our WSN model. Thereafter, we look at basic WSN concepts and terminology.

2.1 PCTMCs

In population models it is assumed that a large number of identical individuals belonging to a particular population interact with individuals from other populations. While we generally assert that individuals within a population have the same behaviour, this does not prevent populations of different individuals from interacting with one another. Interactions either result in a proportion of individuals changing their state locally or in births and deaths of individuals. The abstraction from individuals to populations can help to reduce the complexity and the state-space of the underlying model. Moreover, in the case of CTMCs, PCTMCs arise from a lumpable state-space, which can be efficiently analysed using mean-field techniques discussed below. Common examples of population models are chemical reaction models [12], where populations represent molecule concentrations, ecology models [31] describing the behaviour of groups of animals or plants and software performance models [29] capturing the interactions between components in massively parallel systems.

Population Continuous-Time Markov Chains (PCTMCs) consist of a finite set of populations S, $n = |S|$ and a set E of transition classes. States are represented as an integer vector $\vec{P}(t) = (P_1(t), \ldots, P_n(t)) \in \mathbb{Z}^n$, with the i^{th} component being the current population level of species $S_i \in S$ at time t. A transition class $(r_e, \vec{c}_e) \in E$ for an event e describes a transition with negatively exponentially distributed delay D at rate $r_e : \mathbb{Z}^n \to \mathbb{R}$ which changes the population vector $\vec{P}(t+D)$ to $\vec{P}(t) + \vec{c}_e$. The analogue to PCTMCs in the systems biology literature are Chemical Reaction Systems, were $\vec{P}(t)$ describes a molecule count vector and transition classes represent chemical reactions between the molecules with r_e being the reaction rate function and $\vec{c}_e$ the stoichiometric vector for a specific reaction. For notational convenience we write an event/reaction e as

$$\underbrace{S_* + \cdots + S_*}_{\text{in}} \to \underbrace{S_* + \cdots + S_*}_{\text{out}} \qquad \text{at } r_e(\vec{P}(t)) \quad (1)$$

where $S_* \in S$ represent different species that are involved in the event. The corresponding change vector is $\vec{c}_e = (s_1^{\text{out}} - s_1^{\text{in}}, \ldots, s_n^{\text{out}} - s_n^{\text{in}}) \in \mathbb{Z}^n$ where s_i^{in} represents the number of occurrences of a particular species $S_i \in S$ on the left hand

side of the event and s_i^{out} its number of occurrences on the right hand side. The event rate is

$$\begin{cases} r_e(\vec{P}(t)) & \text{if } P_i(t) \geq s_i^{\text{in}} \text{ for all } i = 1, \ldots, n \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

An important feature of PCTMC models is that approximations to the evolution of population moments of the underlying stochastic process can be represented by the following system of ODEs [18]

$$\frac{\mathrm{d}}{\mathrm{d}t}\mathbb{E}[T(\vec{P}(t))] = \sum_{e \in E} \mathbb{E}[(T(\vec{P}(t)+\vec{c}_e) - T(\vec{P}(t)))r_e(\vec{P}(t))] \quad (3)$$

To obtain the ODE describing the evolution of the mean of a population, all we need to do is to substitute $T(\vec{P}(t)) = P_i(t)$ in the above equation, where $P_i(t)$ is the random variable representing the population count of species S_i at time t. In the literature the resulting ODEs are often referred to as mean-field approximations [23]. Similarly ODEs for higher joint moments can be obtained by choosing adequate $T(\vec{P}(t))$, e.g. $T(\vec{P}(t)) = (P_i(t) - \mu_i)^2$ for the variance of $P_i(t)$. Alternatively stochastic simulation [12] can be used to evaluate PCTMCs. Like discrete event simulation for low-level protocol models, this latter simulation technique captures the stochastic behaviour of PCTMCs exactly, but does not scale for models with larger populations.

2.2 Spatial PCTMCs

When modelling spatially distributed networks, it is often easier to use Spatial PCTMCs (SPCTMCs). SPCTMCs have a discrete, finite number of locations each with a finite number of population for different agent states. When evaluating an SPCTMC we keep track of the evolutions of all agent state populations in all locations. Aside from helping modellers to describe spatial processes more easily, the extra spatial information can occasionally be used to speed-up higher-order moment ODE analysis [27]. When describing SPCTMCs we usually first describe local agent behaviour. Later a composite model which describes topology, initial agent state populations for all locations and interactions between neighbouring populations is created. For the remainder of this paper we refer to a species S at location l at time t as $S@l(t)$ [11]. Moreover, $S@l_*(t)$ is used as a placeholder for defining events that occur in all locations.

2.3 Wireless sensor networks

Nodes in WSNs are small, embedded, battery powered devices with significant processing, bandwidth, radio and energy constraints [1]. The radio range heavily depends on the environment in which the network is deployed [26]. As for bandwidth, nodes such as the MicaZ can transmit up to 250 Kbps [1], although in many applications the actual throughput is much lower because of channel contention and other communication overheads. Since many types of nodes are battery powered, energy has to be used efficiently. To overcome these QoS related challenges a vast number of protocols have been suggested over the last decade [34], each of which aims to optimally balance different QoS constraints.

Figure 1 gives a high-level overview over the basic software architecture of wireless sensor applications. A more detailed representation can be found in [33]. The **Application layer** contains the logic required for data acquisition and data 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. Moreover, applications 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** [34] is responsible for ensuring that data from the application layer is routed towards 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 therefore 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 decentralised. A centralised routing protocol elects one or several nodes which control the routing behaviour of the network, whereas decentralised protocols let nodes decide autonomously where to forward messages to. Protocols in the latter category are sometimes referred to as swarm intelligence based protocols [24]. **MAC layer** protocols on the other hand, control the communication with actual neighbouring nodes. Additionally, MAC-protocols are in charge of managing the node's duty-cycle behaviour. When duty-cycled, nodes turn off their radio units according to a sleep policy. As the name suggests this happens in a cyclic manner such that over a period T a node has its radio turned on $x\%$ of the time. The lower x, the longer the network lifetime will be. There are two classes of MAC protocols, contention based protocols and schedule based protocols. In contention based protocols such as CSMA (*Carrier Sense Multiple Access*), nodes can send messages at any time provided the channel is clear. On the other hand schedule based protocols like TDMA (*Time Division Multiple Access*) allocate a specific time window to each node, during which it can transmit messages [4]. Moreover, many hybrid protocols, which use elements of both contention and schedule based MAC protocols, have been suggested. Finally the **Radio layer** controls nodes' radio hardware and can be used to configure signal modulation, frequency or transmission power.

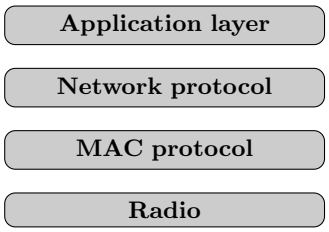


Figure 1: A simple wireless sensor network protocol stack [16].

3. A DYNAMIC WSN DATA FLOW MODEL

In the following we outline a failsafe WSN routing protocol, i.e. a protocol that can automatically detect failing nodes and dynamically close gaps in multi-hop routes, similar to the ones described in [15, 16]. Our analysis focuses on the data flow from source to sink nodes in different topologies. Figure 2 shows a simple example of a WSN. Each node has up to four neighbours and communication is done in a

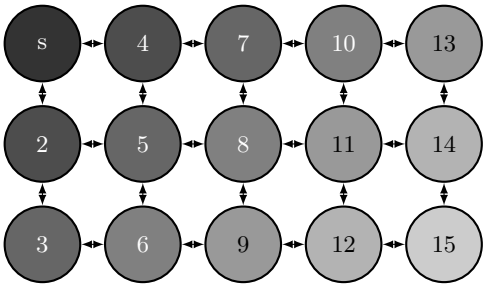


Figure 2: Node 's' is the sink and has the highest pheromone and buffer level. Both levels decrease with increasing hop distance from the sink [16].

Manhattan style pattern, i.e. we assume a unit disk radio model. Note that this is only for illustration purposes, in practice we could incorporate less ideal radio assumptions. We further assume that the grid setup is *ad hoc*, so that no node knows its position in the network in advance. However, some nodes are sink nodes to which all other nodes forward messages, possibly via intermediate relay nodes. To allow nodes to route messages towards the sink, each sink in the network sends a beacon message in regular intervals containing a pheromone value. From zoology, pheromone is a hormone laid down by colony-based insects, to indicate popular routes to food sources or new nest sites. In a similar manner pheromone gradients have been adapted in the WSN literature as an abstract means of studying the evolution of routes from source to sink nodes [7]. Upon reception of a pheromone beacon, a node updates its own pheromone level and broadcasts it. So long as the pheromone level update function generates levels that are strictly lower than the largest level received through incoming beacons, a pheromone gradient will emerge that is highest at sink nodes and lowest at nodes that are farthest away from any sink. Each node can subsequently relay messages to the sink via neighbours that have higher pheromone levels than itself, knowing that the message thereby travels closer to a sink. If nodes fail, they will stop emitting pheromone beacons and neighbours will hence learn to avoid them until they are back online.

While such self-organising routing protocols are preferred over centralised protocols in failsafe WSNs, it can happen that pheromone based routing decisions produce unwanted congestion in the network. During our model analysis we want to study the impact of the pheromone based routing approach on the buffer level of different nodes in the network. To do this we will measure the average buffer occupancy at each node and find out whether dynamic pheromone based routing distributes load in a fair manner. Naturally nodes closer to sink nodes usually have a higher buffer occupancy than more distant nodes, since they need to relay more messages to the adjacent sink. However, in large WSNs with multiple sink nodes we would like to ensure that all nodes that are at a certain distance from the sink will experience similar message loads. If this is the case then we can consider our network fair and well-structured. In the next section we describe our Castalia implementation of the failsafe WSN, followed by our corresponding PCTMC model.

3.1 Castalia protocol model

Our Castalia simulation consists of the following protocol stack. The application layer is a simple sensor application which periodically generates sensor readings, say at *sampleRate* samples per second and forwards them to a sink node. Our routing layer is based on concepts similar to those presented in [24], however, in our protocol the routing decisions are probabilistic. For the MAC layer our network uses CSMA. However, we assume that after having sent a message, nodes wait for a fixed period of time before sending the next message and listen for incoming messages in the meantime. Naturally, in a live application we would use a more energy efficient MAC protocol. While beyond the scope of this paper, one of our future research goals is the integration of realistic duty-cycle behaviour (cf. Section 2.3) into our mathematical WSN models, possibly using deterministic delays such as the ones suggested in [17].

Our pheromone network protocol implementation is as follows. At a regular time interval, every node broadcasts a pheromone beacon to all its neighbours. A neighbour that receives this message will store it in a neighbourhood table, along with the *id* of the sending node. If the pheromone level contained in the beacon is larger than the receiving node's current pheromone level, the receiver will update its pheromone level. If node i receives a beacon from node j and if the pheromone level of j is higher than that of node i, then

$$ph@l_i = (ph@l_i + ph@l_j)/2 \tag{4}$$

where $ph@l_*$ are floating point values stored by each individual node[2]. To ensure that pheromone evaporates over time, a non-sink node i always decreases its pheromone level before broadcasting its own pheromone beacon

$$ph@l_i = ph@l_i - \sqrt{ph@l_i} \tag{5}$$

Sink nodes, however, being a source of pheromone, have constant pheromone levels. Given its own pheromone level and the levels of its neighbours, a node can make informed routing decisions for application layer packets. We assume that in regular intervals of x milliseconds a node attempts to forward a message from its buffer to the next link in the multihop route. x is chosen so that $1000msec/x$ yields the desired number of *fwRate* messages a node can send every second. We assume that this number is much lower than the physical bandwidth of the communication channel such that no message has to be discarded. Buffers contain both messages from the node's own application layer as well as messages from other nodes that it needs to forward to the sink. To decide which node to relay a message to, nodes use a probabilistic routing approach. Let $T_i = \{ph@l_1, \ldots, ph@l_m\}$ be the set of all pheromone levels $ph@l_j$ of neighbours of node i such that $ph@l_j > ph@l_i$, i.e. T_i contains the pheromone levels of all neighbouring nodes of i that are closer to a sink. The probability that node i routes a message via node k then becomes

$$\mathbb{P}(send\ msg\ to\ k) = ph@l_k \ / \sum_{ph@l_j \in T_i} (ph@l_j - ph@l_i) \tag{6}$$

which is a discrete distribution we can easily sample from. When simulating the protocol stack in Castalia we regularly

[2]These should not be confused with species labels $Ph@l_i$ used in Section 3.2, which are part of the PCTMC model definition.

measure the number of messages in every node's buffer. For sink locations we assume that messages are processed immediately and thus we are not concerned about their buffer status. The source code of the resulting simulation as well as detailed simulation parameter configurations can be found on our website [14]. Later, when we discuss our Castalia simulation results we will distinguish between collision free communication and the more realistic collision model that simulates channel interference [6]. Moreover, we assume a sharp noise threshold for the radio reception as well as constant radio signal strength to recreate the unit disk effect.

3.2 PCTMC protocol model

This section reviews the interference-free version of our PCTMC model, which was taken from [16]. Our PCTMC model has been implemented in a stochastic process algebra that is part of the GPA-analyser tool [28]. For illustration purposes, we will represent the core elements of the model using reactions in the style of Eq. (1). In the following we implicitly assume that random variables are time-dependent and thus drop the (t) parameter used in Eq. (1). To avoid confusion between a species label and the random variable that describes the population of a given species we use $State@l_i$ and $\#State@l_i$ respectively.

We will deal with 3 types of nodes: functioning sink and non-sink nodes as well as broken nodes. Non-sink nodes forward their data samples to sink nodes. When a node stops communicating with other nodes it is considered broken. Each node is further assumed to have a finite buffer that can hold up to m messages. The buffer holds a node's own samples as well as messages that it needs to relay. We start by describing the PCTMC events for working sink nodes and non-sink nodes. Clearly, broken nodes do not interact with other nodes and thus we do not need to represent them in the model. The reactions for the pheromone spread for non-sink nodes are as follows

$$\begin{array}{lll} \emptyset & \to Ph@l_* & \text{at } pheroIncRate@l_* \\ Ph@l_* & \to \emptyset & \text{at } pheroDecRate@l_* \end{array} \tag{7}$$

where $pheroIncRate@l_*$ is the sum of the difference between a node's pheromone level and that of its neighbours. For sink nodes the pheromone population level remains constant, so no events need to be defined. Assuming we have a neighbourhood structure as shown in Figure 2, we would obtain the following reaction rates for the pheromone changing events of the node at location 3

$$\begin{aligned} pheroIncRate@l_3 = {c_1}^* \, [&(\max(0, \#Ph@l_2 - \#Ph@l_3) + \\ &\max(0, \#Ph@l_6 - \#Ph@l_3)] \end{aligned} \tag{8}$$

where c_1 is a constant that regulates the rate at which pheromone levels change. In general $pheroIncRate@l_*$ is defined as

$$\sum_{l_j \in neighbours \ \ of \ \ l_*} {c_1}^* \max(0, \#Ph@l_j - \#Ph@l_*) \tag{9}$$

Moreover,

$$pheroDecRate@l_* = {c_2}^* \min(0.1, \#Ph@l_* - 2) \tag{10}$$

describes the rate at which pheromone evaporates, i.e. the PCTMC analogue of Eq. (5), with c_2 being another constant that is chosen so that the pheromone spread is quite linear. The min term ensures that pheromone levels will remain positive. Although this pheromone gradient only encodes a node's distance from the sink, it is generally 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. Figure 3 illustrates the pheromone mean-field results when applying the above reactions to the topology shown in Figure 2 for a given initial pheromone population at sink location l_1.

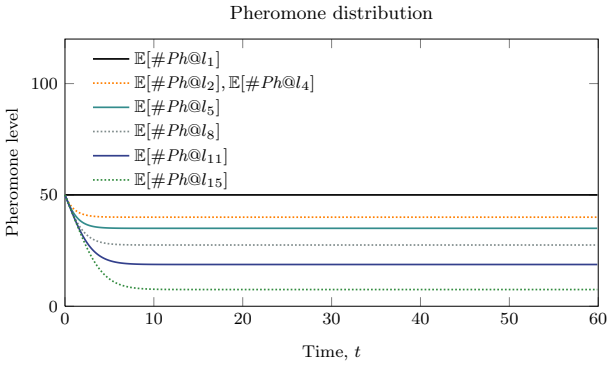


Figure 3: Pheromone distribution in the network depicted in Figure 2.

Having described PCTMC events governing the dynamics of pheromone agents $Ph@l_*$, we now need to represent the routing probability distribution used by the Castalia protocol stack (cf. Eq. (6)). To avoid reactions that have fractions of populations in their reaction rates, we use a routing species which can be used to infer routing probabilities from pheromone levels. Take for instance the node at location l_6, which can route up messages to location l_5, route left to l_3 and route right to l_9. As in Eq. (6) we want nodes with higher pheromone levels to be used more frequently. Hence, we need to increase the proportion of messages sent to the left, as the pheromone level of node 3 becomes larger than that of nodes 5 and 9

$$\begin{aligned} &RouteU@l_6 \to RouteL@l_6 \\ &\quad \text{at } \#RouteU@l_6{}^* \max(0, \#Ph@l_3 - \#Ph@l_6) \\ &RouteR@l_6 \to RouteL@l_6 \\ &\quad \text{at } \#RouteR@l_6{}^* \max(0, \#Ph@l_3 - \#Ph@l_6) \end{aligned} \tag{11}$$

where $\max(0, \#Ph@l_3 - \#Ph@l_6)$ is the pheromone excess of node 3 over 6. Clearly, if location 3 has a lower pheromone level than location 6, node 6 will not route messages via node 3 since all transitions that increase $\#RouteL@l_6$ have a 0 reaction rate. If node 3 has a higher pheromone level, the population $\#RouteL@l_6$ will be non-zero and thus a proportion of messages from node 6 is relayed to the sink via location 3. All reactions for $RouteU@l_*$, $RouteD@l_*$, $RouteL@l_*$ and $RouteR@l_*$ follow the same pattern. We define the constant c_3 to be a normalising constant that allows us to compute the proportion of messages that are forwarded to a particular neighbour

$$\begin{aligned} c_3 = \ & \#RouteU@l_* \ + \ \#RouteD@l_* \ + \\ & \#RouteL@l_* \ + \ \#RouteR@l_* \end{aligned} \tag{12}$$

We then use the resulting normalised distribution to model a similar routing behaviour as described in Eq. (6). The routing population reactions are the same for sink and non-

sink nodes and broken nodes are automatically avoided due to their constant pheromone level of 0.

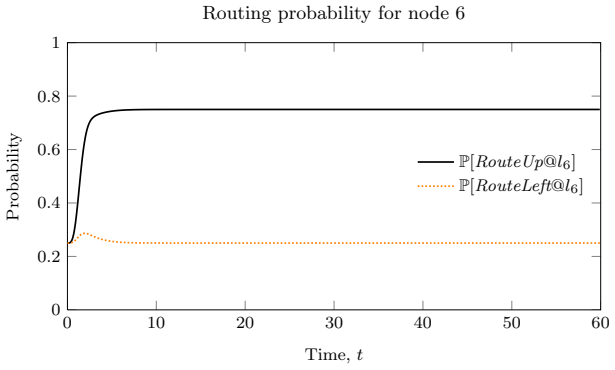


Figure 4: Routing distribution for node 6 in Figure 2.

To express the effects of dynamic routing in the message exchange, we need to feed back the routing information into the PCTMC events that describe message exchanges between nodes. For the message buffer we decided to represent different buffer levels as individual states in our PCTMC. A single node has buffer states $\{Buf_0, \ldots, Buf_m\}$, so that if a node is in state Buf_i its buffer contains i messages. In all locations we chose m to be sufficiently high, so state Buf_m would hardly ever be reached, i.e. we do not need to handle buffer overflow scenarios. Buffers are affected by two different types of reactions. The simpler one is the increase that occurs when the sensor of a non-sink node generates a new sample

$$Buf_i@l_* \to Buf_{i+1}@l_* \quad \text{at } \#Buf_i@l_* * sampleRate \tag{13}$$

Change also occurs due to message exchange with neighbouring nodes. Take for example the message exchange between nodes 2 and 3

$$\begin{aligned} &Buf_i@l_2 + Buf_j@l_3 \to Buf_{i+1}@l_2 + Buf_{j-1}@l_3 \\ &\quad \text{at } \#Buf_i@l_2 * \#Buf_j@l_3 * \\ &\quad\quad \#RouteU@l_3/c_3 * fwRate \\ &Buf_j@l_2 + Buf_i@l_3 \to Buf_{j-1}@l_2 + Buf_{i+1}@l_3 \\ &\quad \text{at } \#Buf_j@l_2 * \#Buf_i@l_3 * \\ &\quad\quad \#RouteD@l_2/c_3 * fwRate \end{aligned} \tag{14}$$

where $0 \leq i < m$, $0 < j \leq m$. $fwRate$ represents the number of packets that each node can forward per time unit. Similar reactions can describe all other unicast message exchanges in the network. Moreover, we have

$$\sum_{i=0}^{m} \mathbb{E}[\#Buf_i@l_*] = 1 \tag{15}$$

In other words each node only has a single buffer instance. While sink nodes receive messages like non-sink nodes, we assume that they are connected to an uplink to which they immediately forward incoming messages

$$Buf_j@l_* \to Buf_{j-1}@l_* \text{ at } \#Buf_j@l_* * 4 * fwRate \tag{16}$$

For any node, the average number of elements in the buffer is given by

$$\sum_{j=1}^{m} j * \mathbb{E}[\#Buf_j@l_*] \tag{17}$$

A complete implementation of the model can be found in the model files provided on [14].

3.3 Modelling channel interference

It is not hard to see that the simple WSN model shown in Section 3.2 is only accurate in the absence of channel interference. By channel interference, we mean interference that is caused by the hidden terminal problem. Imagine two nodes A and C that have a common neighbour B with which they can both communicate, while at the same time A and C cannot directly communicate. Before sending a message using CSMA, both nodes check whether the channel is currently occupied. If it is free they send a message, otherwise they try again later. The hidden terminal problem occurs when C assesses the channel while A is transmitting a message to B. Since C cannot sense A's signal it assumes that the channel is clear and sends data to B. However, B now receives two signals simultaneously and can at best demodulate one of them. This effect is generally unavoidable in WSNs. Unlike wired networks, WSNs share a limited frequency band and hence no physically separated channel can be guaranteed for any pair of senders and receivers. Although newer protocols utilise multiple frequencies to reduce the number of nodes sharing a particular frequency, messages sent over a particular frequency can still be received by other nodes in hearing distance. As a consequence, unicast and broadcast of small packets essentially work the same in WSNs, the only difference being that in unicast mode nodes discard messages that are not intended for them.

Modelling channel interference in PCTMCs is generally nontrivial as it requires us to introduce further state variables in order to keep track of every node's radio state. To add these extra states to the model without increasing model complexity unnecessarily, we found that further simplifying independence assumptions were required. While it was possible to accurately reflect correlations between senders and receivers in the PCTMC events for the idealised network presented in Section 3.2, this becomes a lot harder as we introduce interference effects. To faithfully represent interference effects in a PCTMC model we need to incorporate the state of all 1- and 2-hop neighbours of the receiver when expressing message exchange events like in Eq. (3.2). However, this would result in an extremely large number of complex reactions involving 10 or more species each. Fortunately, as we only intend to use fast first-order mean-field approximation techniques to analyse the model, simplifying assumptions can be made, which allow us to incorporate channel interference behaviour into our model without resorting to complex and verbose model descriptions.

While mean-field analysis often outperforms simulation even for moderate population sizes, it is important to bear in mind that although the number of first-order ODEs grows linearly in the number of PCTMC states, the number of moments required to approximate higher-order moments increases exponentially in the order of the highest moment that we require [18]. In practice this limits mean-field analysis of large WSNs to first-order moment approximations. However, depending on the nature of the reactions, we require both first- and higher-order moments just to obtain ac-

curate first-order mean-field results [9, 14]. Technically this is the case whenever the PCTMC has reactions such as Eq. (14) where rates are non-linear in population terms. In such cases the only way to obtain a closed system of first-order approximating mean-field ODEs is to assume statistical independence between populations. Although accurate correlations are generally vital for models, empirical evidence suggests that for many PCTMC models the independence assumption required for first-order mean-field analysis often has a limited effect on the accuracy of the mean approximation [7, 13]. In the following we will therefore describe interference dynamics in our PCTMC model assuming statistical independence between populations. Naturally, the resulting simplified PCTMC model can only be used for a first order mean-field approximation of the semantically more accurate PCTMC model whose reactions capture all correlations.

For our interference model we assume that a node's radio is either in RX, TX or $Idle$ state. When a node is $Idle$ it can either receive a new message and switch to RX until the exchange has completed or decide to send a message and switch to TX for the duration of the transfer. Hence, a node can only send a new message when it is in $Idle$ state. As we assumed the total bandwidth to be much larger than the number of messages a node attempts to forward every second (cf. Section 3.2), the introduction of the radio state will have no visible impact on the rate of the event defined in Eq. (3.2). The following reactions describe the local transition of a node's radio state

$$\begin{array}{lll} RX@l_* & \rightarrow Idle@l_* & \text{at } \#RX@l_*{}^*900 \\ TX@l_* & \rightarrow Idle@l_* & \text{at } \#TX@l_*{}^*900 \end{array} \tag{18}$$

such that the transmission of a single messages takes about 1/900 seconds. Moreover, we need to have

$$\#RX@l_* + \#TX@l_* + \#Idle@l_* = 1 \tag{19}$$

The following events describe state changes due to pheromone beacon broadcasts, assuming a single beacon is sent every second

$$\begin{array}{lll} Idle@l_* & \rightarrow RX@l_* & \text{at } \sum_{n\in N@l_*} \#Idle@l_* \\ Idle@l_* & \rightarrow TX@l_* & \text{at } \#Idle@l_* \end{array} \tag{20}$$

where $N@l_*$ is the set of 1-hop neighbours of the node. Similarly a node changes from $Idle$ to RX when one of its neighbours is transmitting a message and to TX when sending a message itself

$$\begin{array}{l} Idle@l_* \rightarrow RX@l_* \\ \quad \text{at } \#Idle@l_*{}^* \sum_{n\in N@l_*} (1 - \#Buf_0@l_n)^* fwRate \\ Idle@l_* \rightarrow TX@l_* \\ \quad \text{at } \#Idle@l_*{}^* (1 - \#Buf_0@l_*)^* fwRate \end{array} \tag{21}$$

Having defined these transitions we can now easily describe interference as using the following reactions. The first one describes interference of data message transfers caused by pheromone beacons

$$\begin{array}{l} Buf_j@l_* \rightarrow Buf_{j-1}@l_* \\ \quad \text{at } \sum_{n\in N2@l_*} \#RX@l_*{}^* \#Buf_j@l_* \end{array} \tag{22}$$

where $0 < j \leq m$ and $N2@l_*$ is the set of all 1-hop and 2-hop neighbours of the receiving node. Although we assume that we generally cannot receive messages from 2-hop neighbours, transmission of messages by 2-hop neighbours can still cause interference. Similarly to the pheromone messages above, data messages sent by the receiving node's 1-hop and 2-hop neighbours may also result in message loss

$$\begin{array}{l} Buf_j@l_* \rightarrow Buf_{j-1}@l_* \\ \quad \text{at } \#RX@l_*{}^* \sum_{n\in N2@l_*} (1 - \#Buf_0@l_n)^* fwRate \end{array} \tag{23}$$

Finally, we need to add one more interference event which ensures that when 1-hop neighbours interfere with the node that they want to send a message to, two messages are lost

$$\begin{array}{l} Buf_j@l_* \rightarrow Buf_{j-1}@l_* \\ \quad \text{at } \#RX@l_*{}^* fwRate^* \\ \qquad \sum_{n\in N@l_*} (1 - \#Buf_0@l_n)^* \#Routel_*@l_n/c_3 \end{array} \tag{24}$$

where $\#Routel_*@l_n/c_3$ is the routing probability from node n to the node located at l_*.

4. ANALYSIS

In this section we compare the normalised buffer levels of nodes in a WSN running the failsafe application described in Section 3. The aim of our analysis and the way it is presented is to show that there is good qualitative agreement between the Castalia simulations results and their PCTMC mean-field approximations, even when the number of messages sent per second is low. If we can show such agreement with low-level WSN simulation results in a range of scenarios, we will have gained a measure of belief in the mean-field approach for spatial WSN analysis. To obtain the buffer levels at steady-state from Castalia, we first allowed the network a 60 second period to establish an initial pheromone gradient and subsequently measured the average buffer level at each node over a period of 240 seconds. For each configuration we then took the average over 50 such simulations to obtain the mean buffer level at each location. The resulting 95% confidence intervals for the mean buffer levels lay within $< 3\%$ for all locations. Similarly, we evaluated the mean buffer level using mean-field ODEs and subsequently took the average buffer occupancy at each location in the network after 60 seconds. Throughout our experiments we found that the absolute buffer levels predicted by the PCTMC and Castalia model were quite different. Most likely this is due to the simplifying assumptions that we made in order to create a PCTMC model that is suitable for fast ODE solution techniques. While this error is of interest for future work, in the following we only study the normalised mean buffer levels at steady-state. By normalised we mean that all locations are normalised by the non-sink node with the highest buffer level, which is then assumed to have a buffer level of 100%. We use the busiest non-sink node rather than the busiest sink node as we do not want to make additional assumptions about the uplink capacity of sink nodes. To visualise hot spots and idle locations we use spatial heat maps. In these heat maps a node with a relative buffer level of $x\%$ is coloured black at $x\%$ opacity. Sink nodes are coloured at 100% opacity and are marked 's', whereas broken nodes are marked 'x' and coloured white. In addition to these heat maps, we also deploy difference maps to visualise relative errors between the PCTMC and the Castalia models. To obtain these we simply subtract normalised PCTMC buffer levels from normalised Castalia ones for each location and divide by the normalised Castalia buffer levels. In contrast to the mean buffer level heat maps the difference maps show the resulting error in % for each location. While the heat maps support our claim that the

PCTMC and the Castalia results match qualitatively, i.e. relative differences between busy and idle locations are preserved, the difference plots capture the quantitative accuracy better. Unless stated otherwise we conduct the below experiments with a forward rate of up to 20 messages per second and message production rate of 1 message per second.

4.1 Ideal communication

First, we will investigate the Castalia and the PCTMC predictions for the data flow assuming ideal communication, i.e. without interference or radio signal strength variation. Figure 5 shows the results for a network with 49

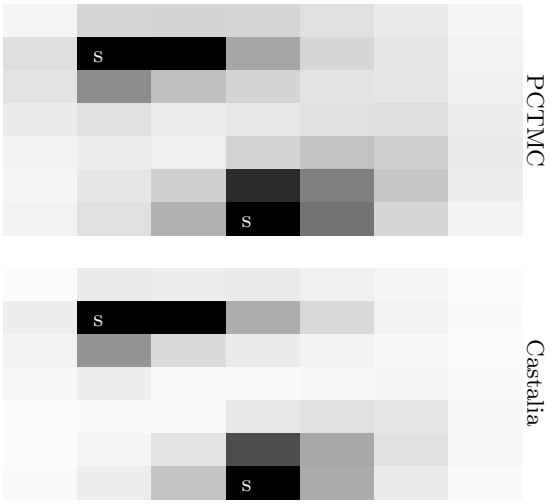


Figure 5: Data flow in a network with 49 nodes, 2 sinks, a forward rate of 20 messages per second and a message production rate of 1 message per second. The top diagram is based on the steady-state PCTMC mean-field approximation, the bottom one on the averaged Castalia simulation results under ideal radio conditions.

nodes and 2 sinks. In both models the nodes under high load are the same, despite the fact that the average absolute buffer level in the Castalia level is roughly 50% higher than in the PCTMC model. Note, however, that the Castalia simulation heat map has a much higher contrast. In other words the actual magnitude of the difference between busy and idle locations approximated by the mean-field analysis is lower than the Castalia simulation results suggest.

Our second example in Figure 6 illustrates the effect on normalised buffer levels as we increase the message production rate to 1.5 messages per second without changing the network topology. The increase in contrast in the heat map of the PCTMC model indicates that the relative difference between the buffer level of the busiest node compared to the other nodes has become larger. This is what we expected to happen, as the busiest nodes will naturally experience much higher buffer occupancy under load. In Section 4.2 we will show that interference slows down this effect since busier nodes start to lose more packets. The difference plot in Figure 6 shows that the results of the PCTMC mean-field analysis and the Castalia simulation model are much better in locations with high traffic than in areas of low traffic. This is not surprising since mean-field analysis is generally

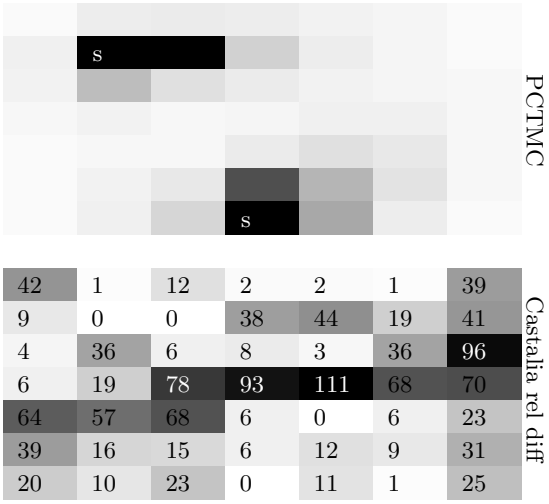


Figure 6: Data flow in a network with 49 nodes and 2 sinks with ideal communication and 1.5x message production rate.

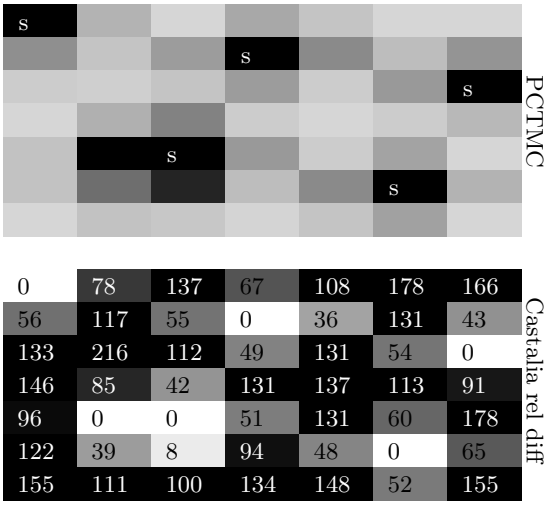
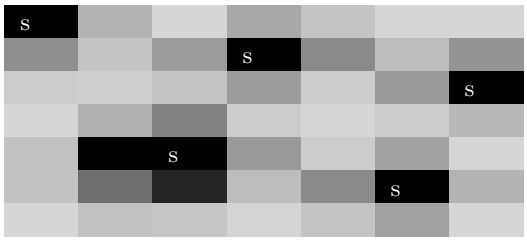


Figure 7: Data flow in a network with 49 nodes and 5 sinks with ideal communication.

most accurate when events rates, i.e. the message production and forwarding rate in this example, are high. If our PCTMC model truly represents the process described by the Castalia model, the error should thus decrease as we scale both message production and forwarding rate. To illustrate this effect, our next example looks at the effect on the error in a model with 49 nodes and 5 sinks as we scale the message production rate of 1 message per second and the forwarding rate of 20 messages per second (cf. Figure 7), by a factor of 10 (cf. Figure 8). As can be seen in the difference plots, the error between the PCTMC mean-field and the Castalia simulation results decreases drastically as the rates are increased. Therefore, in the ideal communication scenario the normalised PCTMC mean-field buffer levels can be seen as a limiting case for the network and the protocol when both forward and sampling rate are extremely high.

In practice, individual WSN nodes fail eventually, requir-



0	3	2	1	3	15	11
3	4	4	0	3	16	1
9	31	1	0	5	1	0
2	7	3	8	2	5	3
7	0	0	0	8	6	15
16	7	4	1	4	0	6
6	18	1	3	12	3	6

Castalia rel diff

Figure 8: Data flow in a network with 49 nodes and 5 sinks with ideal communication, 10x forward and message production rate.

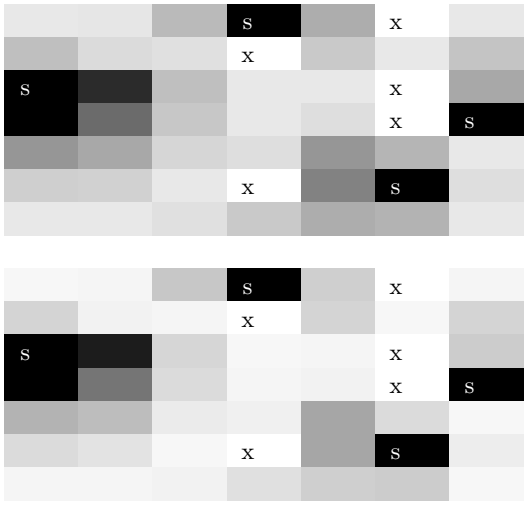


Figure 9: Data flow in a network with 49 nodes and 4 sinks and 5 broken nodes with ideal communication.

ing the network to reconfigure itself. In Figure 9, we can see that even in presence of broken nodes, both models produce qualitatively similar results. This confirms that the failsafe aspects of the pheromone model are captured well in both instances. Finally, in order to illustrate that our PCTMC model is indeed scalable, we created a large model with 100 nodes, several sinks and broken nodes. The data flow created during the subsequent analysis is shown in Figure 10. As can be seen even in this large network the ODE approximation produces useful data flow predictions.

4.2 Interference

Having compared at the Castalia simulation results with PCTMC mean-field predictions for the data flow in an idealised communication scenario, we now repeat the analysis taking signal interference into account. Apart from this, no changes were made with respect to the analysis. In particular we still assert fixed radio signal strength throughout our experiments. The unit disk model has been configured such that nodes can only detect a busy nodes if they are within their transmission distance, i.e. 1-hop neighbours. However, we assume that the signal strength of 2-hop neighbours is strong enough to cause interference.

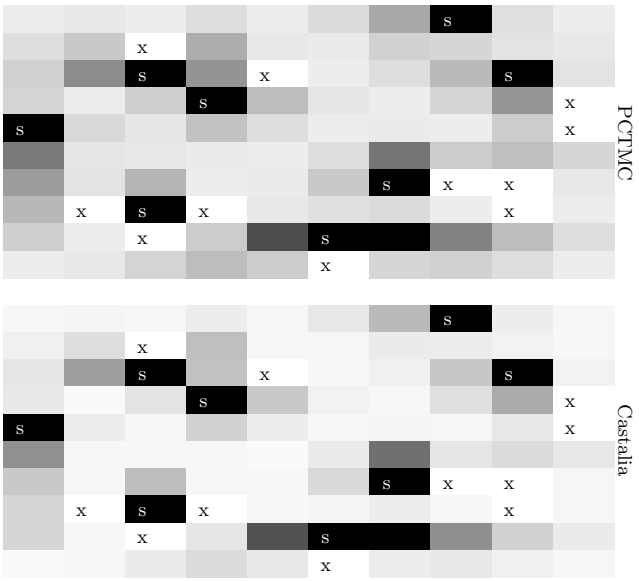


Figure 10: Data flow in a network with 100 nodes several sinks and broken nodes with ideal communication.

Interference is most likely to cause message loss in areas of high traffic, i.e. among nodes close to a sink. As a consequence we expect that under equal network load the relative differences between buffer levels become less extreme than in the examples shown in Section 4.1. Indeed, comparing the interference-free results shown in Figures 5, 6 and 10 with those in Figures 11, 12 and 13 we can see that the contrast of the heat maps is visibly lower when we model interference. In Figures 11, 12 and 13 it is also noticeable that the contrast of the cells in the PCTMC heat maps are lower than in the Castalia case, which implies that the channel interference effect is less severe in the realistic low-level simulation. All four examples with interference produce decent qualitative results, although the relatively small message load causes qualitative differences in some places. We further compared the PCTMC mean-field without interference to the Castalia results with interference for Figures 11, 12 and 13. When comparing the resulting relative errors of the 5 busiest locations in both cases, it turns out that the maximum error of our extended PCTMC model with interference (cf. Section 3.3) produces between 3 to 10 times more accurate results than the simple PCTMC model without interference (cf. Section 3.2).

Like the comparison between Figure 7 and Figure 8 in the previous section, the difference between Figure 13 and Figure 14 shows that also in the presence of interference our extended PCTMC model approaches the Castalia model as we scale the message forwarding and generation rate.

Most interestingly the quality of the prediction does not visibly deteriorate as we add interference effects. This is es-

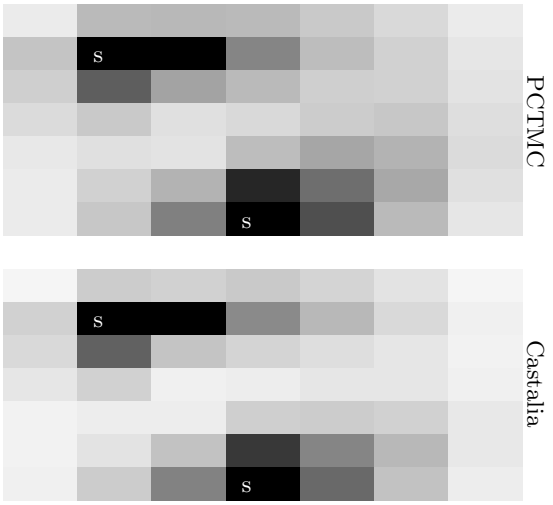


Figure 11: Data flow in a network with 49 nodes, 2 sinks, a forward rate of 20 messages per second and a message production rate of 1 message per second. The heat map at the top is based on the PCTMC mean-field approximation, the bottom one on the Castalia simulation results. We assume communication with interference without variation in radio signal strength.

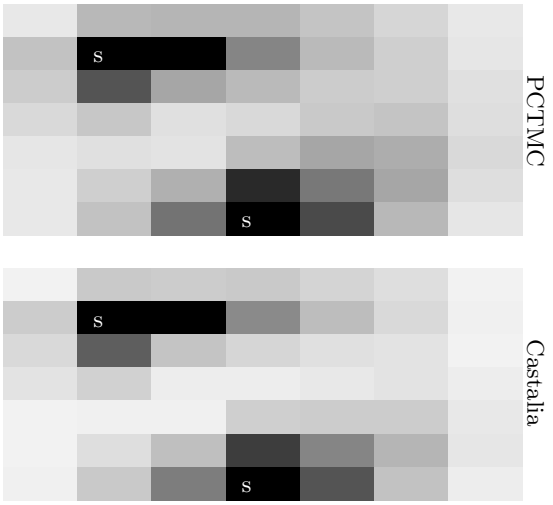


Figure 12: Data flow in a network with 49 nodes and 2 sinks with signal interference and 1.5x message production rate.

pecially surprising since we compare a simple mathematical PCTMC model to simulation results from a sophisticated low-level packet collision model. However, we do need to stress that we are just looking at first-order moments and that in its current form our model cannot be used to approximate higher-order moments of the underlying stochastic process.

4.3 Low-level Simulation vs high-level ODEs

One of our main arguments for the use of mean-field techniques over discrete event simulation is that the computational burden of simulations increases drastically as we make our model larger. The generation and the analysis of the ODEs representing the first order moments of the PCTMC models generally took $1-2$ minutes for the 49/100 node models respectively, while 50 Castalia simulation runs took between $5-15$ minutes on a modern quad-core desktop computer, depending on the message load and forwarding rate used in a given scenario.

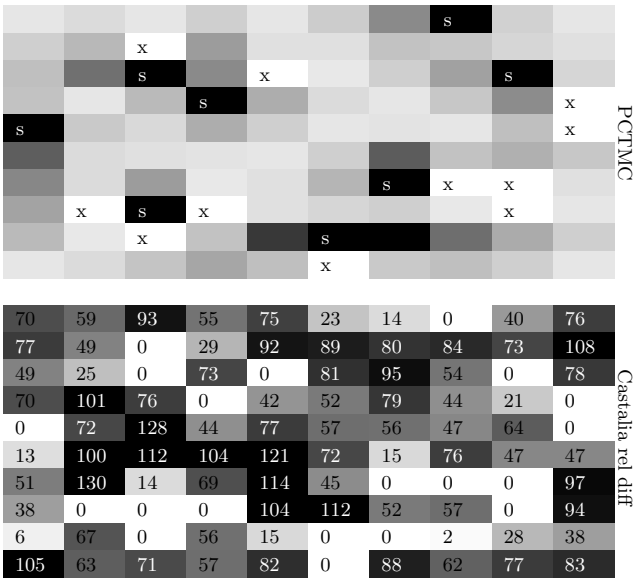


Figure 13: Data flow in a network with 100 nodes several sinks and broken nodes in presence of interference.

While it is not unlikely that the simulation time grows linearly with the model size, it has to be taken into account that different setups, for instance a higher message production rate, slows down the simulation, whereas the evaluation time of the ODEs is not affected by such parameter changes unless the system becomes stiff[3]. However, for a fair comparison between simulation and ODE analysis speed, we need to take into account confidence intervals for the simulation results of the buffer size in order to determine the optimal length for each simulation run and the optimal number of replications required in order to obtain good sample mean estimates. Here, our decision compute the average buffer levels using 50 simulation produced tight 95% confidence intervals were ($< 3\%$ of the actual mean buffer size for all locations). The reason we only need a small number of repetitions to obtain tight intervals is that the buffer level, computed for each location in each simulation run, represents a steady-state average over a period of 240 seconds rather than a single measurement at the end of each simulation run. Needless to say that for a WSN with a single busy location we would thus require even less simulation runs to identify the node with the highest load in the network. However, the more marginal the differences between busy locations are, the more expensive the Castalia simulation will

[3]A stiff system of ODEs can be solved using slower solvers, which dynamically adapt the integration step size according to the relative differences between gradients in order to avoid large numerical errors.

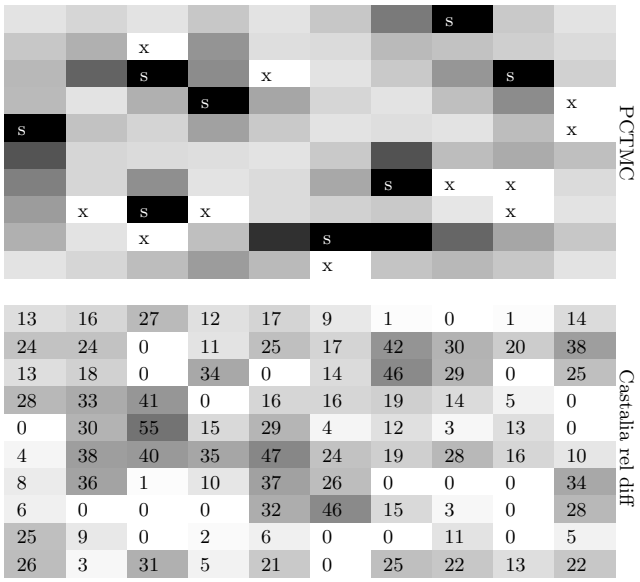


Figure 14: Same setup as in Figure 13 but with 2x forward and message production rate.

become, since we need more repetitions in order to further reduce the confidence interval width. This is especially the case when optimising parameters in a protocol, as we have to ensure that a change in parameter actually manifests in a measurable statistical improvement.

Finally, in order to obtain a fair comparison between the a simulation and a mean-field based evaluation approach we would also need to take into account the framework implementation. Our ODE analysis was conducted using the GPA-analyser tool [28], which is implemented in Java, whereas Castalia is implemented in C++. As this discussion illustrates, an accurate benchmark is itself a difficult challenge and beyond the scope of this paper. However, it is true to say that for the above data flow experiments, the mean-field technique is a fast, scalable evaluation method. Given that low-level WSN simulations are computationally expensive, the mean-field method presents a very attractive approach, not as a replacement analysis technique but rather in unison with simulation. This is particularly useful when the mean-field technique is considered as a heuristic methodology for reducing the design parameter search space that can then be explored in more detail by simulation.

5. CONCLUSIONS

In this paper we have presented a PCTMC model for the data flow analysis of large, distributed WSNs running a fail-safe pheromone based routing protocol. The data flow results presented in Section 4, are the result of a large number of interacting processes including pheromone spread, dynamic routing decisions and buffer level estimation. Moreover, we illustrated that our model produces good qualitative results even in abundance of channel interference or when message volumes are low. Furthermore, we showed that as we scale up the number of messages sampled by each node and the rate at which messages are forwarded, the mean-field approximation becomes even more accurate.

Despite encouraging results, further research is needed to improve the representation of WSN features, such as interference and duty-cycle behaviour, in PCTMC models. The availability of such key concepts is essential if PCTMC modelling aims to become an established analysis tool for the WSN community. While some aspects of WSN protocols might be hard to represent using only exponential delays, we further intend to study mean-field techniques for Generalised Semi-Markov Processes (GSMPs) [17] which are likely to work better under certain circumstances. However, given the variety of tools for mean-field evaluation of PCTMC models, it is worth trying to develop further PCTMC modelling techniques for WSNs first.

6. ACKNOWLEDGEMENTS

Jeremy Bradley is supported in part by EPSRC on the AMPS project, the Analysis of Massively Parallel Stochastic Systems, ref. EP/G011737/1.

7. REFERENCES

[1] Crossbow: Crossbow datasheet on MicaZ, 2006.

[2] 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*, pages 941–948, 2010.

[3] M. S. Al-Fares and Z. Sun. Self-Organizing Routing Protocol to achieve QoS in Wireless Sensor Network for Forest Fire Monitoring. *Systems Research*, pages 211–216, 2009.

[4] A. Bachir, M. Dohler, T. Watteyne, and K. K. Leung. MAC Essentials for Wireless Sensor Networks. *IEEE Communications Surveys Tutorials*, 12(2):222–248, 2010.

[5] 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*, pages 103–104. ACM, 2010.

[6] A. Boulis. Castalia: revealing pitfalls in designing distributed algorithms in WSN. In S. Jha, editor, *Proceedings of the 5th international conference on Embedded networked sensor systems*, pages 407–408. ACM, 2007.

[7] D. Bruneo, M. Scarpa, A. Bobbio, D. Cerotti, and M. Gribaudo. Markovian agent modeling swarm intelligence algorithms in wireless sensor networks. *Performance Evaluation*, 69(3-4):135–149, Mar. 2012.

[8] 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*, 44(7):64–73, 2006.

[9] S. Engblom. Computing the moments of high dimensional solutions of the master equation. *Applied Mathematics and Computation*, 180(2):498–515, 2006.

[10] 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.

[11] V. Galpin. Towards a spatial stochastic process algebra. In *Proceedings of the 7th Workshop on Process Algebra and Stochastically Timed Activities (PASTA)*, Edinburgh, 2008.

[12] D. T. Gillespie. Exact stochastic simulation of coupled chemical reactions. *Journal of Physical Chemistry*, 81(25):2340–2361, 1977.

[13] 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)*, pages 300–305, 2008.

[14] M. C. Guenther. PCTMC model and Castalia simulation code (http://www.doc.ic.ac.uk/˜mcg05/wsnrouting), 2012.

[15] M. C. Guenther and J. T. Bradley. Mean-field performance analysis of a hazard detection Wireless Sensor Network. In *VALUETOOLS'12*, 2012.

[16] M. C. Guenther and J. T. Bradley. PCTMC models of Wireless Sensor Network protocols. In *UKPEW'12, The 28th UK Performance Engineering Workshop*, 2012.

[17] R. A. Hayden. Mean-field approximations for performance models with generally-timed transitions. *ACM SIGMETRICS Performance Evaluation Review*, 2011.

[18] R. A. Hayden and J. T. Bradley. A fluid analysis framework for a Markovian process algebra. *Theoretical Computer Science*, 411(22-24):2260–2297, 2010.

[19] J. Hillston. Fluid flow approximation of PEPA models. *Second International Conference on the Quantitative Evaluation of Systems QEST05*, pages 33–42, 2005.

[20] R. Hornig and A. Varga. AN OVERVIEW OF THE OMNeT++ SIMULATION ENVIRONMENT. *Proceedings of the First International ICST Conference on Simulation Tools and Techniques for Communications Networks and Systems*, pages 1–10, 2008.

[21] P. Levis and N. Lee. TOSSIM : A Simulator for TinyOS Networks. *UC Berkeley September*, pages 1–17, 2003.

[22] Ns. The Network Simulator - ns-2, 2002.

[23] M. Opper and D. Saad. *Advanced Mean Field Methods: Theory and Practice*. The MIT Press, 2001.

[24] 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*, 31(Nca):265–268, 2007.

[25] M. Ramesh, N. Vasudevan, and J. Freeman. Real Time Landslide Monitoring via Wireless Sensor Network. *Geophysical Research Abstracts*, 11(EGU2009-14061):14061–14061, 2009.

[26] 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*, 1(3):571–574, 1999.

[27] 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.

[28] A. Stefanek, R. A. Hayden, and J. T. Bradley. A new tool for the performance analysis of massively parallel computer systems. *Eighth Workshop on Quantitative Aspects of Programming Languages QAPL 2010 March 2728 2010 Paphos Cyprus*, 2010.

[29] A. Stefanek, R. A. Hayden, and J. T. Bradley. Fluid computation of the performance-energy trade-off in large scale Markov models. *SIGMETRICS Performance Evaluation Review*, 2011.

[30] S. Verma, N. Chug, and D. V. Gadre. Wireless Sensor Network for Crop Field Monitoring. In *2010 International Conference on Recent Trends in Information, Telecommunication and Computing*, pages 207–211. IEEE, Mar. 2010.

[31] P. J. Wangersky. Lotka-Volterra population models. *Annual Review of Ecology and Systematics*, 9(1):189–218, 1978.

[32] 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*, 20(7):13–24, 2004.

[33] P. Yadav. *Cross-Layer Protocols to Support Periodic Data Collection and Event Driven Wireless Sensor Network Applications*. Phd thesis, Imperial College, 2011.

[34] J. Yick, B. Mukherjee, and D. Ghosal. Wireless sensor network survey. *Computer Networks*, 52(12):2292–2330, 2008.