Wireless Network Characterization via Phase Diagrams

Maria Michalopoulou and Petri Mähönen
RWTH Aachen University, Institute for Networked Systems
Kackertstrasse 9, D-52072 Aachen, Germany
Email: {mmi, pma}@inets.rwth-aachen.de

***Abstract*—Like physical systems can exist in different phases, the concept of different phases (or states) can be also attributed to other types of systems, such as wireless networks. In this paper we employ the notion of phase diagrams to characterize a wireless network by means of a reduced number of simulation points. By reducing the amount of simulations – compared to fine-grained simulation-based analyses – we decrease the accuracy of the result, however, not in an arbitrary way. The idea is to derive the so-called phase diagram that fully characterizes the system with respect to the points of state changes. Typically, the different states of a system correspond to distinguishable macroscopic properties, therefore the phase transition points are in a qualitative sense points of particular significance for a system under study. We show how phase diagrams can be estimated from a limited set of simulation points by means of Support Vector Machines.**

I. Introduction

A physical system can exist in a number of different phases, which can be distinguished from each other by a set of well-defined parameters, termed *state variables*. For instance, most substances can exist in the solid, liquid, or gas state depending on the values of temperature, pressure, and volume, which are the state variables in this particular example. Nevertheless, this notion of different phases (or states)[1] is not limited to physical systems, but can be applied to other types of systems as well, such as wireless networks. For example, the throughput of Carrier Sense Multiple Access (CSMA)-based protocols can be raised up to a certain point where it becomes saturated and then remains constant. Therefore, we can clearly identify two different phases with respect to throughput, a non-saturated and a saturated throughput phase.

In the present paper we aim to introduce the concept of *phase diagrams* in wireless networking. A phase diagram represents the space of all possible phases of a system with respect to the state variables. More specifically, the lines on a phase diagram refer to the phase boundaries, that is, to the dividing lines between two states. By means of a phase diagram we are able to determine the state of the system for every possible combination of values that the state variables can take.

It is of significant interest to note that phase diagrams are inherently related to *phase transitions* as the crossing of a line on a phase diagram corresponds to a phase transition phenomenon. The term phase transition originates in physics and particularly in statistical physics [1], but phase transition phenomena have been found to occur within the context of many scientific fields, such as graph theory, optimization theory, computational complexity, traffic theory and wireless networking. To the extent of our awareness, phase diagrams have not been considered in wireless networking research. There is, however, a limited amount of literature addressing phase transitions occurring in wireless networks. For example, in [2] Durvy et al. identify two different phases as far as the amount of fairness in CSMA/CA wireless networks is concerned, depending on the size of the topology. Krishnamachari et al. ([3], [4]) study several examples of phase transitions for a number of wireless network properties that are related to the network topology, such as connectivity, neighbour count, probabilistic flooding, etc. The authors in [5] examine the phase transition observed in network connectivity in terms of the rate of random node failures. In our previous work in [6] we attempted to utilize the concept of phase transitions in wireless network optimization in a rather practical way. Finally, in [7] we exploited the concept of equilibrium phase transitions and considered the analogy between the equilibrium state in physics and the Nash equilibrium that appears in game theoretic approaches; this work exhibits some similarity with the present paper in the sense that, apart from the concept of phase transitions itself, we attempt to make use of additional notions and approaches originating in the field of statistical mechanics.

The points where a system undergoes a phase transition are basically the points where significant qualitative changes in the properties of a system are observed. Therefore, the information supplied by a phase diagram is of significant importance for the characterization and understanding of the system under study. For example, we consider a phase transition with respect to some performance parameter; this means that the range of values the parameter can take can be divided in sub-ranges based on some qualitative characteristics. In the simplest case of a two-phase scenario the system can exist in two phases with respect to the specific performance metric, let the one phase correspond to an “acceptable (or good)” performance and the other one to an “unacceptable (or bad)” performance. Then, the corresponding phase diagram shall provide information about which regions of the state variable space lead to a good or – alternatively – bad performance. Therefore, compared to a fine-grained simulation analysis performed to

[1] in this paper we shall use the terms *phase* and *state* interchangeably

978-3-901882-63-0/2014 - Copyright is with IFIP

determine the behaviour of the system for a range of values of one or more parameters, deriving just a phase diagram decreases the amount of information obtained about the system under study, however, not in an arbitrary way.

Moreover, we show that the phase diagram can be estimated by means of a reduced set of simulations in comparison to a full simulation study, which makes our approach have a practical usefulness. Many realistic wireless simulations can take several hours or even a couple of days to run for a single case, thus the brute force simplified parameter space search, or Monte Carlo simulations, is sometimes not an attractive possibility. Alternatively, a phase diagram that can be derived much faster provides a more coarse, but still meaningful characterization of the system under study.

We suggest that a phase diagram can be derived from a limited set of simulation points by means of Support Vector Machines (SVMs). Specifically, simulation outcomes for a set of state variable combinations are used to train the SVM, whose objective will be the determination of the phase transition boundaries. We demonstrate our idea by means of a simple case study on a very well known model. Namely, in the following sections we present an example where the model under study is a CSMA/CA network and the performance metric of interest is the aggregate network throughput.

The remainder of the paper is structured as follows. In Section II we demonstrate the CSMA/CA model we have chosen to present as an example case study, we discuss the phase transitions occurring in the model and we construct the phase diagram. In Section III we discuss how the derivation of the phase diagram can be performed with minimal computational effort by means of an SVM. Finally, the paper is concluded in Section IV.

II. A Phase Diagram of the CSMA/CA Throughput

In this section we discuss the phase transitions observed in the CSMA/CA throughput and we derive the corresponding phase diagram. There are many parameters that determine the aggregate throughput and act as state variables for the system. The number of state variables we take into account specifies the dimensions of the phase diagram. For the sake of simplicity, and without loss of generality, in this section we will focus on a two-dimensional case in order to demonstrate our concept. Therefore, we select two parameters as state variables, namely the frame size and the total offered traffic. The simulated aggregate throughput in a CSMA/CA network with Binary Exponential Backoff (BEB) for several frame sizes against the total offered traffic is shown in Figure 1. As the offered traffic load increases the aggregate throughput increases linearly with the offered traffic, then it continuous to increase with a lower rate until the point it becomes saturated. For larger frame sizes the behaviour remains the same in a qualitative sense, but the performance becomes overall better, i.e., the saturation throughput increases with the frame size.

There is often a conception that the existence of a phase transition is to be considered only in the case of a drastic numerical discontinuity. However, in the context of statistical mechanics – where the term originates from – a phase transition is a sudden qualitative change in some property of the system, and does not necessarily correspond to a numerical discontinuity. For instance, let us consider the CSMA/CA aggregate throughput for a specific, arbitrary frame size (Figure 1). The throughput curve does not exhibit any abrupt and drastic change. But from a qualitative point of view we can identify different regions (phases). For lower values of traffic the aggregate throughput is approximately equal to the amount of offered traffic; this is a fully efficient regime where traffic can be handled with minimal delays. Then, the aggregate throughput continues to increase with the offered traffic but with a slower rate, thus the throughput is not any more approximately equal, but lower than the traffic; this is clearly a less efficient region of operation. Finally, as the offered traffic increases beyond a certain point the aggregate throughput remains constant; this is the saturated phase where we can say that practically the protocol cannot handle anymore the amount of offered traffic.

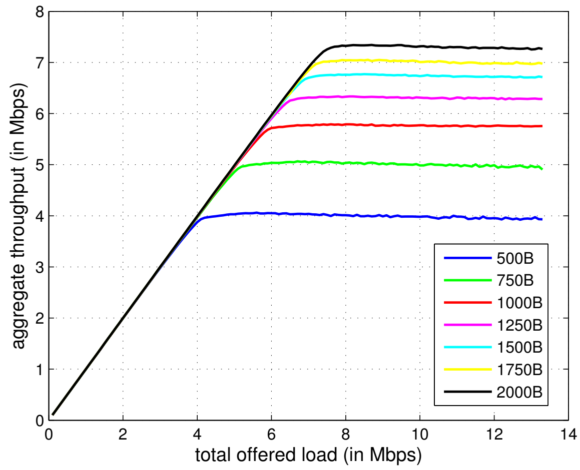

Fig. 1. The aggregate throughput obtained by a CSMA/CA MAC protocol with respect to the total offered traffic load for several frame sizes. Each user generates the same amount of traffic. The minimum Backoff Exponent is 5.

From this point of view the throughput constitutes a three-phase model. As discussed in the previous paragraph, we have a fully efficient phase (Phase 1), a less efficient phase (Phase 2), and a saturated phase (Phase 3). This means that the CSMA/CA throughput undergoes two phase transitions (we have one phase transition between phases 1 and 2 and another one between phases 2 and 3). Now we need to specify mathematically the phase boundaries or, equivalently, the critical points where each phase transition occurs.

Let us define the normalized aggregate throughput, T_{norm}, as the ratio of the aggregate throughput to the amount of offered traffic

$$T_{\text{norm}} = \frac{\text{aggregate throughput}}{\text{total offered traffic}}. \tag{1}$$

By definition of Phase 1,

$$T_{\text{norm}} = 1 \tag{2}$$

in the phase 1 and starts decreasing as the system enters the phase 2. Thus, the critical point of the phase transition between phases 1 and 2 can be defined as the operational point where T_{norm} starts dropping below 1. Of course, in practice the value of T_{norm} in the phase 1 is not strictly equal to, but is fluctuating near unity. Therefore, we define the critical point of the phase transition between phases 1 and 2 to be the following

$$T_{\text{norm}} = 1 - \epsilon_1. \tag{3}$$

The value of ϵ_1 was empirically chosen and set to 0.01; a smaller or larger value can be applied depending on how strictly one wants that the phase 1 satisfies Equation 2. With $\epsilon_1 = 0.01$, if $T_{\text{norm}} \geq 0.99$ the network operates in the fully efficient phase (phase 1), where the aggregate throughput is approximately equal to the offered traffic; otherwise we are either in the phase 2 or in the phase 3. In statistical mechanics terms, every phase transition is characterized by an *order parameter*; the order parameter is a quantity which is zero in one phase and obtains a non-zero value as the system enters the new phase. For the phase transition between phases 1 and 2 we can define the following order parameter

$$m_1 = -T_{\text{norm}} + 1, \tag{4}$$

which is approximately equal to zero if the network is in the fully efficient regime, i.e., Phase 1 ($T_{\text{norm}} \geq 0.99$), and takes positive values as the system enters the phase 2.

For determining the critical point between phases 2 and 3 we consider the first differences[2] of the aggregate throughput,

$$T_{\text{diff}} = \text{ aggregate throughput}_i - \text{ aggregate throughput}_{i-1}. \tag{5}$$

In the saturated phase the throughput differences will be approximately equal to zero. However, in practice we need to take into account a tolerance margin, thus we define

$$T'_{\text{diff}} = \begin{cases} 0, & \text{if } T_{\text{diff}} < +\epsilon_2 \\ T_{\text{diff}}, & \text{otherwise} \end{cases} \tag{6}$$

If $T'_{\text{diff}} > 0$ the aggregate throughput still increases with the offered traffic. Otherwise, if $T'_{\text{diff}} = 0$, the network is in the saturated throughput phase. We would like to stress that since there is no established definition of how to specify a single operational point as the saturation point of a CSMA system, the determination of a value for ϵ_2 should be left as an empirically chosen decision for individuals. The determination of the order parameter for the second phase transition is more straightforward, that is

$$m_2 = T'_{\text{diff}}. \tag{7}$$

We would like, though, to point out that the number of different phases and their boundaries can be defined differently, depending on the operational points of the system one wants to stress on. For example, in the case of the CSMA/CA throughput another rational option would be to specify only two different states that correspond to non-saturated and saturated throughput regimes respectively. Additionally, the exact definition of the phase boundaries can be defined in a different way as well.

[2]The first differences are defined as the differences between consecutive samples, i.e., we subtract each value of the aggregate throughput from its succeeding sample.

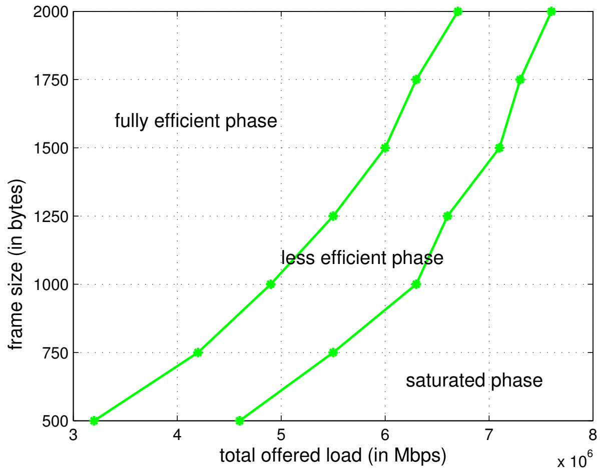

Fig. 2. The phase diagram of the CSMA/CA throughput with respect to the frame size and the total offered traffic load.

At this point we have all the information we need for drawing the corresponding phase diagram after simulating the network for all possible combinations of the state variable values, namely of the frame size and the total offered traffic. We consider a slotted CSMA/CA protocol with BEB. Simulations are performed using Qualnet network simulator. The network consists of 20 users that generate CBR (Constant Bit Rate) traffic. The channel bit rate is 11 Mbps. Specifically, we ran simulations for 7 different frame sizes (500, 750, 1000, 1250, 1500, 1750, 2000 bytes) and 133 different values of offered traffic (from 0.1 to 13.3 Mbps with steps of 0.1 Mbps), which yields a total of $7 \times 133 = 931$ simulation configurations. The resulting phase diagram is shown in Figure 2. The two lines on the diagram represent the phase boundaries and the diagram illustrates at which of the three phases our network operates for every combination of the state variables. The left area of the diagram corresponds to the fully efficient phase, where the condition $T_{\text{norm}} \geq 0.99$ is satisfied. The middle area is the less efficient phase, where $T_{\text{norm}} < 0.99$ and $T'_{\text{diff}} > 0$, and finally the right area corresponds to the saturated phase, where $T'_{\text{diff}} = 0$.

Up to this point we showed how a phase diagram can be created given that the outcome of the system is known for all combinations of the state variables. However, on its own, this is not particularly useful in practice; our objective is to be able to derive a phase diagram by means of a reduced number of simulation points and this will be presented in the following section.

III. Derivation of the Phase Diagram by means of a Support Vector Machine

In Section II we created the phase diagram of the CSMA/CA throughput after simulating the network for a set of 931 different simulation points, i.e, 931 combinations of the state variables. In this section we use SVMs to show how we can derive the same phase diagram from a significantly reduced number of simulation points. We note that in the present paper we employ SVMs as an efficient method for deriving phase diagrams without, however, precluding the suitability of other machine learning methods. For example, in [8] genetic algorithms are proposed for locating phase transitions in agent-based scenarios.

Support Vector Machines [9] belong to the class of supervised learning algorithms and have become a popular machine learning technique [10] for addressing classification problems. The task of a classification problem is to classify some objects into a number of classes (categories) based on a specified set of features. In supervised learning we use a training set, i.e., a set of objects whose class is known, in order to derive a classifier function that will be then used to map objects to classes. The objective of a SVM is to find the maximum margin hyperplane, that is the hyperplane that maximizes the distance to the closest training points of each class. In other words, the maximum distance hyperplane is the classifier function to which a SVM converges. SVMs can handle non linearly separable classes by applying the so-called kernel trick; the idea is to map the feature vectors through a kernel function into a higher dimensional feature space in which they become linearly separable.

For the derivation of a phase diagram the state variables of the system under study shall constitute the features of a classification task. In the context of the scenario we presented in Section II these are the frame size and the amount of offered traffic. Thus, a combination of a specific frame size and a specific amount of offered traffic is a so-called feature vector in a two-dimensional classification problem, denoted as

$$F = [f, t]^T, \quad (8)$$

where f is the value of the frame size and t the offered traffic.

Our aim is to approach the phase boundary lines. For each of them we apply a two-class SVM classifier. The objective is that the function to which the SVM will converge coincides with the phase boundary line. For the first phase transition (between phases 1 and 2) the one class corresponds to state variable combinations belonging to the phase 1 and the other class corresponds to the combinations belonging either to Phase 2 or 3. For the second phase transition (between phases 2 and 3) the one class corresponds to phases 1 and 2 and the other class corresponds to the phase 3. Since SVM classification belongs to the family of supervised learning methods we need to generate a set of training vectors, that is a set of feature vectors, of which we know their classes. The same set of training vectors can be used for the two SVM classifiers meaning that we shall assign two different class labels to each training vector; the first class label will apply for the first SVM classifier to find the boundary of the first phase transition and the second class label will apply for the second SVM classifier for the second phase transition. The training vectors are selected randomly and in order to classify them, i.e., to determine the two class labels for each training vector for the first and for the second classifier, we have to simulate the network for the corresponding parameter values f and t.

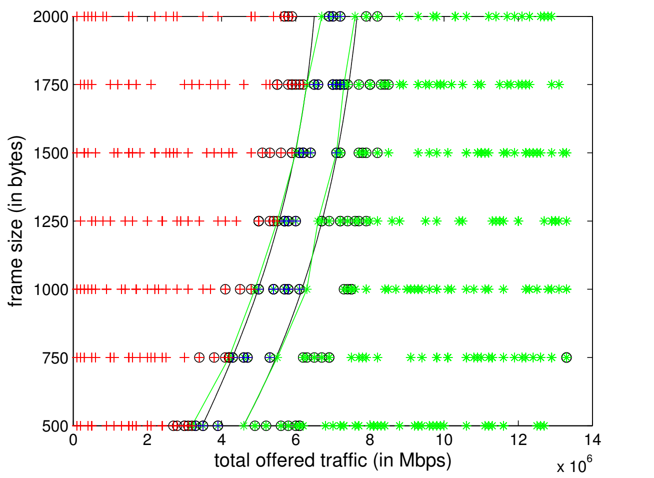

Fig. 3. The 2-dimensional phase diagram of the CSMA/CA throughput derived by means of SVMs. The coloured points correspond to the training vectors. The green phase transition boundaries are those resulting from a full simulation of the system, whereas the black lines are the boundaries obtained by the two SVMs.

In order to reduce further the number of simulation points we apply the following two rules. The first rule concerns the first SVM classifier; for a randomly selected training vector

$$F' = [f', t']^T,$$

if we already have another training vector, $F_1 = [f_1, t_1]^T$, with $f' = f_1$ and $t' < t_1$, and F_1 belongs to the class that corresponds to the phase 1, then we can directly classify F' to the same class without performing a simulation. For acting more on the safe side and avoid missclassifications of training vectors we will expect to have two training vectors, instead of one, that satisfy the conditions like F_1 before classifying F' without a simulation.

Similarly for the second classifier, for a training vector

$$\tilde{F} = [\tilde{f}, \tilde{c}]^T,$$

if we have already two (instead of one, as before) training vectors $F_2 = [f_2, t_2]^T$, with $\tilde{f} = f_2$ and $\tilde{t} > t_2$, and F_2 belongs to the class that corresponds to the phase 3, then we can directly classify $\tilde{F}$ to the same class.

The phase diagram obtained by means of the two SVM classifiers is illustrated in Figure 3. The coloured points correspond to the training vectors. The green phase transition boundaries are those resulting from a full simulation of all the 931 simulation points, whereas the black lines are the boundaries obtained by the two SVMs. The set of training vectors

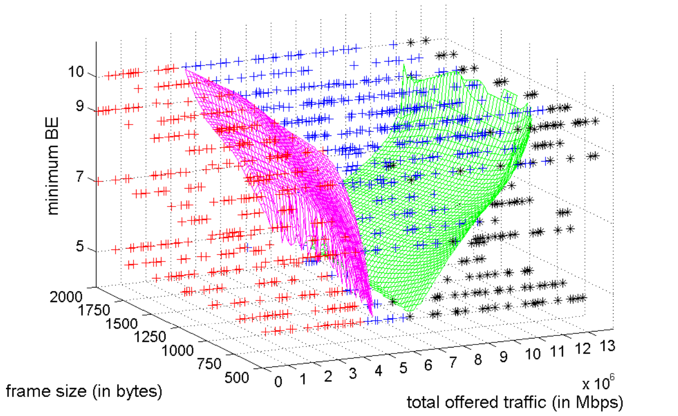

Fig. 4. The 3-dimensional phase diagram of the CSMA/CA throughput derived by means of SVMs.

we used was approximately the one half of the full set of 931 simulation points, but the number of configurations we had to simulate was significantly smaller due to the enhancement described in the two previous paragraphs. Specifically, the number of simulation points we have to run varies somewhere between the 25% and 30% of the full set of the 931 points.

Following exactly the same procedure we can increase the dimensions of the problem and derive, for example, a 3-dimensional phase diagram (Figure 4). Along with the frame size and the offered traffic the minimum value of the Backoff Exponent (BE) is added to the state variables that we take into account, and which comprise the feature vectors of the classification problem. In Figure 4 the green and magenta surfaces correspond to the phase boundaries resulting from the two SVM classification problems. Compared to the full simulation analysis (for 7 different frame sizes, 133 traffic loads and 4 minimum BEs) that would require to simulate 3724 different configurations, the phase diagram can be derived from approximately the 30% of the full simulation set.

IV. Conclusions

In this paper we considered the concept of phase diagrams within the context of wireless networking. More specifically, we employed phase diagrams to characterize a wireless network by means of a reduced number of simulation points in comparison to a full simulation study. Given the fact that sometime a full simulation analysis requires an unattractively large amount of computation time, a much faster derivation of the system's phase diagram can be an interesting alternative in certain practical cases.

A phase diagram provides a complete picture of the phase space; it illustrates in which state our system shall be for every possible combination of values the state variables can take. A phase diagram does not provide an outcome of the same accuracy level as a full simulation study, but it fully characterizes the system with respect to the points of state changes. The points where a system changes its state correspond to qualitative changes in the properties of the system, therefore there are points of significant interest and importance as far as the characterization and understanding of a system is concerned. We stress that our approach of characterizing wireless networks via phase diagrams is very well suited to cross-layer optimization objectives, in which case a phase diagram can be simply created with respect to state variables from different layers. In order to derive a phase diagram with an attractive computational effort, that is, a limited amount of simulations, we propose the application of SVMs.

As a case study we presented a simple CSMA/CA model where the aggregate throughput is the performance parameter of interest. The throughput with respect to the offered traffic can be divided into three phases: a fully efficient phase, a less efficient phase, and a saturated throughput phase. We generated the corresponding phase diagram and we showed how this task can be solved as an SVM classification problem.

Finally, we would like to point out that our suggestion to employ SVMs for deriving phase diagrams with an attractive computational effort is not limited to wireless networks, but can be used within the context of various scientific fields.

Acknowledgment

The authors would like to thank the RWTH Aachen University and the German Research Foundation (Deutsche Forschungsgemeinschaft, DFG) for financial support through the UMIC research center. We would also like to thank the European Union for providing partial funding of this work through the PURSUIT project.

References

[1] L. Reichl, *A Modern Course in Statistical Physics*. Wiley, 1998.

[2] M. Durvy, O. Dousse, and P. Thiran, "Border effects, fairness, and phase transition in large wireless networks," in *INFOCOM 2008. The 27th Conference on Computer Communications. IEEE*, Apr. 2008, pp. 601 –609.

[3] B. Krishnamachari, S. Wicker, and R. Bejar, "Phase transition phenomena in wireless ad hoc networks," in *Global Telecommunications Conference, 2001. GLOBECOM '01. IEEE*, vol. 5, 2001, pp. 2921 – 2925.

[4] B. Krishnamachari, S. Wicker, R. Bejar, and M. Pearlman, *Critical Density Thresholds in Distributed Wireless Networks*. USA: Kluwer Academic Publishers, 2003, pp. 279–296.

[5] F. Xing and W. Wang, "On the critical phase transition time of wireless multi-hop networks with random failures," in *Proc. of the 14th ACM international conference on Mobile computing and networking*, ser. MobiCom '08, 2008, pp. 175–186.

[6] M. Michalopoulou and P. Mähönen, "Addressing phase transitions in wireless networking optimization," in *7th International Workshop on Self-Organizing Systems (IWSOS 2013)*, May 2013.

[7] ——, "Game theory for wireless networking: Is a nash equilibrium always a desirable solution?" in *Personal Indoor and Mobile Radio Communications (PIMRC), 2012 IEEE 23rd International Symposium on*, sept. 2012, pp. 1249 –1255.

[8] F. Stonedahl, "Genetic algorithms for the exploration of parameter spaces in agent-based models," Ph.D. dissertation, Northwestern University, Evanston, Illinois, 2011.

[9] S. Abe, *Support Vector Machines for Pattern Classification (Advances in Pattern Recognition)*. Springer, 2010.

[10] E. Alpaydin, *Introduction to Machine Learning*. The MIT Press, 2004.