OMAN: A Mobile Ad Hoc Network Design System

Lex Fridman, *Student Member*, *IEEE*, Steven Weber, *Senior Member*, *IEEE*, Charles Graff, *Member*, *IEEE*, David E. Breen, *Member*, *IEEE*, Kapil R. Dandekar, *Senior Member*, *IEEE*, and Moshe Kam, *Fellow*, *IEEE*

Abstract—We present a software library that aids in the design of mobile ad hoc networks (MANET). The OMAN design engine works by taking a specification of network requirements and objectives, and allocates resources which satisfy the input constraints and maximize the communication performance objective. The tool is used to explore networking design options and challenges, including: power control, adaptive modulation, flow control, scheduling, mobility, uncertainty in channel models, and cross-layer design. The unaddressed niche which OMAN seeks to fill is the general framework for optimization of any network resource, under arbitrary constraints, and with any selection of multiple objectives. While simulation is an important part of measuring the effectiveness of implemented optimization techniques, the novelty and focus of OMAN is on proposing novel network design algorithms, aggregating existing approaches, and providing a general framework for a network designer to test out new proposed resource allocation methods. In this paper, we present a high-level view of the OMAN architecture, review specific mathematical models used in the network representation, and show how OMAN is used to evaluate tradeoffs in MANET design. Specifically, we cover three case studies of optimization. The first case is robust power control under uncertain channel information for a single physical layer snapshot. The second case is scheduling with the availability of directional radiation patterns. The third case is optimizing topology through movement planning of relay nodes.

Index Terms—Cross-layer design, optimization, mobile ad hoc networks, simulation, software systems.

1 Introduction

Network design, like other broad optimization fields, is often treated as a collection of distinct problems with significant effort focused on finding solutions to each of those individual problems. In developing the Optimization for Ad hoc Networks (OMAN) system, our main goal is to bring together network resource allocation problems and the methods for solving them into a cohesive, extendable, object-oriented software package with a standard application programming interface (API) and graphical user interface (GUI) for software developers and network design engineers, respectively.

In OMAN, network design is approached as a process of optimizing variables in a well-defined scenario across time, space, and the communication network stack. Fig. 1 shows the high-level data flow in OMAN. Inputs are the scenario specification and optimization variables and objectives, as well as an extensive set of parameters defining every aspect of the network model, simulation, and optimization. In other words, the system input is the definition of the design problem and a list of directives on how OMAN should find a good solution to this problem. The output of the system is a set of optimal or locally optimal resource allocations and an estimate of the performance expected when the network operates under those design decisions. OMAN defines a portfolio of objectives which include, among others, power, capacity, connectivity, survivability, delay, fairness, load balancing, throughput, and redundancy [1]. These and several other objectives are defined explicitly in the case studies presented in this paper.

The optimization of network parameters in OMAN is a feedback process of optimization and performance estimation through simulation. The first step is to find an optimization method most appropriate to the set of control variables and objectives provided as input. If no specialized algorithm is available in OMAN for the specified problem, then the problem is formulated as a mathematical program in the AMPL modeling language [2] as shown in Fig. 2. An appropriate generic solver is then used to solve the program, depending on whether objectives and constraints are linear or nonlinear, and whether the variables are discrete or continuous. The power control problem of minimizing power under a signal-to-noise-interference constraint is an example of a linear program which is optimized using this generic solver approach. If a specialized method is available for the problem, OMAN automatically uses it to find a solution. An example of a specialized method is a heuristic packing procedure which schedules a set of concurrent transmissions that maximize sum-rate capacity and ensures a chance for every node to transmit at least once. Both these examples are described in further detail in this paper.

The rest of this paper is organized as follows: Section 2 covers tools which share a degree of functionality with OMAN. Section 3 broadly covers the software system

- *L. Fridman, S. Weber, K.R. Dandekar, and M. Kam are with the Department of Electrical and Computer Engineering, Drexel University.*
- *C. Graff is with the US Army RDECOM, Fort Monmouth, NJ.*
- *D.E. Breen is with the Department of Computer Science, Drexel University.*

Manuscript received 22 Jan. 2010; revised 15 Mar. 2011; accepted 10 June 2011; published online 17 May 2012.
For information on obtaining reprints of this article, please send e-mail to: tmc@computer.org, and reference IEEECS Log Number TMC-2010-01-0038. Digital Object Identifier no. 10.1109/TMC.2011.174.

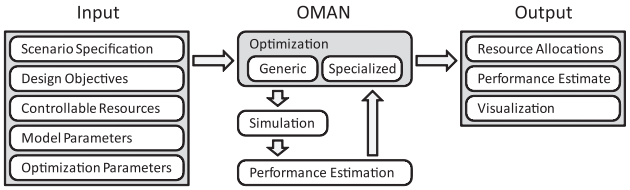


Fig. 1. High-level data flow in OMAN. The system's input is the design problem to be solved. The system's output is the solution. The optimization process is a feedback loop of optimization and performance estimation through simulation.

architecture and design of OMAN. Section 4 describes the aspects of OMAN's network model which are relevant to the three examples covered in this paper. Section 5 presents two cases of validation with measured testbed data. Sections 6-8 describe three case studies of network design problems investigated and solved using OMAN.

2 Related Work

The generally accepted network design cycle consists of three steps: 1) developing a network model, 2) estimating the performance of the proposed network through simulation, 3) manually tweaking the model parameters until an acceptable performance is achieved for the target set of scenarios. The estimation is usually done with network-specific discrete event simulators such as Qualnet [3] or Opnet [4]. In addition, existing design tools aid in a well-defined specialized aspect of the resource allocation problem that is part of step 3. For example, Motorola Enterprise Planner [5] aims to help a network designer build indoor wireless networks. This tool is used to show the designer the best locations for access points by using predictive algorithms and propagation modeling based on geographic topology.

In the second and third steps of the traditional design cycle, the performance of a proposed design is analyzed through simulation, and changes to the model are made until the model behaves as desired. However, due to the complexity of networks and the large number of design parameters, changes in the design of a model may have unintended effects which must then be accounted for. To alleviate the timely process of designing and redesigning a network, we intend to apply optimization theory to generalized networking models in order to efficiently design MANETs. OMAN brings together a large portfolio of novel and existing optimization techniques together with

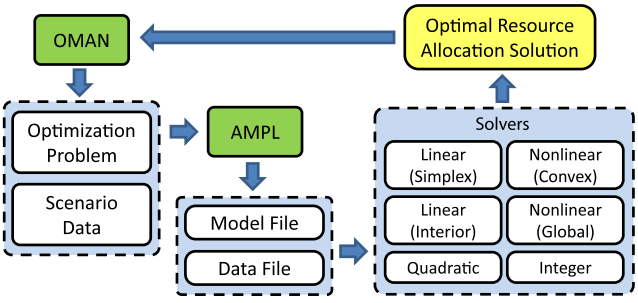


Fig. 2. The data flow of the interaction between OMAN and the generic numerical optimization problem solvers through the AMPL modeling language.

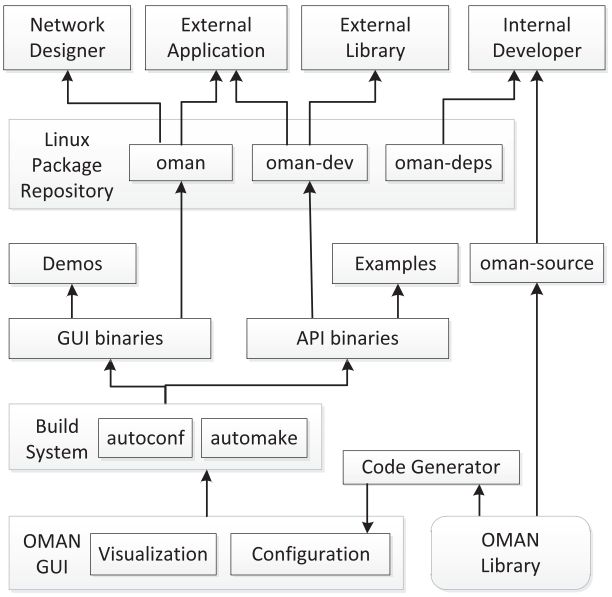


Fig. 3. The OMAN API and GUI: from source code to binary form to packaged form to four classes of users.

the simulation capabilities of such tools as Opnet for the task of performance estimation. The focus on the rich interplay of these two tasks of design and simulation is one of the main novelties of our system. Specifically, OMAN automates step 3 of the cycle allowing the designer to control high-level objectives instead of controlling low-level decision variables.

3 Software Architecture

3.1 Two Levels of User Interface

OMAN has two distinct forms: 1) an application with a graphical user interface and 2) a library with an application programming interface as shown in Fig. 3. The former is an interface for human users (e.g., network design engineers), while the latter is an interface for other programs that link against it.

This paper presents three case studies that are part of the "Examples" codebase in Fig. 3. There are currently over 200 such examples. Each one is no more than 100 lines of code that provide the input data as shown in Fig. 1. The rest is handled by OMAN through its API without the user having to know any implementation details of the simulation and optimization algorithms. For example, an external developer with a detailed Opnet model may wish to optimally allocate resources within the framework of their Opnet-defined scenario. They can access the resource-allocation functionality of OMAN through the API (after converting the Opnet network data structures to those of OMAN).

One of the goals of OMAN as a network design tool is to provide a mechanism for comparing network technologies, i.e., radios, MAC protocols, routing protocols, and battery models. Each such model or algorithm is implemented in OMAN in a modular way such that it can be swapped out for any number of alternatives. The GUI provides a streamlined way of configuring multiple alternatives, and comparing them through concurrent simulation and optimization. An "internal developer" can extend OMAN by providing an implementation of a new alternative model or

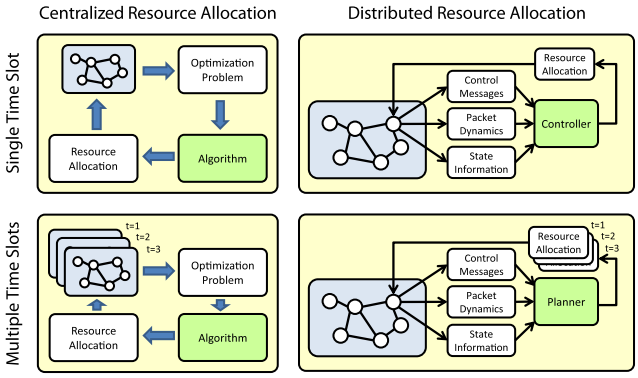


Fig. 4. The four network design paradigms under which OMAN operates. The two distinguishing characteristics are 1) whether the resource allocation has a temporal dimension and 2) whether the optimization is performed in a centralized or distributed fashion.

algorithm. The API supports such an extension without needing to be modified. The developer only has to add a set of control parameters associated with the new extension. These parameters are then automatically added to the GUI through active generative programming [6].

3.2 Network Design Paradigms

The resources which are to be efficiently allocated on an ad hoc wireless network are naturally distributed, residing either on the nodes or the edges of the graphs that represent the network state. The algorithms in OMAN are separated into two categories: 1) centralized (network centric) and 2) distributed (node centric). The former operates on single "snapshots" or on a time-averaged model of the global network state. The latter operates as a control mechanism on the node. To allow for asynchronous control decisions on the nodes, OMAN provides a full-fledged discrete event simulator [7].

Fig. 4 shows two examples each of design for the network-centric and node-centric approaches. The static and time-averaged paradigms view the network as a whole, and efficient resource allocation as a global decision process. The event-based reactive controller and robust planner move the decision process to each individual node. The distinction within each pair is whether the result of the optimization has a temporal dimension.

The distinction between distributed versus centralized and event-based versus slotted algorithms is orthogonal to another critical parameter: the completeness, certainty, and reliability of information available to a node or network in performing the efficient resource allocation. The network model and the discrete event simulator operating on the model have perfect information about the past, present, and future of network state. The degree to which that information is available to the nodes is controlled by the "information filter" shown in Fig. 5. In other words, the information under which decisions are made is a strict subset of the information which drives the simulator.

3.3 Data Structures and Algorithms

OMAN can be logically separated into three types of classes: those that store data, those that process data, and those that integrate the two into a cohesive simulation and optimization system. Specifically, the three categories of classes are:

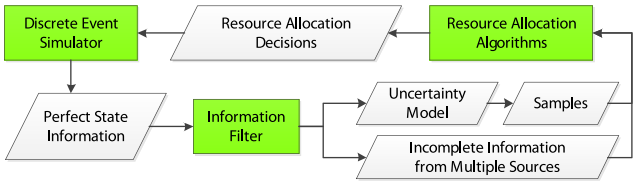


Fig. 5. The process of modeling the uncertainty, incompleteness, and unreliability of information when making resource allocation decisions.

1. **Data structures.** Objects such as the graphs in Section 4.1 which represent the current and past states of the network, including resource allocation decisions, packet transmissions, channel conditions, movement paths, etc.
2. **Algorithms.** A modular collection of encapsulated functions which process the object data. These include classes that optimize the resource allocation decisions across the network stack, visualize the network state, convert, aggregate, integrate available information into a form accepted by other algorithms.
3. **Directors.** The "glue" classes that control which algorithm gets executed when and on what data. Depending on the network design paradigm, this involves everything from a simple class that evolves the position of the nodes as some aspect of the network is optimized to a full-blown event-based simulator which controls every possible decision and change affecting the network state using an arbitrarily expandable set of events.

4 Network Model Overview

4.1 Capacitated Graph Construction

For the three case studies in this paper, we use the PHY and MAC model abstraction similar to the one used in [8]. Specifically, scheduling is done in the time domain, where the transmissions are synchronized. Each schedule graph $G_P = (V_P, E_P)$ is a choice of transmitters with associated powers, and the set of schedules form a period of concurrent slotted transmissions which are looped continuously. We use a random packing procedure that picks a set of schedules that 1) guarantee each node at least one chance to transmit and 2) guarantee concurrent transmitters each succeed (i.e., the signal-to-interference-plus-noise ratio (SINR) seen at each receiver is acceptably high). A capacitated graph $G_F = (V_F, E_F)$ is formed by multiplexing these schedules together. That is, if in one of the schedules the node i can successfully communicate with node j, edge (i, j) is added to G_F with a capacity corresponding to the sum of capacities on each such feasible instance of communication.

The control parameters of the PHY graph are 1) the power P_i on each node i and 2) the minimum acceptable signal-to-interference-plus-noise ratio γ which defines the quality-of-service requirement for the channels. Given these values, the random packing procedure adds transmitters to a time slot incrementally such that the SINR requirement is not violated for any of the links in E_P. SINR is computed by

$$\mathrm{SINR}_{ij} = \frac{P_i H_{ij}}{\sum_{x \in \mathcal{I}_j^k} P_x H_{xj} + \eta}, (i, j) \in E_P, \qquad (1)$$

where SINR_{ij} is the SINR on the channel between transmitter i and receiver j, η is the channel noise power which is assumed to be receiver independent. H_{ij} is the channel between transmitter i and receiver j defined as $H_{ij}=(\frac{d_{\text{ref}}}{d_{ij}})^{\alpha}$, where d_{ij} is the distance between nodes i and j, d_{ref} is a channel reference distance, and α is the attenuation exponent. $\mathcal{I}_j^k$ is the set of k nearest interfering transmitters to receiver j. In Section 6, we look at the effect of a small interference neighborhood on the optimality robustness of the returned solution.

The result of the packing procedure is a collection of graphs $\mathbf{G}_P \equiv \{G_P^1, G_P^2, \ldots, G_P^m\}$. All edges (i,j) in graphs of $\mathbf{G}_P$ satisfy the requirement that $\text{SINR}_{ij} > \gamma$. The capacity on each of those edges is assigned in one of two ways: 1) capacity of 1 on all feasible edges, or 2) SINR-dependent Shannon capacity defined as follows:

$$C_{ij}\left(G_P^z\right) = B_{ij}\log_2\left(1+\text{SINR}_{ij}\left(G_P^z\right)\right), \tag{2}$$

where B_{ij} is the bandwidth on edge (i,j) which is assumed to be 1, effectively setting capacity to the spectral efficiency of the channel, C_{ij} is the capacity on edge (i,j), and $z = 1,2,\ldots,|\mathbf{G}_P|$ is the index of the physical graph. Note that C and SINR are shown as functions of a specific graph. This is used to distinguish the capacities and SINRs on individual physical graphs from the single multiplexed flow graph. Unless otherwise specified, capacity C_{ij} refers to the flow graph capacity $C_{ij}(G_F)$.

The multiplexing procedure is additive for capacity, that is, the capacity of an edge (i,j) in the flow graph G_F is computed from $\mathbf{G}_P$ by

$$C_{ij}(G_F) = \sum_{z:(i,j)\in E_P^z} C_{ij}(G_P^z). \tag{3}$$

The result of the multiplexing procedure is a capacitated flow graph $G_F = (V_F, E_F)$ that has an edge $(i,j) \in E_F$ if and only if $C_{ij} > 0$.

4.2 Multicommodity Flow

In the time-averaged design paradigm, we adopt the standard multicommodity-flow abstraction of routing [9]. Let there be K commodities, each commodity specified by a source node $\sigma_k \in V_F$, and a destination node $\delta_k \in V_F$. A commodity specifies a stream of unique data from node σ_k to node δ_k. A *throughput vector* $\mathbf{f} = \{f_1,\ldots,f_K\}$ is *feasible* if it respects both network capacity constraints and conservation of flow constraints. The objective is to maximize the sum throughput summed over the K commodities. Each commodity's throughput may be split across multiple paths; we define a *flow* as the set $\{x_e^k, e \in E_F, k \in [K]\}$, so that x_e^k is the flow for commodity k on edge e. The max multicommodity flow problem is

$$\begin{aligned}
\max_{\mathbf{x}} \quad & F(\mathbf{f}) = \sum_{k=1}^{K} f_k \\
\text{s.t.} \quad & \sum_{i\in V_F} x_{i,\delta_k}^k = f_k, k \in [K] \\
& \sum_{k\in[K]} x_{ij}^k \le C_{ij}, i \in V_F, j \in V_F \\
& \sum_{j\in V_F} x_{ji}^k = \sum_{j\in V_F} x_{ij}^k, k \in [K], i \in V_F.
\end{aligned} \tag{4}$$

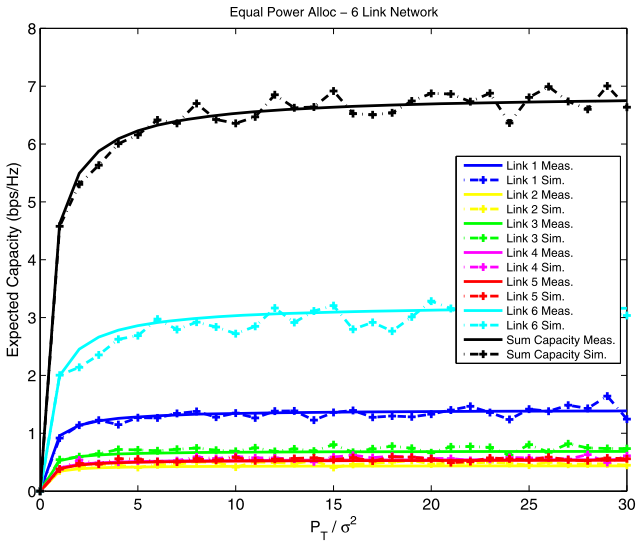


Fig. 6. Validation of MIMO equal power allocation expected capacities for a six link topology.

The first line states the objective is to maximize the sum commodity throughput. The first constraint defines the throughput for each commodity as the sum of the flow over all edges which terminate in commodity k's terminal node δ_k. The second constraint is the capacity constraint; the third is the conservation of flow constraint.

In the following sections, we detail three case studies of optimization over the mobility, PHY, MAC, and transport layers described in this section. The case studies build on each other, each one using the optimization performed in the previous one.

5 Validation of Simulated Results

The network models in OMAN vary in the resolution of their abstractions and the restrictiveness of their constraints. The benefit of these abstractions in simulation is the tractable interpolation of resource allocation performance, often in real time. Their cost is the potential inaccuracy of those interpolations as predictors of real-world system performance. For this reason, a major part of the effort in building OMAN is validating simulation results with measured testbed data. We briefly present two cases of validation at the PHY layer and the MAC layer, respectively.

5.1 Multiple Input Multiple Output (MIMO)

Fig. 6 shows how MIMO communication techniques in OMAN have been validated experimentally using software defined radio (SDR) testbed measurements. The measured topology consisted of a six link, point-to-point network measured using the WARP SDR platform [10] in the 2.4 GHz band in an indoor campus building environment. A two element array of whip antennas with an interelement spacing of 3λ was used at every node. Each of the transmitters allocated power equally to each of its antennas without the use of closed-loop communication techniques to feedback channel information. The same network topology was simulated in OMAN. The MIMO matrix channel in OMAN was simulated using the full correlation channel modeling technique [11], where the correlation matrix of each link was estimated from measured data. The

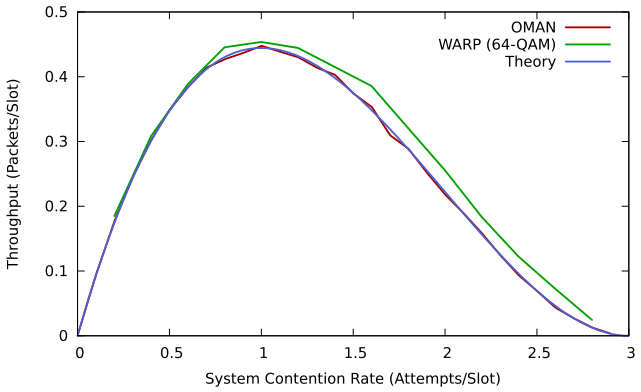


Fig. 7. Slotted Aloha normalized throughput comparison of theory, OMAN simulation, and WARP testbed measurement for a 3-user uplink topology.

figure shows good agreement between the measured and simulated shannon capacity for each of the links individually and for the sum capacity of the network.

5.2 Aloha

Fig. 7 shows the throughput in packets per time slot under slotted Aloha for three transmitters as a function of the system contention rate (defined as Np for $N = 3$ users and p the per-user contention probability). The figure shows the theoretical throughput under the channel collision model (error free reception iff a transmitter is the sole transmitter in the slot), the measured throughput using the WARP testbed when the nodes employ 64-QAM, and the simulated throughput under OMAN when the nodes operate in the collision channel model. This experiment validates the OMAN implementation of Aloha as matching both theoretical and experimental results.

6 Case 1: Robust Power Control (RPC)

6.1 Overview

Allocation of physical resources (e.g., transmission power) based on knowledge of the network state is often complicated by the presence of uncertainty in the available information. Therefore, when the characteristics of the wireless propagation channel are highly dynamic or only noisy measurements are available, OMAN represents the uncertainty as a collection of S samples of each channel state H_{ij} in (1) which represent a range of values that each channel between a transmitter and a receiver can take on.

The problem of optimally allocating resources under such a statistical representation of the channels can be solved in OMAN by assuming the distribution mean for each channel state, or by using a Robust Optimization (RO) method which seeks to quantify the dependability of the resource allocation solution. A fundamental problem in RO is the tradeoff between feasibility and optimality. RO may be interpreted to be a multiobjective optimization problem with two objectives: maintain feasibility and seek optimality. With this view in mind, a Pareto front [12] can be constructed to demonstrate the tradeoff between the two objectives. A network designer then only provides OMAN with 1) the requirement of sufficiently high feasibility or 2) a ceiling for the transmission power on the network. An in-depth look at application of robust optimization to power control in OMAN is done in [13].

6.2 Channel Model

The inherent sources of uncertainty in the channel are due to the combination of reflection, diffraction, and scattering mechanisms of electromagnetic propagation. The relative importance of these mechanisms is a function of the propagation environment. These three mechanisms lead to three nearly independent phenomena: path loss, slow log-normal shadowing, and fast multipath fading. An additional major source of uncertainty is uncertainty in position, due either to noise in GPS measurements or random mobility of the nodes. We consider the latter source in this case study.

Using standard link budget analysis in wireless communication design, it is safe to approximate that an attempted transmission is successful provided the signal-to-interference-plus-noise ratio seen at the intended receiver exceeds some fixed threshold (corresponding to receiver sensitivity). We assume all receivers share a common SINR threshold requirement of γ. The SINR measured by a receiver is computed by (1).

The power control solution $\mathbf{P} = \{P_i | i \in \mathcal{T}\}$ is *feasible* for a given realization of the channels and the noise terms if

$$P_i \geq 0, \quad \text{SINR}_{ij} \geq \gamma, \quad i \in \mathcal{T}, (i,j) \in E_P, \tag{5}$$

where $\mathcal{T}$ is the set of active transmitters.

6.3 Robust Power Control

Mulvey et al. propose the following linear program in their 1995 paper [14] as a canonical optimization problem incorporating uncertainty. It is natural to seek power control solutions that minimize the sum power, while also maintaining feasibility. When channel conditions are unknown, the transmission powers must be selected a priori. This corresponds to making the transmission powers design decision variables.

For a single sample of the channel state $s \in \{1, \ldots, S\}$, we first define an indicator function F_{ij}^s which is 1 if the SINR_{ij} constraint is violated and 0 otherwise

$$F_{ij}^s = I\big(\text{SINR}_{ij}^s < \gamma\big), (i,j) \in E_P. \tag{6}$$

In other words, F_{ij}^s is the penalty or "failure" function for channel (i,j). The probability that channel (i,j) is feasible can then be computed from the set of channel samples by

$$\begin{aligned} \mathbb{P}(\text{SINR}_{ij} < \gamma) &= \mathbb{E}[I(\text{SINR}_{ij} < \gamma)] \\ &= \lim_{S \to \infty} \frac{1}{S} \sum_{s=1}^{S} F_{ij}^s, \end{aligned} \tag{7}$$

as $S \to \infty$ by the law of large numbers.

As previously stated, the robust power control problem has two objectives: 1) minimize power and 2) minimize the probability of infeasibility

$$\begin{aligned} \min_{\mathbf{P}}: \ & \sum_{i \in \mathcal{T}} P_i + \omega \sum_{(i,j) \in E_P} \sum_{s=1}^{S} F_{ij}^s \\ \text{s.t.}: \ & P_i \geq 0, i \in \mathcal{T} \\ & P_i \leq P_{\max}, i \in \mathcal{T}, \end{aligned} \tag{8}$$

where ω is a constant "weight" determining the importance of feasible SINR relative to low transmission power, $\mathcal{T}$ is the set of transmitters, and $P_{\max}$ is the maximum power that can be allocated to a transmitter.

Intuitively, the cost of uncertainty is increasing in ω since large ω yields robust solutions that minimize the chance of insufficient SINR through conservative (large) allocations of power.

The solution to (8) is a set of transmit powers which optimize the tradeoff between the two objectives of the problem for a given ω. In order to form a set of Pareto optimal solution, (8) is solved for a range of ω values, each solution providing a point on the Pareto front whose x-axis is the total transmit power and the y-axis is the probability of infeasibility. It should be noted that the optimization problem in (8) is nonlinear due to the nonlinearity of function F_{ij}^{s}. Therefore, the allocation of powers is not guaranteed to be optimal, but it is guaranteed to be Pareto efficient relative to the other computed solutions because that is one of the requirements for it to be included in the returned solution set.

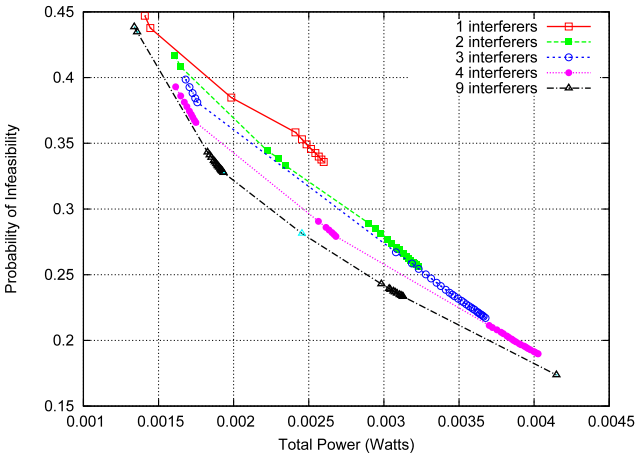


Fig. 8. Tradeoff between the feasibility objective (minimizing probability of violating the SINR requirement) and the optimality objective (minimizing the total transmit power). The bottom curve is returned by OMAN when the interference neighborhood for each receiver includes *all* the active transmitters in the network. The other four curves show the effect on the Pareto front of decreasing the interference set size.

6.4 Simulation and Results

The Pareto front formed from solving the RPC problem allows the network designer to choose an operating point based on the prioritization of the two objectives: transmit power and channel feasibility. We first look at a single mobile network where the uncertainty in the channel state (represented by the set of samples of each channel) comes from the changing topology due to the movement of the nodes. We then look at the effect of considering only a fraction of nearest interferers at each active receiver.

6.4.1 Pareto Optimal Tradeoff

The simulation setup is a mobile network of 100 nodes on a $1\ \mathrm{km}^2$ square arena. The nodes are placed uniformly at random. The nodes then move under a random waypoint model [15] at a speed of 2 m/s for a duration of 1 second during which the channel sampling process is performed. The transmission packing procedure is performed once and results in 10 unicast transmission-receiver pairs.

The interference set $\mathcal{I}^{\infty}$ in (1) is used for computing the SINR. In other words, in this case, every active transmitter is defined in the optimization problem as a potential source of interference. A single simulation run for 100 nodes with a full interference set takes approximately 1 second to execute on a 2.4 GHz processor.

The bottom curve in Fig. 8 shows a Pareto front of solutions produced by OMAN. Given the network topology, this solution set provides the network designer a range of optimal transmission power allocations. That is, attached to each point x on the x-axis is an allocation set $\mathbf{P} = \{P_i | i \in \mathcal{T}\}$ s.t. $\sum_{i \in \mathcal{T}} P_i = x$. The designer can then choose one of these solutions based on the relative value of the power objective versus the feasibility objective.

6.4.2 Effect of Shrinking the Interference Set

In the previous section, we looked at a single network whose topology is changing due to the mobility of the nodes. The set of channel state samples is collected and optimized over for that single network, producing a Pareto front of solutions. In this section, we consider the effect of k in the interference set $\mathcal{I}_j^k$. This set, as described in Section 4.1, is the set of k closest interferers to active receiver j. Fig. 8 shows the effect of decreasing k from the maximum of 9 to 1 for the single network described in the previous section. This plot provides an intuition that increasing k has diminishing returns. This pattern is looked at more closely in this section.

Note the distinction between the interference set $\mathcal{I}^k$ used in computing SINR in the optimization process and the interference set used to generate an estimate of probability of infeasibility for the two plots in this section. In the latter case, we always use $k = |\mathcal{T}| - 1$. That is, for the plotted values, every potential interferer is considered. However, for the objective function in the optimization problem, a more limited interference set is used.

The importance of keeping k small and independent of network size is twofold. First, a small constant k significantly reduces the complexity of the optimization problem as the size of the SINR computation in (1) no longer depends on the number of active transmitters, and thus is independent of network size. Second, a constant k removes the need for every transmitter-receiver pair to have channel state information from every other interfering transmitter on the network to this pair's receiver. Therefore, the significant overhead of sharing this information between nodes is removed, allowing for distributed power control approaches to make resource allocation decisions without first gathering channel state information from the whole network. The robust optimization framework we present in this paper is solved with a centralized method; however, it can just as well be solved with a distributed approach as in [16].

Figs. 8 and 9 show a decreasing probability of infeasibility as k increases. This is due to the fact that when $k < |\mathcal{T}| - 1$, not all interferers are considered, and thus the SINR on each edge is overestimated. As a result, the optimization returns a lower power than it would for $k = |\mathcal{T}| - 1$.

We consider a set of networks, ranging from 100 to 500 nodes, at 10-node increments, and repeat the same

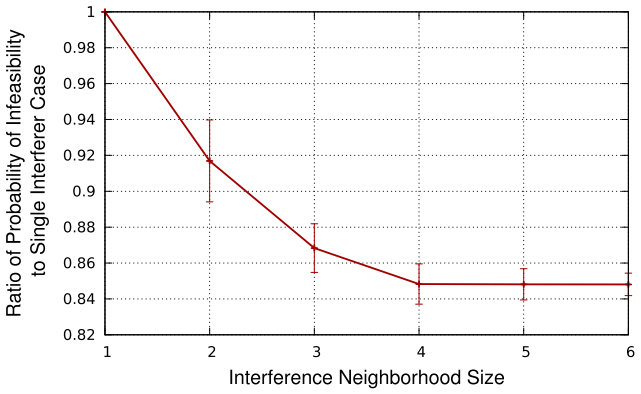


Fig. 9. The diminishing returns of increasing the size of the interference set beyond four closest interferers. The y-axis shows the ratio of the probability of infeasibility to the base case of an interference set of size 1 (just the one closest interferer). The error bars designate the standard deviation for the 4,000 simulation runs from which the plot is formed.

mobility simulation as described in the previous section. For each network size, we generate 100 different initial topologies. So, we run 4,000 simulations, generating for each a set of Pareto fronts like the ones shown in Fig. 8. From each of those sets of Pareto fronts, we compute the ratio of probability infeasibility to the base case of a size 1 interference set. Fig. 9 shows the average of those ratios over the 4,000 simulations. As the size of the neighborhood set increases, the set of Pareto-optimal solutions improves, converging on average to the best possible Pareto front at an interference set size of 4. In other words, this empirical result indicates that only a small number of nearest interfering transmitters have a significant effect on the feasibility of a channel.

7 Case 2: Scheduling with Directional Antennas

7.1 Overview

Physical resource allocation, an example of which is robust power control (see Section 6), generates a set of transmissions with transmit power that satisfies a network state objective. Extending this process over time, in order to optimize a higher level objective (i.e., throughput, survivability, etc.), a set of transmission *events* are generated which have a time component. This process is referred to as scheduling. This is an active area of research, especially in relation to joint optimization with other layers of the network stack [17]. In this scenario, we consider a heuristic approach to scheduling using directional antennas. This case study serves as an overview of one of the scheduling models in OMAN, in particular, centralized slotted scheduling. We show that the scheduling process can incorporate any PHY layer model, and as an example use directional antennas.

7.2 Schedule Model

At the lowest level, the OMAN network model is event based; that is, it is defined not by centralized global graph structures but by local events. However, the optimization engine, in most cases, operates on a higher level model of the network which extracts "snapshots" of the network state in one of two ways: 1) by querying the state at fixed time intervals or 2) by querying the state at any point that an event changes the state. The latter model is one that is often employed in event-based simulators, and is usually very data intensive. Therefore, the more efficient optimization techniques (e.g., nominal power control) are applied to this model, and the former querying technique is used for more complex optimization problems (e.g., robust power control).

The most commonly used event, which is relevant for the case in this section, is the *transmission* event, which specifies the transmitter, intended receivers, transmit power, the duration, and the antenna array excitation vector at the transmitter and receiver. In OMAN, a node can have any number of antennas, of which there are three types: 1) omnidirectional, 2) directional, and 3) MIMO antenna array. The first type radiates an equal amount of energy in all directions from the transmitter. The second type is defined by a 3D radiation pattern specified for both the transmitter and receiver which defines the amount of gain on the transmission signal depending on the relative positions of the transmitter and receiver. A MIMO system makes use of antenna arrays at both the transmitter and the receiver. OMAN models MIMO matrix channels using the full correlation channel modeling technique [11] which encapsulates transmit and receive array radiation patterns and channel spatial characteristics.

The set of transmission events form a schedule which when queried can be represented as a sequence of physical graphs as described in Sections 3.3 and 6. The optimization engine is used on each individual graph for optimal physical resource allocation. In this case, we consider the minimization of power as an objective; therefore, the nominal power control problem is solved for each physical graph. The result is a set of minimum-power transmissions which satisfy the feasibility requirements of the channel. In this scenario, channel feasibility is defined by a minimum SINR threshold defined at every node which serves as a receiver for a nonzero duration during the schedule.

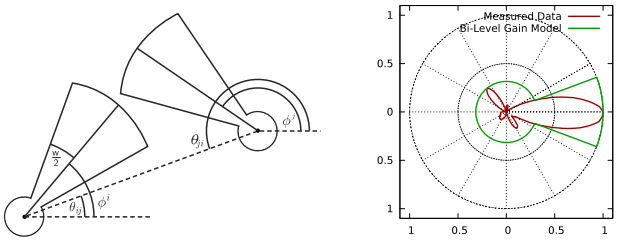
7.3 Directional Antenna Model

In this case study of scheduling, we describe an extension to the omnidirectional antenna model described in Section 4.1. We consider a smart array model, where an antenna can alter its radiation pattern in time. Each achievable and applicable radiation pattern we refer to as a *mode*. The general 3D radiation pattern is a continuous function parameterized by two variables that specify direction. However, for this case study, we assume a bilevel abstraction of the more general radiation pattern defined by the direction of the beam ϕ and the beam's width w. The radiation pattern, therefore, is a step function

$$R(\theta) = \begin{cases} R_H, & \text{if } |\phi - \theta| \leq \frac{w}{2} \\ R_L, & \text{otherwise,} \end{cases} \tag{9}$$

where R_H and R_L are the high and low field-strength, respectively, of the single lobe radiation pattern. A diagram of this bilevel model is shown in Fig. 10a.

The justification for using the simple bilevel model in this case study, as opposed to a more realistic one based on measured data as shown in Fig. 10b, is not implementation



(a) An example diagram for the bilevel model of the channel between node i (left) and j (right). In this case, the transmitter gain and receiver gain are both R_L since θ_{ij} and θ_{ji} are outside the R_H lobes of their respective radiation patterns.

(b) Comparison of a measured azimuth radiation pattern for a leaky wave antenna [18] with a conservative bilevel abstraction of that pattern.

Fig. 10. Radiation pattern of the bilevel antenna model and the leaky wave antenna measured data.

related. OMAN provides a large portfolio of realistic antenna models. The reason for using the bilevel model here is that it provides for a clean parametrized gradient from a pattern that causes no interference on the rest of the network to one that causes as much interference as the omnidirectional antenna. This is discussed further in Section 7.5.

The set of graphs $\mathbf{G}_P$, as described in Section 4.1, that define the schedule, thus also define the radiation pattern for each antenna. The SINR on each channel of the physical graph, as given in (1), can be extended to include the transmitter and receiver radiation patterns

$$\text{SINR}_{ij}^z = \frac{P_i H_{ij} R_i^z(\theta_{ij}) R_j^z(\theta_{ji})}{\sum_{x \in \mathcal{I}_j^k} P_x H_{xj} R_x^z(\theta_{xj}) R_j^z(\theta_{jx}) + \eta}, \qquad (10)$$

where $R_i^z(\theta)$ is the radiation pattern of node i in time slot z in the direction of θ. θ_{ij} is the direction of the vector from node i to node j. In other words, for the channel (i, j), $R_i^z(\theta_{ij})$ is the transmitter radiation pattern, and $R_j^z(\theta_{ji})$ is the receiver radiation pattern. The algorithm for choosing which mode is best for a particular channel is described in the next section.

For this case study, we assume that the antenna can choose the direction ϕ of the beam defined in (9). The INITIALIZERADIATIONPATTERNS heuristic procedure of Algorithm 1 determines ϕ_m^i for each of the M_i modes on node i by pointing the first beam to the farthest receiver, the second beam to the second farthest receiver, etc. A "mode" is a unique direction for the peak of the radiation pattern. In this case, distance serves as a heuristic estimate for the quality of the channel. Thus, the channels which may require the largest transmit power to achieve a reasonable SINR are assigned with the unidirectional radiation pattern that limits the fraction of the arena to which this transmission causes interference.

Algorithm 1. PACKTRANSMISSIONS$(G_C, \mathbf{H}, \mathbf{M}, \boldsymbol{\gamma})$ where G_C is the desired connectivity graph, $\mathbf{H}$ is the channel state for all possible edges in the network, and $\mathbf{M}$ is the vector specifying the number of antenna modes available at each node, and $\boldsymbol{\gamma}$ is the vector of minimum SINR requirements on each node.

Require: $G_C = (V_C, E_C), \mathbf{H} = (H_{ij}, i, j \in V_C)$,
$\mathbf{M} = (M_i, i \in V_C), \boldsymbol{\gamma} = (\gamma_i, i \in V_C)$
Ensure: $\mathbf{G}_P \equiv \{G_P^1, G_P^2, \ldots\}$, $\bigcup_z E_P^z = E_C$, $|\mathbf{G}_P| \leq |E_C|$

1: INITIALIZERADIATIONPATTERNS$(G_C, \mathbf{M})$
2: $W \Leftarrow$ ORDERING(E_C)
3: $S \Leftarrow \emptyset$
4: $i \Leftarrow 1$
5: **while** $W \neq \emptyset$ **do**
6: $\quad G_P^i \Leftarrow (V_C, \emptyset)$
7: $\quad$ **for all** $e \in (W, S)$ **do**
8: $\quad\quad$ **if** POWERCONTROL$(G_P^i \cup e, \mathbf{H}, \gamma)$ **then**
9: $\quad\quad\quad G_P^i \Leftarrow G_P^i \cup e$
10: $\quad\quad\quad$ **if** $e \in W$ **then**
11: $\quad\quad\quad\quad W \Leftarrow$ ORDERING$(W \setminus e)$
12: $\quad\quad\quad\quad S \Leftarrow$ ORDERING$(S \cup e)$
13: $\quad\quad\quad$ **end if**
14: $\quad\quad$ **end if**
15: $\quad$ **end for**
16: $\quad i \Leftarrow i + 1$
17: **end while**

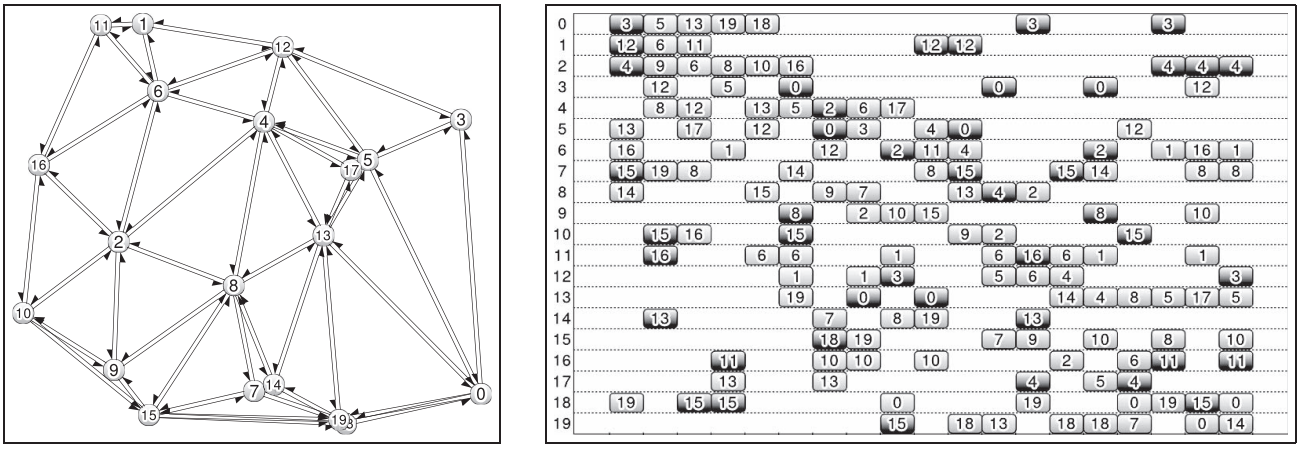
The radiation pattern $R_i(\theta)$ as used in (10) to compute SINR_{ij} is determined by choosing the antenna mode with the highest field strength in the direction of θ.

7.4 Connectivity-Based Transmission Packing

A commonly used heuristic approach for scheduling in OMAN is slotted random transmission packing, as detailed in Algorithm 1. This algorithm is an extension of the elementary capacity graph packing procedure in [8]. The approach is top down, in that it first requires the designer to provide a connectivity objective, specified by a directed graph $G_C = (V_C, E_C)$. The edges of this graph specify the channels for which the scheduling procedure should allocate resources.

The other inputs to the PACKTRANSMISSIONS algorithm are $\mathbf{H}$ which defines the channel state of every possible edge in the graph, and γ which is the vector of minimum SINR requirements on each node. The PACKTRANSMISSIONS procedure computes the set of concurrent transmissions, organized into time slots, until every channel represented as an edge in G_C is scheduled. The high-level steps in this packing method are

1. *Input.* Construct a connectivity graph G_C which specifies the desired channels of communication on the network. This graph serves as one of the inputs to the algorithm.
2. *Line 1.* Initialize the radiation patterns of each of the antenna modes based on the topology of graph G_C as described in Section 7.3.
3. *Line 2.* Generate an ordered queue W (for **W**aiting) of candidate transmissions that when combined together form the connectivity graph in step 1.
4. *While-Loop.* Pack concurrent transmissions selected in-order from the candidate queue W to form a feasible physical graph. Feasibility is determined by the Boolean function POWERCONTROL, described below. A second queue S (for **S**atisfied) of previously scheduled transmissions is then also iterated for packing.
5. *End of For-Loop.* Move the transmissions packed in step 3 to the second queue and if the candidate queue is still not empty, repeat step 4.



(a) Connectivity graph G_C that is one of the inputs to Algorithm 1. It defines the link topology of the capacity graph that the set of multiplexed schedules returned by this algorithm will form.

(b) Visualization of the transmissions events comprising one round of scheduling. There are three antenna modes on each node. The numbers in the boxes designate the index of the receiving node, and the numbers on the left-hand side designate the transmitter associated with all the transmissions in its respective row.

Fig. 11. The visualization of the input and output of Algorithm 1. The 19 slots of the schedule shown on the right, when multiplexed together, form the capacitated connectivity graph on the left.

The POWERCONTROL$(G_P, \mathbf{H}, \gamma)$ function returns "true" if a predefined PHY-layer optimization problem has a feasible solution. The input to this function is the graph G_P which defines the set of desired concurrent transmissions, the channel states $\mathbf{H}$, and minimum SINR requirements γ. This optimization problem needs to reflect the interference model of the network, and the relationship between transmit power, quality of desired channel, and the magnitude of interference seen at each receiver. A simple example of this is requiring that the distance between every active receiver and its closest interfering transmitter is greater than some radius. When channel information is known only through a set of samples, an extension of the RPC framework in (8) could be used with a constraint on the probability of feasibility on each channel. For this case study, we use the nominal power control constraint on the SINR as given by (5). That is, an edge $e \in W \cup S$ is added to G_P^i inside the for-loop at line 9, if an allocation of transmit powers could be found that can support an SINR_{ij} above the threshold γ_j on all the packed edges (i, j) in the physical graph.

When finished, algorithm PACKTRANSMISSIONS returns a set of physical graphs. They are then multiplexed as described in Section 3 so that the resulting schedule forms a capacitated graph where each edge is the sum of capacities on its respective transmission channel in the time slots where that channel is active. For this scenario, we use Shannon capacity as shown in (2).

For each schedule slot, this procedure seeks to pack in as many concurrent transmissions as possible without violating the minimum SINR requirement on each packed edge. The packing of every transmission increases the interference on the network and thus decreases the SINR for every edge already in the physical graph. The order in which the transmissions are packed, therefore, affects both the capacity of each graph and the total number of slots required to guarantee every edge in E_C a corresponding time to transmit in the schedule. The ORDERING function is a heuristic that performs this task. In this case study, we order the candidate transmissions randomly. Examples of other heuristics which provide consistently better solutions include ordering based on edge length (both ascending and descending).

This scheduling framework, as well as most combinatorial scheduling problems, is NP-Hard [19]. The heuristic approach we propose has a running time of $O(|E_C|^2)$. Each step inside the while loop iterates through all the $|E_C|$ edges in the two queues. In the worst case, the length of the schedule is $|E_C|$. Thus, the complexity of this packing algorithm is $O(|E_C|^2)$. The bottle neck operation, performed for each candidate physical graph, is the linear program POWERCONTROL. It is shown to be a linear program by Wu et al. [8], and thus has a strongly polynomial time solution [20]. In OMAN, the COIN-OR LP code [21], which implements the Simplex method, is used to solve this linear program.

Fig. 11 shows an example of the transmission events that form the full schedule required to produce the link topology of the input graph G_C in the capacitated graph that is formed by multiplexing the schedule output by the PACKTRANSMISSIONS algorithm. For this scenario, we use a Delaunay triangulation [22] as the link topology of graph G_C. A number of connectivity graph generation methods are implemented in OMAN including minimum spanning tree [23], k-closest neighbors [24], approximation algorithms for k-node and k-edge connected graphs [25], along with several distributed heuristic methods for k-connectivity [26].

7.5 Simulation and Results

Directional antennas provide a distinct advantage over antennas with omnidirectional transmission and reception radiation patterns in that their negative interference effect on the whole network is negligible except for nodes in the vicinity of the intended receiver. In this section, we use the bilevel abstraction to parametrize two characteristics of a radiation pattern which significantly contribute to the limited network-wide interference it provides. The first parameter is w, the width of the main lobe, and the second parameter is R_L, the gain of the side lobe. As we increase w from 0 to 360 degrees and R_L from 0 to R_H, the radiation

pattern changes from one that causes very little interference on the rest of the network to one that causes as much interference as an omnidirectional antenna.

The measure of delay in this scenario is the number of schedule slots $|\mathbf{G}_P|$ required to allocate a feasible set of transmissions which form a Delaunay triangulation graph. The sum-rate capacity is computed by summing over the Shannon capacity (see (3)) on each edge of the multiplexed capacitated graph G_F

$$C(G_F) = \frac{1}{|\mathbf{G}_P|} \sum_{z=1}^{|\mathbf{G}_P|} \sum_{(i,j) \in E_P^z} C_{ij}(G_P^z). \quad (11)$$

A network of 20 nodes is randomly distributed in a square arena of $1\,\mathrm{km}^2$. The range of possible transmit powers on each node is from 0 to 0.5 watts. The minimum SINR requirement on each node is 5. Each step in the simulation process performs the random-packing scheduling procedure in Algorithm 1 with Delaunay triangulation as the G_C input.

The default values for the two variables are $w = 42°$ and $R_L = 0.316$ with $R_H = 1$. These are the characteristics of the conservative bilevel abstraction of the measured radiation pattern in Fig. 10b. The simulation process is repeated 500 times for each value of the ω or R_L parameters, while the other one is kept at the default value. The average sum-rate capacity of those 500 runs is divided by the delay to produce a single data point in Fig. 12.

The ability to pack concurrent transmissions associated with a directional radiation pattern into a single time slot reduces the delay of the scheduling round. An important distinction to note is that in this case study capacity is only a metric and not an explicit objective. The packing of each addition transmissions in a schedule slot increases the interference on the network and thus decreases sum-rate capacity. Also, the power we use to compute the capacity metric is the minimum power required to satisfy the SINR on all the active channels in $\mathbf{G}_P$. Capacity could instead be made an objective in the POWERCONTROL function.

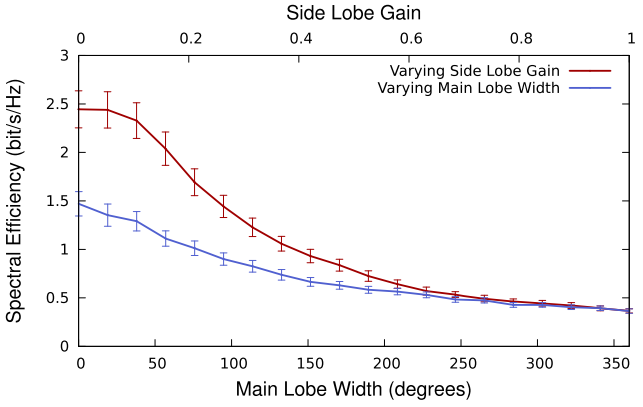


Fig. 12. Decrease in sum-rate capacity with an increase in side-lobe gain and main beam width. The results are averaged over 500 networks of 20 nodes. The default values for side-lobe gain and main lobe width used when the other value is being varied are computed from the measured radiation pattern shown in Fig. 10b. Error bars indicate the standard deviation estimate.

8 Case 3: Motion Planning for Connectivity and Throughput

8.1 Overview

Traditionally, the problem of planning movement is solved based on the physical characteristics of the field of operation or on the strategic waypoint requirements of the mission goal set. After a set of movement plans are chosen by the mobile agents, the network design process begins by allocating communication resources based on the position constraints defined by the chosen movement plans. We consider bridging the gap between the path planning problem and optimal network resource allocation, by allowing a fraction of the nodes to form a feedback loop in which a node is able to adjust its movement plan based on some measure of communication quality on the network state. These nodes form the support backbone of the network. They are unconstrained by any strategic waypoint requirements, but instead are exclusively tasked with moving according to kinematic constraints in such a way as to maximize network performance. The presence of such nodes may be essential when connectivity, survivability, or high-bandwidth throughput on the network is of great importance, or when conditions of the communication channel are poor and could be improved by an adjustment in the topology of the network. The emergence of unmanned aerial and ground vehicles as a real possibility in field [27] allows for this form of automated communication support planning to be a feasible solution for scenarios that otherwise would suffer from costly communication blackouts. The authors have proposed methods for various forms of this problem in [28], [29], [30] which include fixed-path planning, cooperative planning, and an ant colony optimization approach under incomplete knowledge of the environment, respectively.

Related problems such as formation control [31] and facility location [32] have received considerable attention. Both of these problems seek approaches which do not consider either the dynamic nature of a MANET topology or do not allow for an arbitrary communication objective over which to optimize movement. The novel aspect of this work that we seek to emphasize with this case study is that we define the node's position and velocity as just a small subset of decision variables involved in the cross-layer network design process. This allows us to jointly optimize

1. end-to-end throughput via routing,
2. capacity and delay via scheduling,
3. network lifetime via power control, and
4. topology via path planning.

In other words, the mobility control framework we present is significantly more complex, and realistic, in its model of communication than similar recent work in network-aware movement planning [33].

8.2 Method

The key parameter for this design framework is the number of nodes with path planning capability. The nodes which are assigned this feature, referred to in this section as "intelligent nodes," are assumed to also have the ability to acquire information about movement plans of other nodes in the network. This information can have high degrees of

uncertainty, but it nevertheless must be present for the planning algorithms to attain improvement in global network performance.

The high-level planning process involves three steps:

1. *Generate.* Compute movement options under kinematic constraints.
2. *Evaluate.* Compute (potentially multiple) global measurements of performance of each option.
3. *Execute.* Aggregate utility valuations to determine and execute the locally optimal option.

Movement options are generated randomly. In this case study, we look at the approach when the options are at a normally distributed distance from the current position of the intelligent node. The position must be in an area that is 1) traversable by the node, 2) not blocked off by a wall of obstacles, and 3) reachable by the node given its kinematic constraints (e.g., how fast it can move and how quickly it can turn). Once a single movement option is generated, a path to that position is computed assuming a point geometry in a space filled with simple polygons. This is computed using the algorithm given by Kapoor et al. [34].

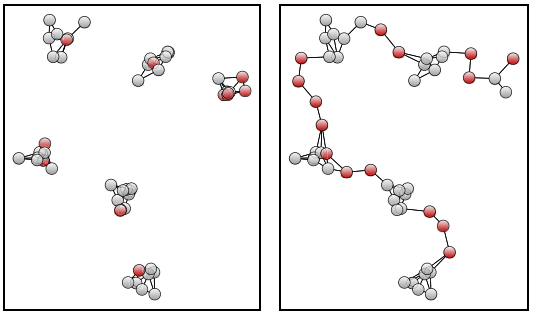
A "movement option" is described as a tuple of position (in space) and time (when the position is to be reached). The feasibility of this option is ensured by considering the maximum velocity of the intelligent node and the physical characteristics of the planning space. A "network state" (or simply "state") of this option is the position of all the nodes in the network at the time of the option.

Step 2 of the planning process is the evaluation of each movement option based on a set of metrics. The metrics are computed on an instance of a **FlowGraph** class introduced in Section 3.3. The construction of a capacitated graph is performed by the packing procedure in Algorithm 1. The connectivity graph G_C provided as input for this is one where $(i,j) \in G_C$ if $d_{ij} < R = \frac{1}{\pi}\sqrt{A \log n / n}$, where A is the area of the arena and n is the number of nodes. This distance threshold heuristic is based on the asymptotic connectivity radius from [35]. The throughput on this graph is computed by solving multicommodity max flow problem (4) over graph G_F.

8.3 Simulation and Results

The movement planning scheme in OMAN seeks to optimize an arbitrary objective function which evaluates the desirability of a network topology where the control variables are the positions of the nodes with path planning capability. The only constraint on the nodes is that they move through free space and within their kinematic capabilities. Therefore, any evaluation function can be provided to the planning system, and it would optimize movement of the intelligent nodes based on that function.

We demonstrate the improvement in network performance that can be achieved in specific cases by presenting two simulation examples. In the first example, only nodes with path planning capability are moving. This limited case is considered for purposes of visualization, to demonstrate the evolution of the network topology caused by the planning process. In the first example, an initial clustered topology is used.



(a) 6 connected components (at 0 seconds) (b) 1 connected component (at 4 seconds)

Fig. 13. Clustered 50-node topology example of path planning to minimize the number of connected components. Thirty percent of the nodes (marked red) have planning capability. The other nodes are not moving. The images show the evolution of the network topology toward an optimal solution.

The second simulation example seeks to demonstrate the network performance achievable in the general mobility case where all nonplanning nodes are moving according to the random-waypoint movement model [36]. Four different objective functions are used to determine the improvement attainable through planning as the number of intelligent nodes increases.

The maximum velocity of nodes is chosen for these experiments such that R meters can be traveled by a mobile node in 1 second. Therefore, 1 second in those two plots has the interpretation as the time it takes to travel sufficiently far to almost certainly have some effect on the connectivity of the network.

8.3.1 First Scenario: Connected Components

The first simulation scenario is of a 50 node clustered topology shown in Fig. 13a. Fifteen nodes are randomly assigned with the path planning capability. In Fig. 13, these nodes are indicated with red. The SNR requirement for feasible channels is chosen conservatively such that nodes need to be relatively near each other to be within range of effective communication. In the figures, the lines between nodes indicate feasible communication channels. The number of connected components is then measured on the flow graph F_G formed based on these edges. Fig. 13b shows that in 4 seconds, 15 planning nodes are able to fully reconnect the network.

8.3.2 Second Scenario: Cross-Layer Objectives

The second simulation scenario is of 100 nodes moving under the random waypoint model at maximum velocity. The parameter of interest is the fraction of nodes which are assigned path planning capability. For each value of this parameter, we run the simulation for 10,000 seconds in simulation time, averaging the evaluation of the network state at 0.1 second intervals. Since the nonplanning nodes are constantly moving, there is no convergence in network performance as seen in the previous two sections when those nodes were not moving. Each simulation step of the 100,000 per run for 100 nodes, 0.4 fraction of nodes with planning capability, with the objective to maximize throughput took from 0.05 to 0.2 seconds to compute on a 2.4 GHz processor. This represents the parameters for the most computationally intensive simulation step in this scenario.

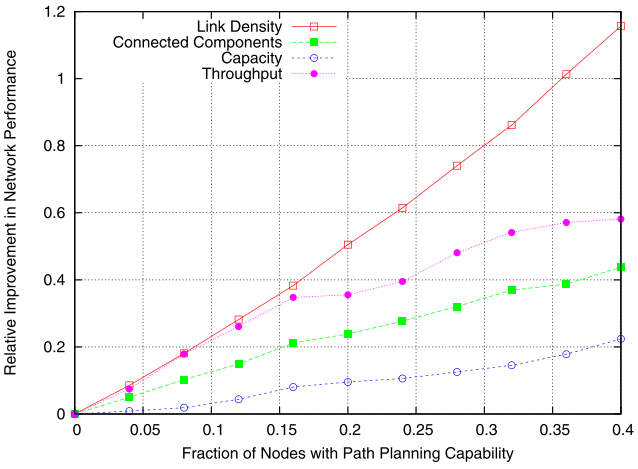


Fig. 14. The increase of improvement attained from movement planning as related to the density of intelligent nodes.

We consider four metrics in this simulation scenario:

- *Link density.* The number of feasible edges in the flow graph G_F.
- *Connected components.* The number of strongly connected components in the flow graph G_F.
- *Capacity.* The sum-rate capacity on the flow graph as computed by (11).
- *Throughput.* The maximum achievable throughput on G_F as computed by (4).

The first two are from the category of connectivity metrics, while the latter two are from the category of data capacity metrics. The throughput is computed by choosing one source node i and sink node j on the flow graph such that i and j are the two nodes farthest away from each other while residing in the same connected component.

Fig. 14 shows the relative improvement in network performance as the fraction of intelligent nodes in the scenario increases. Improvement is measured as the difference between the current performance and the base performance divided by the base performance. The base performance is the average utility of the network state when none of the nodes have path planning capability. Therefore, when $x = 0$ in the figure, improvement also equals zero.

As expected, improvement in network performance increases as the number of nodes with planning capability increases. One interesting observation from these results is that greater improvement is generally achievable in the throughput than in the capacity of the network, because movement planning is generally effective at load balancing and removing flow bottlenecks in the network.

9 Conclusion

We detail three representative case studies of network design which showcase the tradeoffs and optimization approaches implemented in the OMAN library. The first case covers a method for finding a robust power allocation which solves the biobjective power control problem. The second case builds on the PHY layer framework of the first case, and solves the scheduling problem with the addition of unidirectional radiation patterns. The third case builds on the first two and optimizes the throughput of the network by treating the movement of some of the nodes as decision variables. These cases serve as examples of using OMAN to study cross-layer optimization problems in network design.

Acknowledgments

This work is funded by the US Army Communications-Electronics Research, Development and Engineering Center (CERDEC), under contract #DAAB-07-01-9-L504.

References

[1] L. Fridman, R. Primerano, S. Weber, and M. Kam, "Cooperative Surveillance in Video Sensor Networks," *Proc. Second ACM/IEEE Int'l Conf. Distributed Smart Cameras (ICSDC)*, Sept. 2008.
[2] R. Fourer, D. Gay, and B. Kernighan, *AMPL: A Mathematical Programming Language.* AT&T Bell Laboratories, 1989.
[3] Scalable Network Technologies, http://www.scalable-networks.com, 2012.
[4] Opnet Modeler, http://www.opnet.com, 2012.
[5] Motorola Enterprise Planner, http://www.motorolasolutions.com, 2012.
[6] K. Czarnecki, "Overview of Generative Software Development," *Proc. Int'l Workshop Unconventional Programming Paradigms*, J.-P. Banâtre, P. Fradet, J.-L. Giavitto, and O. Michel, eds., pp. 97-97, 2005.
[7] H. Mouftah and R. Sturgeon, "Distributed Discrete Event Simulation for Communication Networks," *IEEE J. Selected Areas in Comm.*, vol. 8, no. 9, pp. 1723-1734, Dec. 1990.
[8] Y. Wu, P. Jain, and K. Kung, "Network Planning in Wireless Ad Hoc Networks: A Cross-Layer Approach," *IEEE J. Selected Areas in Comm.*, vol. 23, no. 1, pp. 136-150, Jan. 2005.
[9] S. Even, A. Itai, and A. Shamir, "On the Complexity of Time Table and Multi-Commodity Flow Problems," *Proc. 16th Ann. Symp. Foundations of Computer Science*, pp. 184-193, Oct. 1975.
[10] P. Murphy, A. Sabharwal, and B. Aazhang, "Design of Warp: A Flexible Wireless Open-Access Research Platform," *Proc. European Signal Processing Conf. (EUSIPCO)*, 2006.
[11] E. Bjornson, D. Hammarwall, and B. Ottersten, "Beamforming Utilizing Channel Norm Feedback in Multiuser MIMO Systems," *Proc. IEEE Workshop Signal Processing Advances in Wireless Comm. (SPAWC)*, 2007.
[12] J. Lin, "Multiple-Objective Problems: Pareto-Optimal Solutions by Method of Proper Equality Constraints," *IEEE Trans. Automatic Control*, vol. AC-21, no. 5, pp. 641-650, Oct. 1976.
[13] L. Fridman, R. Grote, S. Weber, K.R. Dandekar, and M. Kam, "Robust Optimal Power Control for Ad Hoc Networks," *Proc. 40th Ann. Conf. Information Sciences and Systems (CISS)*, Mar. 2006.
[14] J. Mulvey, R. Vanderbei, and S. Zenios, "Robust Optimization of Large-Scale Systems," *Operations Research*, vol. 43, pp. 264-281, Mar. 1995.
[15] T. Camp, J. Boleng, and V. Davies, "A Survey of Mobility Models for Ad Hoc Network Research," *Wireless Comm. and Mobile Computing*, vol. 2, no. 5, pp. 483-502, 2002.
[16] S. Agarwal, R. Katz, S. Krishnamurthy, and S. Dao, "Distributed Power Control in Ad-Hoc Wireless Networks," *Proc. 12th IEEE Int'l Symp. Personal, Indoor and Mobile Radio Comm.*, vol. 2, 2001.
[17] T. ElBatt and A. Ephremides, "Joint Scheduling and Power Control for Wireless Ad Hoc Networks," *IEEE Trans. Wireless Comm.*, vol. 3, no. 1, pp. 74-85, Jan. 2004.
[18] D. Piazza, M. D'Amico, and K. Dandekar, "Performance Improvement of a Wideband Mimo System by Using Two-Port RLWA," *IEEE Antennas and Wireless Propagation Letters*, vol. 8, pp. 830-834, 2009.
[19] P. Brucker, *Scheduling Algorithms*, fifth ed. Springer, Mar. 2007.
[20] N. Megiddo, "On the Complexity of Linear Programming," *Advances in Economic Theory: Fifth World Congress*, T. Bewley, ed., pp. 225-268, Cambridge Univ., 1987.
[21] R. Lougee-Heimer, "The Common Optimization INterface for Operations Research: Promoting Open-Source Software in the Operations Research Community," *IBM J. Research and Development*, vol. 47, no. 1, pp. 57-66, 2003.

[22] X. Li, G. Calinescu, P. Wan, and Y. Wang, "Localized Delaunay Triangulation with Application in Ad Hoc Wireless Networks," *IEEE Trans. Parallel and Distributed Systems,* vol. 14, no. 10, pp. 1035-1047, Oct. 2003.
[23] R. Graham and P. Hell, "On the History of the Minimum Spanning Tree Problem," *IEEE Annals of the History of Computing,* vol. AHC-7, no. 1, pp. 43-57, Jan.-Mar. 1985.
[24] X. Jia, D. Kim, S. Makki, P. Wan, and C. Yi, "Power Assignment for k-Connectivity in Wireless Ad Hoc Networks," *J. Combinatorial Optimization,* vol. 9, no. 2, pp. 213-222, 2005.
[25] D. West et al., *Introduction to Graph Theory.* Prentice Hall, 2001.
[26] F. Dai and J. Wu, "On Constructing k-Connected k-Dominating Set in Wireless Networks," *Proc. 19th IEEE Int'l Parallel and Distributed Processing Symp.,* 2005.
[27] A. Richards, J. Bellingham, M. Tillerson, and J. How, "Coordination and Control of Multiple UAVs," *Proc. AIAA Guidance, Navigation, and Control Conf.,* 2002.
[28] L. Fridman, P.J. Modi, S. Weber, and M. Kam, "Communication Based Motion Planning," *Proc. 41st Conf. Information Sciences and Systems (CISS),* Mar. 2007.
[29] L. Fridman, S. Weber, C. Graff, and M. Kam, "Path Planning for Network Performance," *Proc. IEEE GlobeCom,* Nov. 2007.
[30] L. Fridman, S. Weber, V. Kumar, and M. Kam, "Distributed Path Planning for Connectivity under Uncertainty by ant Colony Optimization," *Proc. 27th Am. Control Conf. (ACC),* June 2008.
[31] T. Balch and R. Arkin, "Behavior-Based Formation Control for Multirobot Teams," *IEEE Trans. Robotics and Automation,* vol. 14, no. 6, pp. 926-939, Dec. 1998.
[32] Z. Drezner, *Facility Location: A Survey of Applications and Methods.* Springer Verlag, 1995.
[33] C. Dixon and E.W. Frew, "Maintaining Optimal Communication Chains in Robotic Sensor Networks Using Mobility Control," *Mobile Networks and Applications,* vol. 14, pp. 281-291, June 2009.
[34] S. Kapoor, S.N. Maheshwari, and J.S.B. Mitchell, "An Efficient Algorithm for Euclidean Shortest Paths among Polygonal Obstacles in the Plane," *GEOMETRY: Discrete and Computational Geometry,* vol. 18, pp. 377-383, 1997.
[35] P. Gupta and P. Kumar, "Critical Power for Asymptotic Connectivity in Wireless Networks," *Stochastic Analysis, Control, Optimization and Applications: A Volume in Honor of WH Fleming,* pp. 547-566, Springer, 1998.
[36] T. Camp, J. Boleng, and V. Davies, "A Survey of Mobility Models for Ad Hoc Network Research," *Wireless Comm. and Mobile Computing,* vol. 2, no. 5, pp. 483-502, 2002.

Lex Fridman received the BS and MS degrees from Drexel University in 2010. He is currently working toward the PhD degree in computer engineering from Drexel University. His research interests include numerical optimization of communication networks and applied artificial intelligence. He is a student member of the IEEE.

Steven Weber received the BS degree in 1996 from Marquette University in Milwaukee, Wisconsin, and the MS and PhD degrees from the University of Texas at Austin in 1999 and 2003, respectively. He joined the Department of Electrical and Computer Engineering at Drexel University in 2003, where he is currently an associate professor. His research interests are centered around mathematical modeling of computer and communication networks, specifically streaming multimedia and ad hoc networks. He is a senior member of the IEEE.

Charles Graff is a senior project engineer at CERDEC, US Army, RDECOM Fort Monmouth, New Jersey. As an expert on wireless and mobile ad hoc networking, he serves as an advisor to several key projects at CERDEC. He is a member of the IEEE.

David E. Breen received the BA degree in physics from Colgate University in 1982. He received the MS and PhD degrees in computer and systems engineering from Rensselaer Polytechnic Institute in 1985 and 1993. He is currently an associate professor in the Computer Science Department of Drexel University. Before coming to Drexel, he held research positions at the California Institute of Technology, the European Computer-Industry Research Centre, the Fraunhofer Institute for Computer Graphics, and the Rensselaer Design Research Center. His research interests include geometric modeling, medical image analysis, self-organization algorithms, and biological simulation. He has authored or coauthored more than 85 technical papers, articles, and book chapters on these subjects. He is a member of the IEEE.

Kapil R. Dandekar received the BS degree in electrical engineering from the University of Virginia in 1997. He received the MS and PhD degrees in electrical and computer engineering from the University of Texas at Austin in 1998 and 2001, respectively. In 1992, he worked at the US Naval Observatory and from 1993 to 1997, he worked at the US Naval Research Laboratory. In 2001, he joined the Electrical and Computer Engineering Department at Drexel University in Philadelphia, Pennsylvania. He is currently an associate professor and the director of the Drexel Wireless Systems Laboratory (DWSL). DWSL has been supported by the US National Science Foundation (NSF), Army CERDEC, National Security Agency, Office of Naval Research, and private industry. His current research interests involve MIMO ad hoc networks, reconfigurable antennas, free space optical communications, ultrasonic communications, and sensor networks. He is a senior member of the IEEE.

Moshe Kam received the BS degree in electrical engineering from Tel Aviv University in 1976 and the MSc and PhD degrees from Drexel University in 1985 and 1987, respectively. He is the Robert Quinn professor and Department head of electrical and computer engineering at Drexel University, and the director of Drexel University's DHS/NSA Center of Excellence in Information Assurance Education. His professional interests are in system theory, detection and estimation, information assurance, robotics, navigation, and engineering education. He is a member of the board of directors of IEEE and ABET, Inc. He served as the IEEE Director of Region 2 (Eastern United States from 2003-2004), IEEE Vice President of Educational Activities (2005-2007), and 2011 President and CEO of IEEE. He is a fellow of the IEEE and the IEEE Computer Society.