Noname manuscript No.
(will be inserted by the editor)

Maximum Weight Independent Sets in an Infinite Plane with Uni- and Bidirectional Interference Models

Jarno Nousiainen · Jorma Virtamo · Pasi Lassila

the date of receipt and acceptance should be inserted later

Abstract We study the maximum weight independent sets of links between nodes distributed randomly in an infinite plane. Different definitions of the weight of a link are considered, leading to slight variations of what is essentially a spatial reuse problem in wireless multihop networks. A simple interference model is assumed with the interference radius equaling the transmission radius. In addition to unidirectional interference from a transmitter to the receivers of other links, also an RTS/CTS-type bidirectional handshake is considered. We study both the case where the transmission radius is fixed and tunable through power control. With a fixed transmission radius, we derive asymptotic results for the low and high density regimes. The main contribution is in the numerical results for the maximum weight, establishing some previously unknown parameters of stochastic geometry. The results are obtained by the Moving Window Algorithm that is able to find the maximum weight independent set in a strip of limited height but unlimited length. By studying the results as a function of the height of the strip, we are able to extrapolate to the infinite plane.

Keywords Maximum weight independent set · Spatial reuse · Random geometric graph · Asymptotics · Wireless multihop network · Capacity

J. Nousiainen · J. Virtamo · P. Lassila
Aalto University, Department of Communications and Networking, P.O.Box 13000, FI-00076 Aalto, Finland

J. Nousiainen
Tel.: +358-9-470 24786, Fax: +358-9-470 22474
E-mail: jarno.nousiainen@tkk.fi

J. Virtamo
E-mail: jorma.virtamo@tkk.fi

P. Lassila
E-mail: pasi.lassila@tkk.fi

1 Introduction

Consider the following fundamental problem: given a network in an infinite plane with nodes placed according to a planar Poisson process and connected to each other via wireless links with given weights and a given transmission range, what is the total weight (per unit area or per node) of the maximum weight set of non-interfering links under two Boolean-type interference models, the interference range being assumed to equal the transmission range? In the terminology of graph theory, the set of non-interfering links maps to what is called an independent set; we use these terms interchangeably. The problem is purely one of stochastic geometry, but it has a close connection to modeling the capacity of large-scale wireless multihop networks, and this paper is written with that viewpoint in mind.

In this paper[1], we study the above problem with three different kinds of weights: A) unweighted, i.e., each link has the weight one, B) weighted by the length of the projection of the link in a given direction, e.g., on the x-axis, and C) weighted by the length of the link. In case A, only the number of links in an independent set, i.e., the size of the set is counted. The problem is then just the maximum independent set problem. Case B corresponds to the concept of forwarding capacity, as the x-progress tells us how much traffic the network can instantaneously forward in a given direction. Case C is similar but without sense of direction.

Each of these three cases defines a challenging problem in stochastic geometry. Finding the maximum independent set for a given finite graph is known to be an NP-complete problem, and in our case the graph is even infinite. However, in the present problem the

[1] A short version of the paper [16] has been published at ITC 21

graphs are not arbitrary but the interference is localized; links separated far enough do not interfere with each other. In our approach, we will efficiently exploit this special structure of the graph.

The results are obtained by applying the Moving Window Algorithm, originally developed in a different context in [19], but fine-tuned for the present application. The algorithm is able to find the maximum weight of an independent set for any network realization in a strip of limited height but unlimited length by moving a window along the strip. The window is high enough to cover the strip in the vertical direction and wide enough so that no two links on different sides of the window interfere with each other. The algorithm maintains the cumulative weight of the maximum weight independent set so far conditioned on the choice of the conflict-free set in the window. By studying the results as a function of the height of the strip, we are able to extrapolate to the infinite plane.

We study the problem with two different assumptions on the interference model. In the Boolean interference model, a transmission on a link is successful if there are no other active transmitters whose transmission range reaches the receiver. In the bidirectional Boolean interference model, a successful transmission requires that both the sender and the receiver are outside the range of competing transmissions. This models, e.g., the effect of RTS/CTS handshaking in 802.11 networks during which the transmitter and receiver both need to be able to hear each other. In the basic case the transmission range is assumed to be fixed. Also the impact of power control is studied where the transmission range can be adjusted up to a given maximum radius. The idea is that not every sending node uses the maximum radius but a radius just large enough to reach the receiving node, thereby minimizing the interference.

Our analytical contribution comprises of asymptotic analyses of the different cases A, B, and C with a fixed transmission radius. The asymptotics are analyzed both when the mean number of neighbors tends to zero and to infinity (loosely speaking, the low and high density asymptotics). In the previous case, the analysis is simple and the results are exact. In the high density limit, the problem is more intricate, and we present only a rudimentary analysis, which however, we believe, captures the essential dependency. The asymptotic behaviors at both ends yield insight on the behavior also in the intermediate range.

The main contribution of this paper is in the numerical results representing the total weight of the maximum independent sets of the studied systems as a function of the mean neighborhood size. In the case of a fixed transmission radius, these curves have a maximum, which is of special interest in defining the best that can be obtained, as well as the optimal value of the transmission radius in relation to the mean distance between the nodes. In the case of an adjustable transmission range (up to a given maximum), no maximum exists, since all the independent sets that are feasible with a given maximum range are feasible when the maximum range is made larger. Similarly, we see that the results with an adjustable range are always as good as or better than when only the maximum range can be used since anything that can be done with a fixed range can be done with an adjustable range.

The algorithm is computationally demanding, especially for large windows and high mean neighborhood sizes. The results are presented for as large a parameter range as it has been possible to proceed. For cases A and B the explored range covers the points where the curves with fixed transmission range reach their maxima; for case C this is only attainable with the bidirectional interference model.

The rest of this paper is organized as follows. Section 2 presents the related work. In Section 3, we introduce the notation and present scaling considerations to reduce the unknowns to the minimum. The asymptotic results are derived in Section 4. In Section 5, we describe the Moving Window Algorithm. The numerical results are presented in Section 6, and we conclude in Section 7.

2 Related work

The graph theoretic correspondence of our problem to the well-known maximum weight independent set problem results from the following mapping between the graph model of the wireless network and the so-called interference graph. An independent set of links in a wireless network is an independent set (of vertices) in the network's interference graph, where each link corresponds to a vertex, and two vertices are adjacent if the links interfere with each other. Determining the maximum weight of an independent set of links is equivalent with finding the maximum weight independent set in the interference graph. The maximum weight independent set problem is one of the most studied problems in combinatorial optimization and is known to be NP-hard for a general graph [7]. The exact solutions of NP-hard problems can typically be found only for moderate input sizes. For recent results on exact algorithms, see [5] and its references.

In wireless networking, the question of maximum weight independent sets of links arises, e.g., in the context of analyzing the local forwarding capacity of massively dense networks [2,9,10,12] motivated by the future applications of large scale sensor networks. From a single node's perspective the surrounding network appears as an infinite network of randomly placed wireless nodes. The solution to the maximum weight problem (B) gives the maximum *instantaneous* forwarding capacity in the neighborhood of the considered node, which in turn sets an upper bound for the local sustainable mean forwarding capacity, i.e., the average rate at which traffic can be "moved" in a given direction [14,15]. (Note that the maximum weight independent set cannot be used repeatedly for forwarding traffic because it consists of independent, isolated links that do not form a connected network.) The concept is similar to density of progress, see [1,17].

Results like these yield useful information about the achievable gains from utilizing optimal global coordination in multihop communications, and thus complement the well-known scaling results for the capacity of multihop networks, see [6,8]. The maximum weight independent set problem is also related to the challenging global optimization phase of maximum weight scheduling [18]. To relax the need for any global optimization, there has been an effort to solve this using distributed scheduling [11,13].

A spatial reuse problem similar to the one of this paper has been further studied from the MAC point of view in [3]. The approaches provided by stochastic optimization in its various incarnations, like simulated annealing or the "packing approach" of [4], can also be used to find the maximum weight independent set. We believe, however, that the ability of the Moving Window Algorithm to give exact results in a stripe gives it an advantage that is hard to beat.

The present paper extends the work of [16] to cover the much used and in practice relevant bidirectional interference of IEEE 802.11. For this case, we provide the asymptotic analysis and the numerical results obtained with the Moving Window Algorithm. Previously, we have applied the algorithm for case B (weighted with x-progress) in [15] with unidirectional interference.

3 Preliminaries

As discussed in the Introduction, we consider a system where the locations of nodes are assumed to obey a planar Poisson process. The intensity of this process is denoted by λ. In the basic case, the transmission radius is fixed to the maximum that is denoted by R. In the case of power control, the transmission range can be chosen freely up to this maximum. Thus, the topology of the network can be modeled as a directed graph $G = (V, E)$ where there exists an edge $e = (u, v) \in E$, between nodes $u, v \in V$ if $d(u, v) \leq R$, where $d(u, v)$ is the Euclidean distance between the nodes. If $(u, v) \in E$, node u is called a neighbor of node v and vice versa.

3.1 Interference

Interference is modeled using two Boolean-type interference models with the interference radius equaling the (used) transmission radius. In the first one, the basic Boolean interference model, a transmission of a unidirectional link interferes with all the receptions inside its range, implying that a link is only possible if the receiver hears exactly one transmission (excluding simultaneous transmission and reception). Using the above notations, we see that links a and e interfere with each other if

$$d(t(a), r(e)) \leq R(a) \vee d(r(a), t(e)) \leq R(e), \tag{1}$$

where $t(e)$ is the transmitting node of link $e \in E$, $r(e)$ the receiving node, and $R(e) \leq R$ the actual transmission radius of $t(e)$ that in the fixed case equals the maximum, R. The Boolean interference model is a rough approximation of a SINR model with a critical threshold for reception. It unrealistically allows multiple transmissions just outside the range from a receiver, though these situations are not particularly common in random node deployments.

In the bidirectional Boolean interference model, both the origin and destination node transmit during a session, and the communication is successful only if neither of them hears a competing transmission. Thus, the links a and e interfere with each other if

$$\begin{aligned} &d(t(a), r(e)) \leq R^* \vee d(r(a), t(e)) \leq R^* \vee \\ &d(t(a), t(e)) \leq R^* \vee d(r(a), r(e)) \leq R^*, \end{aligned} \tag{2}$$

where $R^* = \max\{R(a), R(e)\}$. Bidirectional communication occurs, e.g., when the transmitter and the receiver exchange control messages before the actual data transmission. This is characteristic to the RTS/CTS handshake of IEEE 802.11.

3.2 Scaling considerations

The beauty of the system with the used interference models is that there are only two parameters in the model, λ and R, and there is only one (independent) dimensionless parameter that can be formed from these two system parameters. We use the most natural one, viz. the mean number of neighbors within the transmission range, denoted by ν,

$$\nu(\lambda, R) = \pi \lambda R^2. \tag{3}$$

In the case of an adjustable transmission range, this corresponds to the mean number of neighbors with the maximum transmission radius. With proper scaling considerations, as detailed below, all unknown functions of the two system parameters can be reduced to functions of this single variable.

Our goal is to find the maximum weight (or size) of an independent set of links per unit area in the following cases:

A. Unweighted,
B. Weighted by the x-progress of the links,
C. Weighted by the length of the links.

In the first case (A), we aim to calculate the maximum number of links per unit area, denoted by $U(\lambda, R)$. By dimensional analysis, we can write

$$U(\lambda, R) = \lambda\, u(\nu(\lambda, R)), \tag{4}$$

where $u(\nu)$ is a dimensionless function of a single variable to be determined. In fact, $u(\nu(\lambda, R))$ represents the number of links per node.

The second case (B) differs from the previous as we are interested in the maximum density of progress, $U_x(\lambda, R)$, that is the maximum x-progress (the progress of the link in a fixed direction, see [1,17]) per unit area, and by dimensional analysis we get

$$U_x(\lambda, R) = \sqrt{\lambda}\, u_x(\nu(\lambda, R)), \tag{5}$$

where $u_x(\nu)$ is another unknown dimensionless function.

In the last case (C) we are interested in the total length of the links per unit area, U_l, that can be expressed with the help of a yet another dimensionless function u_l, exactly as in (B),

$$U_l(\lambda, R) = \sqrt{\lambda}\, u_l(\nu(\lambda, R)). \tag{6}$$

In short, our task is to find the dimensionless functions $u(\nu)$, $u_x(\nu)$, and $u_l(\nu)$. In the sequel, speaking generally we use $u_*(\nu)$ to represent any of these functions. The notation $U_*(\lambda, R)$ is used similarly.

4 Low and high density asymptotics

In this section we consider the asymptotic behavior of the dimensionless functions $u_*(\nu)$ when the mean node degree approaches zero or infinity. The analysis when ν approaches infinity is rudimentary but believed to capture the essential dependency. The obtained theoretical limits provide useful insight and will be compared to simulations in Section 6.3.

4.1 Asymptotics in the limit $\nu \to 0$

When the mean degree of a node approaches zero, the same consideration is valid for both interference models. Basically, it is more about being able to form links than it is about the links interfering with each other.

In the unweighted case (A), a general upper bound for the function U, the maximum number of links per unit are, is given by

$$U(\lambda, R) \le \frac{1}{2}\, \lambda(1 - e^{-\nu}). \tag{7}$$

The reasoning with the above inequality is that there are on the average λ nodes per unit area, and that one obviously gets an upper bound for $U(\lambda, R)$ if each node can freely choose the neighbor to form a link with, without any restrictions imposed by other links. The factor $\frac{1}{2}$ accounts for the fact that it takes two nodes to form a link. The parenthetical expression is the probability that a node has a neighbor. Written in terms of $u(\nu)$ the upper bound takes the form

$$u(\nu) \le \frac{1}{2}(1 - e^{-\nu}). \tag{8}$$

It is also obvious that asymptotically when $\nu \to 0$ the upper bound becomes tight, since in the rare cases when a node has a neighbor within its transmission radius, they can indeed form a link with a high probability without any other link interfering. In this asymptotic regime the probability $(1 - \exp\{-\nu\}) \approx \nu$, and we have

$$u(\nu) \sim \frac{1}{2}\, \nu, \text{ when } \nu \to 0. \tag{9}$$

For the case weighted by the x-progress (B), the same reasoning otherwise applies but instead of the probability of having a neighbor, we have the mean x-progress of the link to the furthest neighbor. Hence, the general upper bound for the function U_x becomes

$$U_x(\lambda, R) \le \frac{1}{2}\, \lambda R\, X(\nu), \tag{10}$$

where $X(\nu)$ is the mean distance (in units of R) from a randomly chosen node to its most distant neighbor node in the x-direction, i.e., absolute value of the x-distance (if there is none, the distance is taken to be zero). For $u(\nu)$, we have

$$u_x(\nu) \le \frac{1}{2}\sqrt{\frac{\nu}{\pi}}\, X(\nu). \tag{11}$$

When ν is small, $X(\nu) \approx 4\nu/(3\pi)$, where $4/(3\pi)$ is the mean x-distance to a neighbor, and ν is the approximate probability of having a neighbor. Thus, we have

$$u_x(\nu) \sim \frac{2}{3}\left(\frac{\nu}{\pi}\right)^{3/2}, \text{ when } \nu \to 0. \tag{12}$$

The third case (C) is similar to the second case, but we have to replace the x-distance between the nodes by the actual distance, $L(\nu)$. Hence, $L(\nu) \approx 2\nu/3$, and

$$u_l(\nu) \sim \frac{1}{3\sqrt{\pi}}\,\nu^{3/2}, \text{ when } \nu \to 0. \tag{13}$$

4.2 Asymptotics in the limit $\nu \to \infty$

We now turn our attention to how $U(\lambda, R)$ behaves for large λ when R is considered to be fixed and present a plausible reasoning for the asymptotics. The analysis is different for the unidirectional and bidirectional interference models.

Unidirectional interference leads to the starting observation that if the end points of a link can be arbitrarily placed on a continuous plane, then the most efficient way of packing links is to form vertical columns. The claim is most obvious in the case with x-progress (B). There has to be a distance larger than R between two consecutive links, as illustrated in Figure 1, but the vertical distance between the links can be small. In fact, the Boolean interference model (unrealistically) sets no limit on how densely the links can be vertically packed: two parallel links of maximal length R, however close, never interfere with each other. This suggests that for a very high λ, when there are nodes almost everywhere, the maximum weight independent sets consist of links forming vertical columns.

In cases A and C, that are undirected, the packing can be done even more efficiently by changing the direction of every other column. This way a small distance ε is enough between the columns as the endpoints near each other are all either transmitters or receivers.

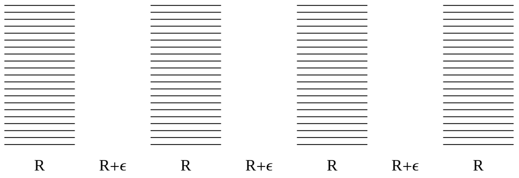


Fig. 1: On a continuous plane links can be efficiently stacked in vertical columns (case B). In cases A and C, links can be packed even tighter since ε margin is enough between columns transmitting in alternating directions.

The next step is to estimate the expected vertical distance between the links. Based on the above observation we consider a naive model where, starting from a vertical link of length R, the end points of the next link above are determined independently by proceeding in the vertical direction in the shown areas of Figure 2 until next node (from the Poisson process) is found.

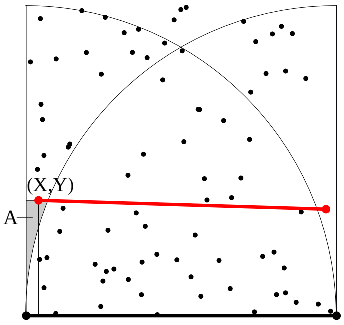


Fig. 2: The simplified model for estimating the vertical distance between the stacked links.

The width x of the area between the vertical line and the circle is for small heights y approximately parabolic, $x \approx y^2/(2R)$. Denote the coordinates (random variables) of the node by (X, Y). Since $A \sim \mathrm{Exp}\,(\lambda)$ and $A \approx Y^3/(6R)$, we have the complementary distribution function of Y,

$$\mathrm{P}\{Y > y\} = \mathrm{P}\{A > y^3/(6R)\} = e^{-\lambda y^3/(6R)},$$

i.e., $\mathrm{P}\{Y^n > y\} = \mathrm{P}\{Y > y^{1/n}\} = e^{-\lambda y^{3/n}/(6R)}$, from which

$$\begin{aligned}\mathrm{E}[Y^n] &= \int_0^\infty e^{-\lambda y^{3/n}/(6R)} dy \\ &= \left(\frac{6R}{\lambda}\right)^{n/3} \frac{n}{3} \int_0^\infty t^{n/3-1} e^{-t} dt \\ &= \left(\frac{6R}{\lambda}\right)^{n/3} \frac{n}{3}\, \Gamma\left(\frac{n}{3}\right) = \left(\frac{6R}{\lambda}\right)^{n/3} \Gamma\left(1+\frac{n}{3}\right),\end{aligned}$$

where, in the second step, a new variable $t = \lambda y^{3/n}/(6R)$ has been introduced. In particular we have

$$\mathrm{E}[Y] = \Gamma\left(\frac{4}{3}\right)\left(\frac{6R}{\lambda}\right)^{1/3}, \quad \mathrm{E}[Y^n] = \frac{\Gamma(1+\frac{n}{3})}{\Gamma(\frac{4}{3})^n}\,\mathrm{E}[Y]^n,$$

whence the variance is

$$\mathrm{V}[Y] = \left(\frac{\Gamma(\frac{5}{3})}{\Gamma(\frac{4}{3})^2} - 1\right) \mathrm{E}[Y]^2 \approx 0.132\mathrm{E}[Y]^2.$$

The distribution of X is determined by that of Y, $X \sim \mathrm{Uniform}(0,\, Y^2/(2R))$, from which

$$\mathrm{E}[X] = \mathrm{E}[\mathrm{E}\,[X\,|\,Y]] = \mathrm{E}[\frac{Y^2}{4R}] = \frac{\Gamma(\frac{5}{3})}{4R\Gamma(\frac{4}{3})^2}\,\mathrm{E}[Y]^2$$

and

$$\begin{aligned}\mathrm{V}[X] &= \mathrm{E}[\mathrm{E}\left[X^2\,|\,Y\right]] - \mathrm{E}[X]^2 = \mathrm{E}[\frac{1}{3}(\frac{Y^2}{2R})] - \mathrm{E}[\frac{Y^2}{4R}]^2 \\ &= \frac{\frac{1}{3}\Gamma(\frac{7}{3}) - \frac{1}{4}\Gamma(\frac{5}{3})^2}{4\Gamma(\frac{4}{3})^4 R^2}\,\mathrm{E}[Y]^4 \approx \frac{0.0759}{R^2}\,\mathrm{E}[Y]^4.\end{aligned}$$

Now, consider the random walk $\mathbf{X}_n = \sum_{i=1}^{n}(X_i, Y_i)$, $n = 1, 2, \ldots$. When $\lambda \to \infty$ this random walk tends to a deterministic motion along the vertical line with constant rate. This is because both $\mathrm{E}[X]$ and $\mathrm{V}[Y]$ go to zero quadratically in $\mathrm{E}[Y]$. Thus over a finite interval y, which takes on the average $n = y/\mathrm{E}[Y]$ steps, the expected total displacement in the x-direction is $n\mathrm{E}[X] \sim y\mathrm{E}[Y]$, which goes to zero with $\mathrm{E}[Y]$ as $\lambda \to \infty$. Similarly the total variance of the displacement in the y-direction after n steps is $n\mathrm{V}[Y] \sim y\mathrm{E}[Y]$ and goes to zero as $\lambda \to \infty$ (the total variance of the x-displacement goes to zero even faster as the one step variance $\mathrm{V}[X] \sim \mathrm{E}[Y]^4$).

The fact that the independent random walks of both the end points tend to constant deterministic motion along the vertical lines, in the hindsight justifies considering each step starting from a vertical link of maximal length R; the wiggle and contraction of the added links tend to zero.

Finally, we are able to calculate the asymptotic behavior in the three cases starting from the unweighted one (A). From the above it follows that $\mathrm{E}[Y]$ defines the vertical packing distance. As there is one vertical link in every rectangle of height $\mathrm{E}[Y]$ and width $(1+\varepsilon)R$, cf. Figure 1, the reward per unit area is asymptotically $U(\lambda, R) \approx 1/(R\,\mathrm{E}[Y])$,

$$U(\lambda, R) \approx \frac{1}{\Gamma(\frac{4}{3})}\left(\frac{\lambda}{6R^4}\right)^{1/3}, \tag{14}$$

$$u(\nu) = \frac{1}{\Gamma(\frac{4}{3})}\left(\frac{\sqrt{6}}{\pi}\,\nu\right)^{-2/3}. \tag{15}$$

Similarly in the weighted case (B), as there is one vertical link of length R in every rectangle of height $\mathrm{E}[Y]$ and width $(2+\varepsilon)R$, the weight per unit area is asymptotically $U_x(\lambda, R) \approx 1/(2\mathrm{E}[Y])$,

$$U_x(\lambda, R) \approx \frac{1}{2\,\Gamma(\frac{4}{3})}\left(\frac{\lambda}{6R}\right)^{1/3}, \tag{16}$$

$$u_x(\nu) = \frac{1}{2\,\Gamma(\frac{4}{3})}\left(\frac{36}{\pi}\,\nu\right)^{-1/6}. \tag{17}$$

Finally in the third case (C), $U_l(\lambda, R) = 2\,U_x(\lambda, R)$, and $u_l(\nu) = 2\,u_x(\nu)$ as the number of links compared to the packing in Figure 1 can be doubled. Hence,

$$U_l(\lambda, R) \approx \frac{1}{\Gamma(\frac{4}{3})}\left(\frac{\lambda}{6R}\right)^{1/3}, \tag{18}$$

$$u_l(\nu) = \frac{1}{\Gamma(\frac{4}{3})}\left(\frac{36}{\pi}\,\nu\right)^{-1/6}. \tag{19}$$

For more realistic interference models, one can conjecture that the asymptotic tail of $u_*(\nu)$ comes down more rapidly than for the unidirectional interference model due to the fact that this model unrealistically allows multiple transmissions just outside the interference range of a receiving node.

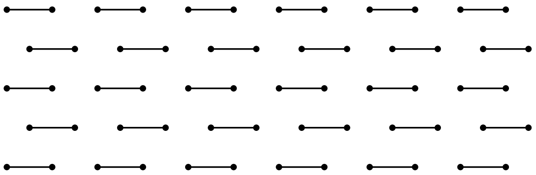


Fig. 3: When the end points of the links from a triangular lattice (with bidirectional interference in cases B and C), the number of transmitting nodes at the minimum distance from a node is maximized.

Bidirectional interference prevents the links from packing very close to each other. In the unweighted case (A), short links are preferred because then the interference ranges of the origin and destination overlap, and the exclusion area of the link is minimized. In the limit, the length of the links tends to zero, and the link constellation approaches equilateral triangular lattice with one transmission range between the links. Hence,

$$U(\lambda, R) \approx \frac{2}{\sqrt{3}R^2}, \tag{20}$$

$$u(\nu) = \frac{2\pi}{\sqrt{3}}\,\nu^{-1}. \tag{21}$$

In the weighted cases (B) and (C), the end points of the links form the same lattice as depicted in Figure 3. Hence, we have the reward R per area of $\sqrt{3}R^2$, and

$$U_{x,l}(\lambda, R) \approx \frac{1}{\sqrt{3}R}, \tag{22}$$

$$u_{x,l}(\nu) = \sqrt{\frac{\pi}{3}}\,\nu^{-1/2}. \tag{23}$$

The presented asymptotic behavior of $u_*(\nu)$ presumably gives everywhere an upper bound to the true curve. We return to the comparison with the numerical values later in Section 6.3. Note also that the $\nu \to \infty$ asymptotics do not apply when adjusting the transmission radius is allowed. Though the number of potential neighbors increases with a greater R, it is always possible to use the previous link configuration unless a better one becomes available. Thus in these cases, $u_*(\nu)$ approaches some limit.

5 Moving Window Algorithm

In this section we derive an algorithm similar to Retrospective optimization introduced in a study of reservation systems [19]. The algorithm is presented using an illustrative example. A pseudocode of a reference implementation and a note on complexity can be found from Appendix A.

The algorithm considers a small portion of the network at a time, a rectangular window that moves, and regarding the strip that the moving window covers during a simulation, the result is exact. The algorithm uses a binary tree to enumerate all the possible link combinations in the window area to find the maximum size or weight of an independent set of links per unit area so far conditioned on the choice of the combination of conflict-free links. The length of the simulation is not limited, and the covered strip can be of any desired length. We repeat the simulation for windows of different height to extrapolate the value of the maximum weight per area for an infinitely large network.

Because the height of the window in practice is limited, the top and the bottom of the strip can be connected to diminish the border effect and represent an infinite dimension, see Figure 4. The perimeter of the formed cylinder needs to be large enough for the results to be meaningful. The other direction can be handled by moving the window along the cylindrical network. The width of the window (i.e., the length of the cylindrical window) needs to be large enough for the window to contain all the links that can possibly interfere with the links that are going to enter the window in the future (that is, $3R$ which is the maximum length of two links and a $R+\varepsilon$ margin). The possible combinations of these links are maintained in the binary tree. The links that have already left the window do not affect the possible on/off-state of the links entering the window and can thus be removed by a procedure explained next.

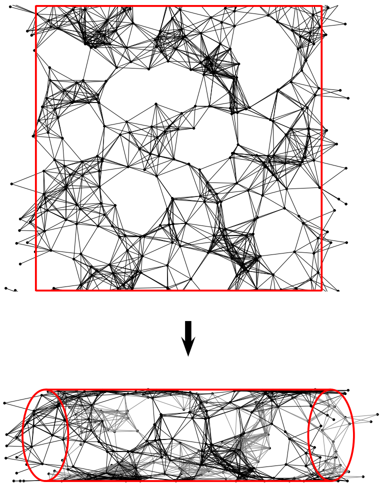


Fig. 4: The top and the bottom of the window are connected together to diminish the border effect. The formed cylinder is moved in the direction of its axis.

A rooted binary tree represents all the possible link combinations in the window area. Every edge of the tree describes whether the link corresponding to that level is active or not, and the value assigned to each leaf shows the maximum size or weight of the independent set thus far (starting from the initial position of the window) conditioned on the combination of active links in the window represented by the leaf. This is illustrated in Figure 5. The figure represents an example of a situation where the simulation of the unweighted case (A) (the values in the vertices represent the size of the independent set of links) with unidirectional interference has just started, and the first four nodes have entered the window making it possible to form six links. The maximum size of an independent set of links is 2 corresponding to activation set $\{uv, xw\}$ or $\{vu, wx\}$.

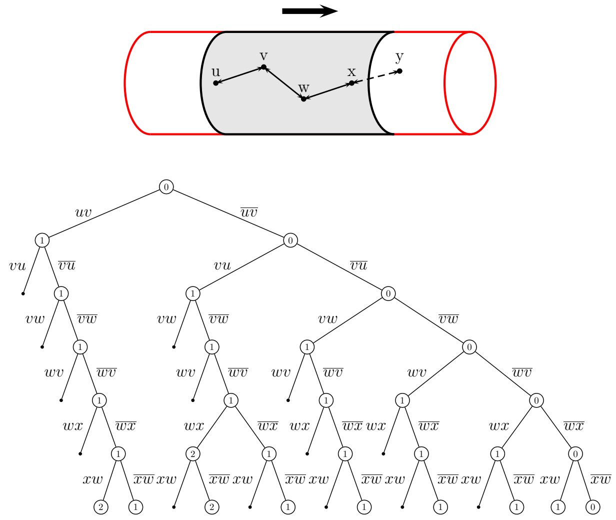


Fig. 5: A window containing 4 links and the corresponding binary tree with 6 levels representing the links (in alphabetical order) in the window.

When the window of Figure 5 is being moved to the right, the first event is the node u leaving the window. Since the entering and exiting links are independent, we can combine the on- and off-branches corresponding to a link whose endpoint has been dropped out of the window and choose the greater values for the new tree. That is, we compare leaves[2] that only differ in the dropped link and choose the maximum of those to be the value of the same node in the new tree where the level corresponding to the dropped link has been eliminated in this way. For example, when the first link to leave the window, uv, is being eliminated from the tree,

[2] Besides the leaves also the other vertices can be updated, but they hold no significance to the final result since the leaves cover all the possible link combinations.

the leaf with value 2 corresponding to the activation set $\{uv, xw\}$ (first from the left in Figure 5) is compared to the leaf with value 1 corresponding to the activation set $\{xw\}$ (second from the right), and the value of leaf $\{xw\}$ in the new tree in Figure 6 (second from the right) is thus 2. Also the link vu has to be removed from the tree when the node u leaves the window.

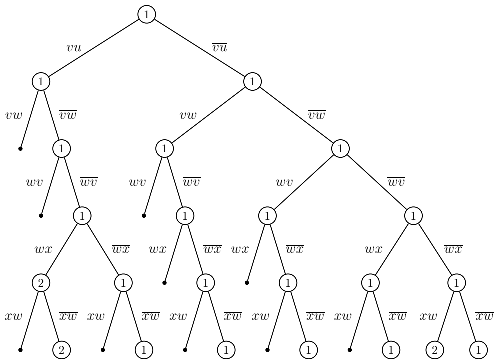


Fig. 6: The binary tree of Figure 5 after the first link to leave the window uv has been removed. In addition to the leaves, also the other vertices have been updated.

The next event, when moving the window, happens when node y enters and makes two new links possible. These new links are then added to the binary tree, after which the shape of the tree is the same as in Figure 5, but the value assigned to each leaf, except for those with vw or wv, is one higher since in these cases it is possible to use either the link uv or vu that have already exited the window. At this point, it is not explicitly visible which dropped links can be activated. Thus, the maximum value in the tree is the maximum size of the independent set of links so far given the set of active links in the window area. In this way we can generate the network realizations on the fly and progressively find the maximum size or weight of the independent set of links.

We do not maintain information about the links belonging to the maximum weight set, although, this information could be extracted from the algorithm with the cost of used memory. To further minimize the memory requirements the links are removed from the window as soon as they stop interfering with links that are going to enter the window in the future. When a link does not interfere with future links anymore, the information whether the link belongs to the maximum weight independent set is no longer required in the calculations, but the link can be removed from the binary tree maintaining the on/off status of the relevant links. This way the size of the binary tree, that is the bottleneck limiting the usefulness of the algorithm, can be kept as small as possible.

The algorithm limits in no way the length of the simulation in the direction in which the window moves, and when the execution is continued, the result converges without bias towards the true value. When the simulation is ended, the maximum size or weight of the independent set of links is the maximum of the values assigned to the leaves of the binary tree. The simulation is repeated to produce confidence intervals for the value. In the other direction, we have to rely on extrapolation and estimate the maximum value for an infinitely wide cylinder, as discussed in the next section.

5.1 Extrapolation

This section concentrates on extrapolating the maximum weight per area for the infinite plane from the measurements considering only strips of the network with limited height.

The simulations with the Moving Window Algorithm produce values $u_*(\nu, p)$, where p (in units of R) is the perimeter of the cylinder. For a given ν, a range of p are needed to extrapolate $u_*(\nu)$ to an infinitely wide cylinder. Figure 7 represents $u(p)$ of case A with unidirectional interference and fixed R for different values of ν. As seen from the figure, the narrowest cylinders do not give a reliable estimate for larger values of ν. The exact number of values of p required for the extrapolation depends heavily on the case studied as discussed next related to the x-progress case.

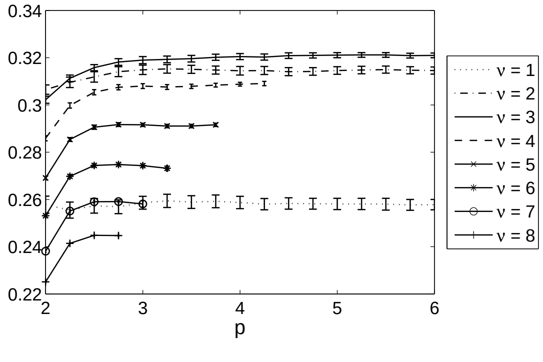


Fig. 7: Function $u(p)$ with unidirectional interference and fixed R for different values of ν, and the 95 % confidence intervals.

The second unidirectional case (B) with x-progress differs from the other unidirectional cases as it is the only directed case. The working principle of the algorithm does not depend on the direction of the traffic, i.e., the direction in which the progress of the maximal independent set is calculated, but it has to be fixed. We have two extremes: the direction is parallel with the direction in which the cylinder moves (along the cylinder) or the progress is calculated perpendicular to the movement of the cylinder (around the cylinder). In the latter case $u(p)$ depends heavily on the number of link columns that we are able to fit around the cylinder. The maxima appear when the perimeter is approximately a multiple of $2R$, meaning that we are able to fit full-length links and the margins $R + \varepsilon$ between them. When the direction of the progress is turned by a right angle, we get more stable results as the vertical distance between the links in a column is more stochastic. This effect is illustrated in Figure 8. The observation supports the assumption made in Section 4.2 about the most efficient way of packing the links, that is, to form vertical columns.

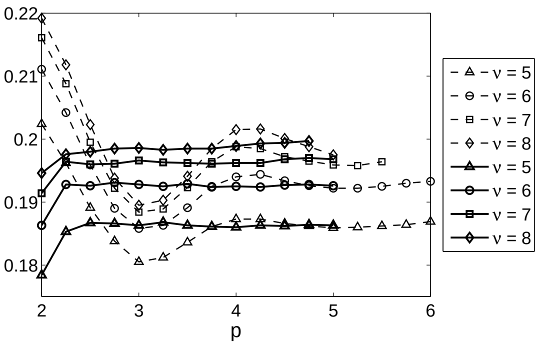


Fig. 8: Function $u_x(p)$ with unidirectional interference and fixed R for different values of ν when direction of progress is around the cylinder (dashed lines) and along the cylinder (solid lines).

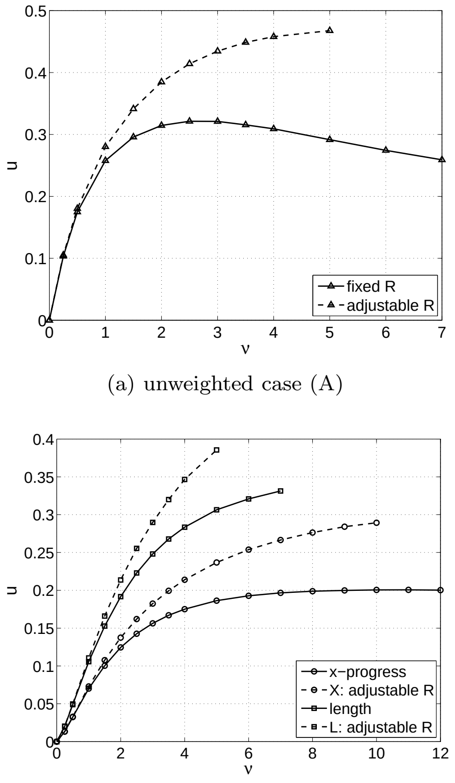
With bidirectional interference and fixed transmission radius, the active links start to form patterns when the network gets denser (recall Fig. 3). The difference to the towers formed in the unidirectional case is that the distance between the links is more deterministic, and $u_*(\nu, p)$ is thus more sensitive to the perimeter of the cylinder. The undulation is the heaviest in the weighted cases and with large values of ν, and the interesting parameter region of case B is thus the most difficult to reach. When the transmission range is adjustable, $u_*(\nu, p)$ is more stable as the effective number of neighbors is smaller.

6 Numerical results

In this section we present the numerical results obtained by the Moving Window Algorithm and extrapolation techniques of Section 5 for the three cases: unweighted (A), weighted by x-progress (B), and weighted by length (C) with both unidirectional and bidirectional interference. In addition to fixed transmission radius R, we consider transmission radii freely adjustable up to this maximum value. In this case the parameter ν is defined to correspond to the mean number of neighbors within the maximum range.

6.1 Unidirectional interference

We begin with the unidirectional interference model. Figure 9a shows $u(\nu)$ for the unweighted case (A) with both fixed and adjustable transmission radius. With a fixed transmission radius the maximum occurs at $\nu^* = 2.7$ and equals 0.32. The curve with power control is an increasing one as all configurations that are feasible with a given maximum radius are also possible with a greater maximum, and being upper bounded by the theoretical maximum of $\frac{1}{2}$, i.e., one link per two nodes, it tends to a limit when $\nu \to \infty$. As can be seen, the limit is relatively close to the theoretical maximum, implying that the maximum gain from a freely adjustable transmission radius is approximately 50 % (30 % at ν^*).



(a) unweighted case (A)

(b) weighted cases (B) and (C)

Fig. 9: Function $u_*(\nu)$ for unidirectional interference with and without power control. The values for case A with fixed R are those extrapolated from Figure 7.

Table 1: Numerical results

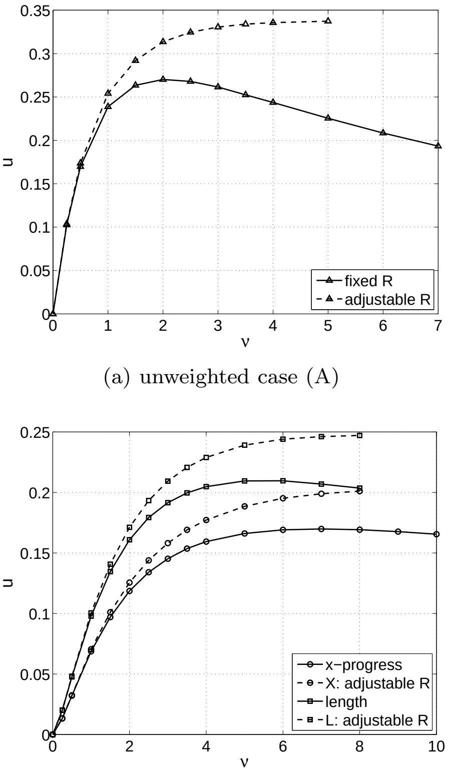
	unidirectional			bidirectional		
	fixed		adjustable	fixed		adjustable
	ν^*	$u(\nu^*)$	$u(\infty)$	ν^*	$u(\nu^*)$	$u(\infty)$
A	2.7	0.322	> 0.46	2.1	0.270	> 0.33
B	10	0.200	> 0.29	7.0	0.170	> 0.20
C	> 7	> 0.33	> 0.38	5.5	0.210	> 0.24

Figure 9b represents $u_*(\nu)$ for the weighted cases (B and C) with fixed transmission radius and with the possibility to reduce the transmission power to the minimum required. Even though the computational complexity grows with the number of links, it is possible to simulate x-progress with fixed transmission radius up to the optimal size of the neighborhood. The maximum occurs at $\nu^* \approx 10$ and equals 0.20. In the case of a length-weighted set (C), the number of links in the window is doubled compared to the second case, since we have to consider both directions separately. Thus we are not able to find the optimal neighborhood size. As with the unweighted case, the curves corresponding to cases with power control do not have a maximum but are increasing functions of ν tending to a limit when $\nu \to \infty$. Again, the maximum gain from an adjustable transmission radius is close to 50 % in case B.

6.2 Bidirectional interference

The unweighted case (A) with the bidirectional interference model is presented in Figure 10a. The maximum with fixed transmission radius equals 0.27 and is achieved with $\nu^* = 2$. The maximum gain from a freely adjustable transmission radius is approximately 25 % (15 % at ν^*).

Figure 10b shows how $u_x(\nu)$ and $u_l(\nu)$ behave with bidirectional interference. The maxima are $u_x(7) = 0.17$ and $u_l(5.5) = 0.21$. The corresponding approximate gains from an adjustable transmission radius are 20 % and 15 % at ν^*.



(a) unweighted case (A)

(b) weighted cases (B) and (C)

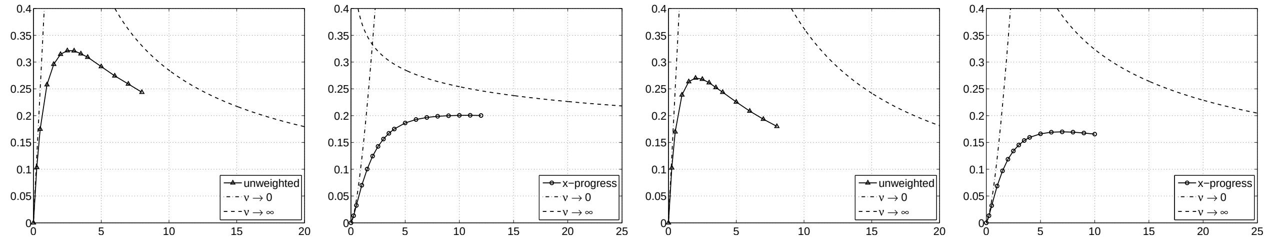
Fig. 10: Function $u_*(\nu)$ for bidirectional interference with and without power control.

6.3 Discussion

For reference, the most important numerical results from all cases have been gathered to Table 1. Since the curves are flat near the maximum, it is harder to determine the optimum neighborhood size than the maximum value itself (given with the accuracy of ±0.001). The values of ν^* are results of parabolic fits near the maximum and given with the accuracy of two significant figures. The lower bounds for the cases with adjustable transmission radius (and for unidirectional case C with fixed transmission radius) are based on the obtained maximum values.

As mentioned, the size of the binary tree places limitations on the feasible simulation parameters (ν and p). Since the process is stochastic, the number of links in the window may temporarily grow very large, and the size of the tree may exceed the available memory. Thus, we are only able to simulate cylinders wide enough until a certain value of ν in each case. An adjustable transmission radius is always computationally more complex than a fixed one since it increases the number of conflict-free link combinations and the size of the tree.

In the unidirectional case weighted by the x-progress (B) and in the bidirectional cases, we do not have to consider both links between two nodes. In case B, we never activate a link with a negative weight, and with bidirectional interference, both the links are equal and only one of them can be active. In these cases, it is possible to run simulations with higher values of ν, but the number of links in the window stays continuously on a high level, and the simulations are slow. The results are presented as far as it has been possible to proceed in



(a) unidirectional $u(\nu)$ (b) unidirectional $u_x(\nu)$ (c) bidirectional $u(\nu)$ (d) bidirectional $u_x(\nu)$

Fig. 11: Numerically evaluated curve for the functions $u_*(\nu)$ along with the low- and high-ν asymptotic curves.

reasonable time using a computer with at most 16 GB memory.

Finally, we compare the numerical results with the asymptotic results of Section 4. Figure 11a presents these for $u(\nu)$ and Figure 11b for $u_x(\nu)$ with unidirectional interference. Figures 11c and 11d present the corresponding curves with bidirectional interference. As can be seen from these figures, even the rudimentary analysis seems to yield a plausible asymptotic behavior for large ν. However, the figures also show that asymptotes alone do not characterize the curves accurately in the most interesting parameter area.

7 Conclusions

We studied the maximum weight independent sets in an infinite plane. This is a problem of stochastic geometry that relates to the question of the largest possible number of simultaneous successful transmissions, i.e., the spatial reuse in wireless multihop networks. The obtained results offer a point of comparison for practical MAC protocol implementations. The used simulation methods are also applicable for point processes other than the Poissonian one as well as for interference models, e.g., asymmetric ones, as long as the range of interference is limited.

We illustrated the working principles of the Moving Window Algorithm that allowed us to study the problem numerically. The algorithm produces exact results (weight per area) for any network realization in an arbitrarily long strip or, to reduce boundary effects, a cylinder obtained from the strip by joining its upper and lower ends. The network realization can be generated on-the-fly as the window moves, thus enabling unlimited simulations and accurate unbiased estimates. The height of the strip (perimeter of the cylinder) is, however, limited and to obtain results for an infinite plane an extrapolation technique was used.

Three different cases were studied. The first considered the number of simultaneous transmissions per unit area, the second the number of transmissions weighted by the progress of the transmissions in a given direction, and the third the number of transmissions weighted by their lengths. The cases were covered both when only one of the end points transmits (unidirectional interference) and when both end points transmit (bidirectional interference). In addition, the effect of power control was studied.

As expected, the problem turned out to be computationally demanding. However, we were able to produce previously unknown numerical results for all the cases. Only in the third unidirectional case, with links weighted by their lengths, were we unable to reach the most interesting parameter region, leaving room for more computational science oriented work in the future.

We presented also asymptotic analyses of the systems. The high-density asymptotics are a challenging problem, and our analysis is to be considered as a first attempt. Though capturing the essence of the problem, the analysis can be made more refined and improved in rigor.

Acknowledgements The work was done in the ABI and AWA projects funded by TEKES, Ericsson, and Nokia Siemens Networks. JN was supported by the Nokia Foundation.

Appendix

A Moving Window Algorithm

Table 2 represents a pseudo code for a reference version of the Moving Window Algorithm of Section 5. The algorithm takes as a parameter the length of the simulation, M. The algorithm first introduces the used variables: the current step, i, the set of nodes in the window, W, the set of links between the nodes in the window, L, the binary tree used to enumerate the possible link combinations in the window, T. At the beginning of the simulation, the binary tree is just a root node r. $x(w)$ and $y(w)$ refer to the coordinates of a particular node w, while $t(l)$ and $r(l)$ are the transmitting and receiving node of a link l, as in Section 3. Variables with a hat (e.g., $\hat{w}$) are temporary variables used to introduce new instances.

The main loop of the algorithm consists of the following phases. 1) The algorithm determines the exponential distance to the next node and moves the window forward. 2) The algorithm removes the nodes that are no longer in the window from W and

Table 3: Binary tree procedures

(a) ADD(n,l,b)

```
if INTERF(link($n$),$l$)
   $\vee$ $b$ = false do
  if rc($n$) = NULL do
    $w(\hat{n})$ := $w(n)$
    rc($n$) := $\hat{n}$
  else
    if lc($n$) $\neq$ NULL do
      ADD(lc($n$),$l$,false)
    end if
    ADD(rc($n$),$l$,false)
else
  if rc($n$) = NULL do
    $w(\hat{n})$ := $w(n) + w(l)$
    lc($n$) := $\hat{n}$
    $w(\hat{n})$ := $w(n)$
    rc($n$) := $\hat{n}$
  else
    if lc($n$) $\neq$ NULL do
      ADD(lc($n$),$l$,true)
    end if
    ADD(rc($n$),$l$,true)
  end if
end if
```

(b) RMV(n,l)

```
if rc($n$) $\neq$ NULL do
  if link(rc($n$)) = $l$ do
    if lc($n$) $\neq$ NULL do
      lc($n$) :=
      CMP(lc(lc($n$)),lc(rc($n$)))
      rc($n$) :=
      CMP(rc(lc($n$)),rc(rc($n$)))
    else
      lc($n$) := lc(rc($n$))
      rc($n$) := rc(rc($n$))
    end if
  else
    if lc($n$) $\neq$ NULL do
      RMV(lc($n$),$l$)
    end if
    RMV(rc($n$),$l$)
  end if
end if
```

(c) CMP(n,m)

```
if $n$ = NULL do
  return m
else if $m$ = NULL do
  return n
else
  if $w(n) > w(m)$ do
    lc($n$) := CMP(lc($n$),lc($m$))
    rc($n$) := CMP(rc($n$),rc($m$))
    return n
  else
    lc($m$) := CMP(lc($n$),lc($m$))
    rc($m$) := CMP(rc($n$),rc($m$))
    return m
  end if
end if
```

Table 2: MWA(M)

```
$i := 0$, $W := \emptyset$, $L := \emptyset$, $T := r$, $y(\hat{w}) := 0$
while $y(\hat{w}) < M$ do
  $i := i + 1$
  $S_i \sim \mathrm{Exp}(\lambda p)$
  $x(\hat{w}) \sim \mathrm{Uni}(p)$, $y(\hat{w}) := y(\hat{w}) + S_i$
  forall $w \in W$ do
    if $y(\hat{w}) - y(w) > 3R$ do
      $W := W \setminus \{w\}$
      forall $l \in L$ do
        if $t(l) = w \vee r(l) = w$ do
          $L := L \setminus \{l\}$
          RMV($r$,$l$)
        end if
      end for
    end if
  end for
  forall $w \in W$ do
    if $d(w, \hat{w}) \leq R$
      $t(\hat{l}) := \hat{w}$, $r(\hat{l}) := w$
      $L := L \cup \{\hat{l}\}$
      ADD($r$,$\hat{l}$,true)
      $t(\hat{l}) := w$, $r(\hat{l}) := \hat{w}$
      $L := L \cup \{\hat{l}\}$
      ADD($r$,$\hat{l}$,true)
    end if
  end for
  $W := W \cup \{\hat{w}\}$
end while
```

the related links from L and updates the tree (RMV). 3) The algorithm determines the new links that can be formed and adds them to L and updates the tree (ADD). Finally, it adds the new node to W.

Maintaining the binary tree is done with the recursive procedures ADD of Table 3a and RMV of Table 3b. They both take as a variable the current node of the tree, n, and the link l to be added or removed. In addition, ADD needs a Boolean variable, b, that tells whether or not it is possible to have the new link activated in the current branch of the tree. Additionally, RMV uses procedure CMP of Table 3c to recursively compare two nodes (n and m) of the tree. In the procedures, $w(\cdot)$ denotes the weight of either a node of the tree or a link. The left (link on) and right (link off) child of a node n are denoted by lc(n) and rc(n) respectively.

For large window sizes and dense networks, the running time of the algorithm is dominated by the time required to update the binary tree. In the worst case, the nodes communicate in pairs, i.e., the links do not interfere with each other (except for the two links between the same two nodes in the undirected cases), and the size of the binary tree is of the order of $O(2^{|L|})$ (i.e., $O(\sqrt{2}^{|W|})$) in case B and $O(\sqrt{3}^{|L|})$ (i.e., $O(\sqrt{3}^{|W|})$) in cases A and C. As in our case the underlying network is random, there is usually heavy interference between the links. If all the links in the window interfere with each other, the size of the tree behaves as $O(|L|^2)$, i.e., $O(|W|^4)$.

References

1. F. Baccelli, B. Blaszczyszyn, and P. Muhlethaler. An aloha protocol for multihop mobile wireless networks. *IEEE Transactions on Information Theory*, 52(2):421–436, 2006.
2. R. Catanuto, S. Toumpis, and G. Morabito. Opti{c,m}al: Optical/optimal routing in massively dense wireless networks. In *IEEE INFOCOM*, pages 1010–1018, 2007.
3. M. Durvy, O. Dousse, and P. Thiran. Self-organization properties of CSMA/CA systems and their consequences on fairness. *IEEE Transactions on Information Theory*, 55(3):931–943, 2009.

4. M. Durvy and P. Thiran. A packing approach to compare slotted and non-slotted medium access control. In *IEEE INFOCOM*, pages 1–12, April 2006.
5. F.V. Fomin, F. Grandoni, and D. Kratsch. A measure & conquer approach for the analysis of exact algorithms. *J. ACM*, 56(5):1–32, 2009.
6. M. Franceschetti, O Dousse, D.N.C. Tse, and P. Thiran. Closing the gap in the capacity of wireless networks via percolation theory. *IEEE Transactions on Information Theory*, 53(3):1009–1018, 2007.
7. M. Garey and D. Johnson. *Computers and intractability.* W. H. Freeman and Company, 1979.
8. P. Gupta and P.R. Kumar. The capacity of wireless networks. *IEEE Transactions on Information Theory*, 46(2):388–404, 2000.
9. E. Hyytiä and J. Virtamo. On traffic load distribution and load balancing in dense wireless multihop networks. *EURASIP Journal on Wireless Communications and Networking*, 2007:Article ID 16932, 15 pages, 2007.
10. P. Jacquet. Geometry of information propagation in massively dense ad hoc networks. In *Proceedings of ACM MobiHoc 04*, pages 157–162, 2004.
11. K. Jung and D. Shah. Low delay scheduling in wireless network. In *IEEE ISIT*, pages 1396–1400, 2007.
12. M. Kalantari and M. Shayman. Energy efficient routing in wireless sensor networks. In *Proc. of Conference on Information Sciences and Systems*, 2004.
13. E. Modiano, D. Shah, and G. Zussman. Maximizing throughput in wireless networks via gossiping. *SIGMETRICS Perform. Eval. Rev.*, 34(1):27–38, 2006.
14. J. Nousiainen and P. Lassila. Approximating maximum directed flow in a large wireless network. In *IEEE ICC*, Dresden, Germany, June 2009.
15. J. Nousiainen, J. Virtamo, and P. Lassila. Forwarding capacity of an infinite wireless network. In *Proceedings of ACM MSWiM*, pages 177–184, Vancouver, CA, Oct 2008.
16. J. Nousiainen, J. Virtamo, and P. Lassila. Maximum weight independent sets in an infinite plane. In *ITC21*, Paris, France, Sept. 2009.
17. H. Takagi and L. Kleinrock. Optimal transmission ranges for randomly distributed packet radio terminals. *IEEE Transactions on Communications*, 32(3):246–257, 1984.
18. L. Tassiulas and A. Ephremides. Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks. *IEEE Transactions on Automatic Control*, 37(12):1936–1949, 1992.
19. J.T. Virtamo. A model of reservation systems. *IEEE Transactions on Communications*, 40(1):109–118, 1992.